Industrial Science
Vol.1,Issue.2/Dec. 2013 **ISSN** : 2347-5420

Research Papers

VEHICLE DETECTION AND CLASSIFICATION USING CONSECUTIVE NEIGHBOURING FRAME DIFFERENCE METHOD

K.SOWJANYA AND G.CHAKRAVARTHY
Dept. Of Electronics and Communication Engineering,
VRSE Autonomonus, JNTU K ,Vijayawada.

Abstract

Background subtraction is a very popular approach for vehicle detection in traffic surveillance systems. A conventional color histogram (CCH) considers neither the color similarity across different bins nor the color dissimilarity in the same bin. Therefore, it is sensitive to noisy interference such as illumination changes and quantization errors. Furthermore, CCHs large dimension or histogram bins requires large computation on histogram comparison. However, structured motion patterns of the background (e.g., Vehicular traffic videos, waving leaves, spouting fountain, rippling water, etc.), which are distinctive from variations due to noise, are hardly tolerated in this assumption. To address these concerns, we introduce a background subtraction algorithm for temporally dynamic texture scenes. Specifically, we propose to adopt a clustering-based feature, called fuzzy color histogram (FCH), which has an ability of greatly attenuating color variations generated by background motions while still highlighting moving objects. Experimental results demonstrate that the proposed method is effective for background subtraction in dynamic texture scenes using LFCH features with adaptive updating procedure compared to several competitive methods proposed in the literature.

KEYWORDS:
Background subtraction, Conventional color histogram, fuzzy c-means ,fuzzy color histogram, illumination changes, membership matrix, structured motion patterns.

INTRODUCTION

An intelligent transportation system (ITS) is the application that incorporates electronic, computer, and communication technologies into vehicles and roadways for monitoring traffic conditions, reducing congestion, enhancing mobility, and so on. ITS is an evolving scientific and engineering discipline whose primary goal is to minimize the travel time of all travellers and merchandise while ensuring safety, through fair distribution of available resources, especially under the scenario of increasing travel speeds, and significantly large number of travellers, and a high demand for precise and timely information by travellers. Traffic surveillance is one of important issues in ITSs for traffic monitoring. The key goal of the traffic surveillance system is to estimate the desired traffic parameters through the detection, tracking and vehicle number counting process.

Video surveillance systems have long been in use to monitor security sensitive areas. The history of video surveillance consists of three generations of systems which are called 1GSS, 2GSS and 3GSS. The first generation surveillance systems (1GSS, 1960-1980) were based on analog sub systems for image acquisition, transmission and processing. The next generation surveillance systems (2GSS, 1980-2000) were hybrids in the sense that they used both analog and digital sub systems to resolve some drawbacks of its predecessors. Third generation surveillance systems (3GSS, 2000-) provide end-to-end digital systems.

The making of video surveillance systems "smart" requires fast, reliable and robust algorithms for moving object detection, classification, tracking and activity analysis. Starting from the 2GSS, a considerable amount of research has been devoted for the development of these intelligent algorithms. Moving object detection is the basic step for further analysis of video. It handles segmentation of moving objects from stationary background objects. This not only creates a focus of attention for higher level processing but also decreases computation time considerably. Commonly used techniques for object detection are background subtraction, statistical models, temporal 3 differencing and optical flow. Due to dynamic environmental conditions such as illumination changes, shadows and waving tree branches in the wind object segmentation is a difficult and significant problem that needs to be handled well for a robust visual surveillance system.

Object classification step categorizes detected objects into predefined classes such as human, vehicle, animal, clutter, etc. It is necessary to distinguish objects from each other in order to track and analyze their actions reliably. Currently, there are two major approaches towards moving object classification, which are shape-based and motion-based methods. Shape-based methods make use of the objects 2D spatial information whereas motion-based methods use temporal tracked features of objects for the classification solution. Detecting natural phenomenon such as fire and smoke may be incorporated into object classification components of the visual surveillance systems. Detecting fire and raising alarms make the human operators take precautions in a shorter time which would save properties, forests and animals from catastrophic consequences.

The next step in the video analysis is tracking, which can be simply defined as the creation of temporal correspondence among detected objects from frame to frame. This procedure provides temporal identification of the segmented regions and generates cohesive information about the objects in the monitored area such as trajectory, speed and direction. The output produced by tracking step is generally used to support and enhance motion segmentation, object classification and higher level activity analysis.

The final step of the smart video surveillance systems is to recognize the behaviours of objects and create high-level semantic descriptions of their actions. It may simply be considered as a classification problem of the temporal activity signals of the objects according to pre-labeled reference signals representing typical human actions. The outputs of these algorithms can be used both for providing the human operator with high level data to help him to make the decisions more accurately and in a shorter time and for offline indexing and searching stored video data effectively. The advances in the development of these algorithms would lead to breakthroughs in applications that use visual surveillance.

Nowadays, there is an instant need for the robust and reliable traffic surveillance system to improve traffic control and management with the problem of urban congestion spreads. Many traffic state parameters can be detected through traffic surveillance system, including traffic flow density, the length of queue, average traffic speed and total vehicle in fixed time interval. To achieve these goals, in past decades, there have been many approaches proposed for tackling related problems. Currently there are two kinds of traffic technologies. The dominant technology is loop sensor detection; this technology is efficient for vehicle speed and flow data collection. Although many detect devices such as closed loop, supersonic and radar are exist and widely used, the most important drawback of these equipment is their limitation in measuring some important traffic parameters and accurately assessing traffic condition. The first reason is that "blind" type of detection technology is employed. These sensors cannot provide full traffic scene information. Another very popular technique is video monitoring system.

The vision-based approach has the advantages of easy maintenance and high flexibility in traffic monitoring and, thus, becomes one of the most popular techniques used in traffic surveillance system [1]. In previous work normal background subtraction method is used for detecting the moving vehicles which resulted tackling and classification problems. In order to avoid those problems a new algorithm has been developed in this paper. However, the main challenging issues to the success of vehicle detection and classification are from vehicle occlusion, perspective effects, and camera configuration [2].

1.1 Motivation

The reason behind any method to incorporate into the system of ETC is to reduce the time at toll plazas and even the safety. In the past decades, numerous research projects aiming to detect traffic flow have been carried out in terms of measuring traffic performance. There are already several kinds of 5 traffic flow detection methods, such as, background subtraction method, frame subtraction method, land mark based method and edge detection method. Even many methods had been derived and implemented still we find some tackling and classification related problems. These problems mainly occur in vehicle occlusion, perspective effects, and camera configuration. Challenging to these problems is an important issue in the present ETC system. In order to overcome these problems on the highways they are some essential traffic parameters, such as vehicle counting and classification.

1.2 Objective

The main objective of this project is to present a robust traffic surveillance system for vehicle counting and classification. In order to avoid occlusions among and classification problems a new method has been implemented. The consecutive neighbouring frame difference method is firstly used to detect the moving regions from the highway scene. Some morphological operations are used to remove the shadow noise and to detect the moving object correctly. After vehicle detection, a region-based vehicle tracking method is used for building the correspondence between vehicles detected at different time instants. The four parameters, such as aspect ratio and compactness are used to classify vehicles.

1.3 Relevance To ETC

Electronic Toll Collection is a generic term for the application of various technologies designed to automate the traditional toll collection process of stop-and-pay. Drivers can pass toll collection points at various speeds depending upon technology deployed and the toll is collected automatically. An ETC system typically includes Automatic Vehicle Identification (AVI), automatic Vehicle classification (AVC) and Vehicle Enforcement system. The proposed system relates the ETC by vehicle detection, tracking, classification and counting. The combination of these techniques allows vehicles to pass through a toll facility without requiring any action by the driver (i.e., stopping at toll plazas to pay cash). Vehicle Classification will be done at ETC counter. 6

1.4 Organization of Thesis

Chapter 1. The first chapter gives a brief introduction to the importance of Intelligent Transportation system.
Chapter 2. The second chapter briefly reviews the literature survey.
Chapter 3. This chapter gives fundamentals of digital image processing and detailed information on Segmentation methods.
Chapter 4. This chapter briefly explains the methods of vehicle detection and classification
Chapter 5. This chapter gives the design implementation of vehicle detection, and vehicle classification.
Chapter 6. This chapter gives the results of the proposed solution followed by discussions.
Chapter 7. In this chapter the thesis is concluded wit

2. LITERATURE SURVEY

This report focuses to construct the traffic surveillance system on the highway for estimating traffic parameters, such as vehicle counting and classification. Vision based vehicle detection and classification plays an important role in real time traffic management systems. The real-time vehicle detection in video streams relies heavily on image processing techniques, such as motion segmentation, edge detection and digital filtering etc. The vehicle detection techniques developed by researchers have been given by this literature survey.

K. Park, D. Lee, and Y. Park,[1] presented an automatic traffic surveillance system to estimate important traffic parameters from video sequences using only one camera. They introduced a new "linearity" feature in vehicle representation. This method is different from traditional methods and it classifies vehicles to only cars and non-cars. The developed system can well tackle the problem of vehicle occlusions caused by shadows, which often lead to the failure of further vehicle counting and classification. This problem is solved by a novel line-based shadow algorithm that uses a set of lines to eliminate all unwanted shadows. The used lines are devised from the information of lane-dividing lines. An automatic scheme is implemented to detect the lane-dividing lines. The found lane-dividing lines can also provide important information for feature normalization, which can make the vehicle size more invariant, and thus improves the accuracy of vehicle classification.

H. Yalcin, M. Herbert, R. Collins, [2] presents algorithms for vision-based detection and classification of vehicles in monocular image sequences of traffic scenes recorded by a stationary camera. Processing is done at three levels: raw images, region level, and vehicle level. Vehicles are modelled as rectangular patches with certain dynamic behaviour. The proposed method is based on the establishment of correspondences between regions and vehicles, as the vehicles move through the image sequence. Experimental results from highway scenes are provided which demonstrate the effectiveness of the method. We also briefly describe an interactive camera calibration tool that we have developed for recovering the camera parameters using features in the image selected by the user.

H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos [3] for the problem of tracking vehicles on freeways using machine vision, existing systems work well in free-flowing traffic. Traffic engineers, however, are more interested in monitoring freeways when there is congestion, and current systems break down for congested traffic due to the problem of partial occlusion. They are developing a feature based tracking approach for the task of tracking vehicles under congestion. Instead of tracking entire vehicles, vehicle sub-features are tracked to make the system robust to partial occlusion. In order to group together sub-features that come from the same vehicle, the constraint of common motion is used. In this paper we describe the system, 0 real-time implementation using a network of DSP chips, and experiments of the system on approximately 44 lane hours of video data.

Z. Zhang, Y. Cai, K. Huang, and T. Tan, [4] describes the of an end-to-end method for extracting moving targets from a real-time video stream, classifying them into predefined categories according to image based properties, and then robustly tracking them. Moving targets are detected using the pixel wise difference between consecutive image frames. A classification metric is applied these targets with a temporal consistency constraint to classify them into three categories: human, vehicle or background clutter once classified, targets are tracked by a combination of temporal differencing and template matching. The resulting system robustly identifies targets of interest, rejects background clutter; and continually tracks over large distances and periods of time despite occlusions, appearance changes and cessation of target motion.

P. G. Michalopoulos [5] present an approach for recognizing and classifying moving vehicles in monocular images sequences of traffic scenes recorded by a stationary camera. A generic vehicle model, represented by a 3D polyhedral model describe by 12 length parameters, is used to cover the different shapes of road vehicles. The object recognition process is initialized by formulating a model hypothesis using a reference model and initial values provided by a motion segmentation step from a model-based tracking system described previously. This model hypothesis is verified and the shape as well as the pose and motion parameters of the object are estimated simultaneously. A recursive estimator updates the state description of the shape and motion parameters .In this way all relevant data from the image sequence evaluated so far are accumulated and used for the shape parameter estimation and classification of a moving vehicle. A classification is based on the assumption that differences between class members can be considered as deformations of the shape of a stored prototype.

E. Rivlin, M. Rudzsky, M. Goldenberg, U. Bogomolov, and S. Lapchev [7] Tracking vehicles is an important and challenging issue in video-based Intelligent Transportation Systems and has been broadly investigated in the past. This paper presents a robust and real-time method for tracking vehicles and the proposed algorithm includes two stages: object region extraction, vehicle tracking. Object region extraction is a key step and the concept of tracking vehicle is built upon the vehicle-segmentation method. According to the segmented vehicle shape, we propose a three-step predict method based on Kalman filter to track each vehicle. The proposed method has been tested on a number of monocular traffic-image sequences and the experimental results show that the algorithm is robust and real-time. The correct rate of vehicle tracking is higher than 85 percent, independent of environmental conditions.

L. Xie, G. Zhu, Y. Wang, H. Xu, and Z. Zhang, [8] present a novel tracking method for effectively tracking objects in structured environments. The tracking method finds applications in security surveillance, traffic monitoring, etc. In these applications, the movements of objects are constrained by structured environments. Therefore, the relationship between objects and environments can be exploited as additional information for improving the performance of tracking. We use the environment state to model the relationship between the objects and environments, and integrate it into the framework of Bayesian tracking. In this paper, distance transform is used to model the environment state, and particle filtering is employed as the paradigm for solving the Bayesian tracking problem. The adaptive dynamics model and environment prior are devised for the particle filter to fully utilize the environment information in the tracking process. Experiments on some video surveillance sequences demonstrate the effectiveness and robustness of our approach for tracking object motions in structured environments.

J. Cao and L. Li, [9] presents a probabilistic approach for automatically segmenting foreground objects from a video sequence. In order to save computation time and be robust tonoise effects, a region detection algorithm incorporating edge information is first proposed to identify the regions of interest, within which the spatial relationships are represented by a region adjacency graph..

Z. Zhang, W. Dong, K. Huang, and T. Tan, [10] present here a new algorithm for segmentation of intensity images which is robust, rapid, and free of tuning parameters. The method, however, requires the input of a number of seeds, either individual pixels or regions, which will control the formation of regions into which the image will be segmented. In this correspondence, we present the algorithm, discuss briefly its properties, and suggest two ways in which it can be employed, namely, by using manual seed selection or by automated procedures.

A. Wedel, T. Schoenemann, T. Brox, and D. Cremers [11] Background subtraction is a method typically used to segment moving regions in image sequences taken from a static 12 camera by comparing each new frame to a model of the scene background. We present a novel non-parametric background model and a background subtraction approach. The model can handle situations where the back-ground of the scene is cluttered and not completely static but contains small motions such as tree branches and bushes. The model estimates the probability of observing pixel intensity values based on a sample of intensity values for each pixel. The model adapts quickly to changes in the scene which enables very sensitive detection of moving targets. We also show how the model can use color information to suppress detection of shadows. The implementation of the model runs in real-time for both gray level and color imagery. Evaluation shows that this approach achieves very sensitive detection with very low false alarm rates.

S. Wender and K. Dietmayer [12] presents a real-time system for pedestrian tracking in sequences of gray scale images acquired by a stationary CCD camera. The objective is to integrate this system with a pedestrian control scheme for intersections. The system outputs the spatio-temporal coordinates of each pedestrian during the period the pedestrian is in the scene. Processing is done at three levels: raw images, blobs, and pedestrians. Our method models pedestrians as rectangular patches with a certain dynamic behaviour. Kalman filtering is used to estimate pedestrian parameters. The system was implemented on a Data cube MaxVideo equipped with a Data cube Max860 and was able to achieve a peak performance of over 20 frames per second.

B. T.Morris and M. M. Trivedi[13] presents a Learning, modeling, and classification of vehicle track patterns from live video.The objective is to improve the classification patteren of moving objects or vehicles.

S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, [14] presents a Detection and classification of vehicles,in different methods by using various filters. Detecting the vehicles using kalman filter method and background subtraction method.it will be easy to identify the vehicle and easy in the traffic flow.

N. U. Rashid, N. C. Mithun, B. R. Joy, and S. M. M. Rahman,[15] presents a Detection and classification of vehicles from a video using time-spatial image, it is the extension of the detection and classification of vehicles.

E. M. Kornaropoulos and P. Tsakalides, [16] presents A novel KNN classifier for acoustic vehicle classification based on alpha-stable statistical modeling,by using KNN the vehicle classification is easy to classify and it gives accurate results of classification.

R. C. Gonzalez and R. E. Woods,[17]presents Digital Image Processing, about image processing and color histogram

W. Pedrycz,[18] presents Knowledge-Based Clustering: From Data to Information Granules, and briefly gives the clarification of the different types of clustering.

Niluthpol Chowdhury Mithun, Nafi Ur Rashid, and S. M. Mahbubur Rahman[19],presents a detection and classification of vehicles from video using multiple time spatial images, it is the extension of the single time spatial images.

3. DIGITAL IMAGE PROCESSING

3.1 Basics of Digital Image Processing

An image may be defined as a 2-dimensional function, f(x, y), where x & y are spatial (plane) coordinates and the amplitude of f(x) at any pair of coordinates of f(x, y) is called the intensity or gray level of the image at that point. When x and y are the amplitude values of f(y) are all finite, discrete quantities, then we call that image as a digital image. The field of digital image processing refers to processing digital images by means of a digital computer. A digital image is composed of a finite number of elements, each of which has a particular location and these elements are referred to as picture elements, image elements, pels, or pixels.

Pixel is the term most widely used to denote the elements of a digital image. Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perceptions. However, unlike humans, who are limited to the visual band of the electromagnetic spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate on images generated by sources that humans are not accustomed to associating with images. These include ultrasound, electron microscopy, a computer generated images. Thus, digital images processing encompasses a void and varied field of applications.

3.1.1 Fundamental steps in digital image processing

Image acquisition is the first process shown in Fig.3.1. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves pre-processing, such as scaling. Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better."

It is important to keep in mind that enhancement is a very subjective area of image processing. Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation. Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a "good" enhancement result.

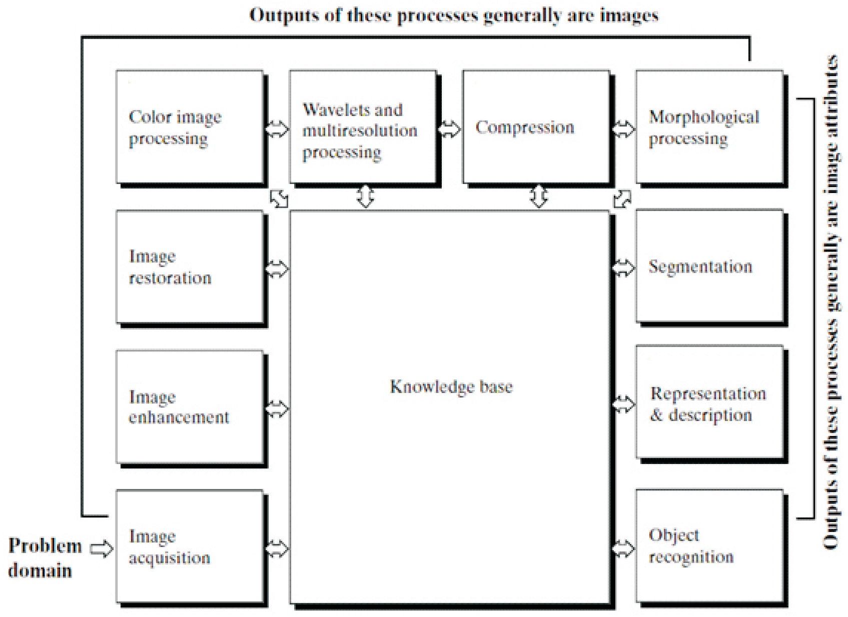


Figure 3.1 Fundamental Steps in Digital Image Processing

Colour image processing is an area that has been gaining importance due of the significant increase in the use of digital images over the Internet. Wavelets are the foundation for representing images in various degrees of resolution. In particular, it is used for image data compression and pyramidal representation, where images are subdivided successively into smaller regions.

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

Image compression can be lossy or lossless. Lossless compression is preferred for archival purposes and often medical imaging, technical drawings, clip art or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artefact. Lossy methods are especially natural suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually lossless.

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The decision that must be made is whether the data should be represented as a boundary or as a complete region.

Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Recognition is the process that assigns a label (e.g.," vehicle") to an object based on its descriptors. We conclude our coverage of digital image processing with the development of methods for recognition of individual objects. So far we have said nothing about the need for prior knowledge or about the interaction between the knowledge base and the processing modules. Knowledge about a problem domain is coded into an image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change-detection applications. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules.

3.1.2 Components of an Image Processing System

As recently as the mid-1980s, numerous models of image processing systems being sold throughout the world were rather substantial peripheral devices that attached to equally substantial host computers. Late in the 1980s and early in the 1990s, the market shifted to image processing hardware in the form of single boards designed to be compatible with industry standard buses and to fit into engineering workstation cabinets and personal computers. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing.

Although large-scale image processing systems still are being sold for massive imaging applications, such as processing of satellite images, the trend 18 Continues toward miniaturizing and blending of general-purpose small computers with specialized image processing hardware. Figure 3.2

shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is followed. In addition to lowering costs, this market shift also served as a catalyst for a significant number of new companies whose specialty is the development of software written specifically for image processing. Figure 3.2 shows the basic components comprising a typical general-purpose system used for digital image processing. The function of each component is followed

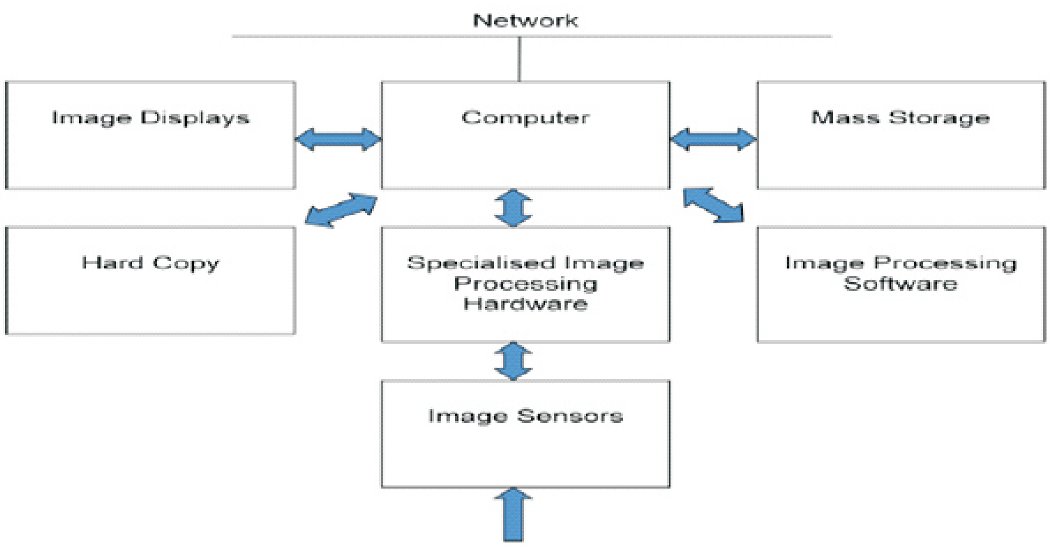


Figure 3.2 Components of an Image Processing System

According to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form. For instance, in a digital video camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts these outputs to digital data.

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed. In other words, this unit performs functions that require fast data throughputs (e.g., digitizing and averaging video images at 30 frames) that the typical main computer cannot handle. The computer in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance, but our interest here is on general-purpose image processing systems. In these systems, almost any well-equipped PC-type machine is suitable for offline image processing tasks.

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules. More sophisticated software packages allow the integration of those modules and general-purpose software commands from at least one computer language.

Mass storage capability is a must in image processing applications. An image of size 1024×1024 pixels, in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed. When dealing with thousands, or even millions, of images, providing adequate storage in an image processing system can be a challenge. Digital storage for image processing applications falls into three principal categories: (1) short-term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access. Storage is measured in bytes eight bits, Kbytes one thousand bytes, Mbytes one million bytes, GBytes meaning Giga, or one billion, bytes, and Tbytes .

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers, that store one or more images and can be accessed rapidly, usually at video rates (e.g., at 30 complete images per second). The latter method allows virtually instantaneous image zoom, as well as scroll (vertical shifts) and pan (horizontal shifts). Frame buffers usually are housed in the specialized image processing hardware unit shown in Fig. 3.2. Online storage generally takes the form of magnetic disks or optical-media storage. The key factor characterizing on-line storage is frequent access to the stored data. Finally, archival storage is characterized by 20 massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

Image displays in use today are mainly colour preferably flat screen TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system. Seldom are there requirements for image display applications that cannot be met by display cards available commercially as part of the computer system. In some cases, it is necessary to have stereo displays, and these are implemented in the form of headgear containing two small displays embedded in goggles worn by the user.

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks. Film provides the highest possible resolution, but paper is the obvious medium of choice for written material. For presentations, images are displayed on film transparencies or in a digital medium if image projection equipment is used. The latter approach is gaining acceptance as the standard for image presentations.

Networking is almost a default function in any computer system in use today. Because of the large amount of data inherent in image processing applications, the key consideration in image transmission is bandwidth. In dedicated networks, this typically is not a problem, but communications with remote sites via the Internet are not always as efficient.

3.2 Image Representation

Digital images need a file format that holds the digital image data securely and permanently. There are different digital formats like jpeg, avi, mpeg etc., Storage of image information is crucial for its long-term preservation. Although digital images can be stored indefinitely without deterioration, they can be lost. A digital file can be permanently "lost" if it is 21 stored without regard for basic computer technology or n inappropriate storage media.

3.2.1 Digital Images

This section contains information about the properties of digital images, image types; file formats, the internal representation of images in IMAQ Vision, image borders, and image masks.

3.3 Types of Images

3.3.1 Gray scale image

In photography and computing, a gray scale or gray scale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

Gray scale images are distinct from one-bit black-and-white images, which in the context of computer imaging are images with only the two coloursand white also called bi-level or binary images. Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation. Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from a full color image; see the section about converting to grayscale.

3.3.2 Colour image

A color image is usually stored in memory as a raster map, a two-dimensional array of small integer triplets; or (rarely) as three separate raster maps, one for each channel. Fig 3.4 shows the color wheels which include all colors. Eight bits per sample (24 bits per pixel) seem to be adequate for most uses, but faint banding artifacts may still be visible in some smoothly varying images, especially those which have been subject to processing. Particularly demanding applications may use 10 bits per sample or more. On the other hand, some widely used image file formats and graphics cards may use only 8 bits per pixel, i.e. only 256 different colors, or 2–3 bits per channel.

Figure 3.3 Colour Wheels which include all colours

3.3.3 RGB Color Model

The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors. RGB is a device-dependent color model i.e. different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Fig 3.5 shows RGB color model. Thus an RGB value does not define the same color across devices without some kind of color management.

Typical RGB input devices are color TV and video cameras , image scanners, and digital cameras. Typical RGB output devices are TV sets of various technologies (CRT, LCD, plasma, etc.), computer and mobilephone displays, video projectors, multicolor LED displays, and large screens as JumboTron, etc. Color printers, on the other hand, are not RGB devices, but subtractive color devices (typically CMYK color model).

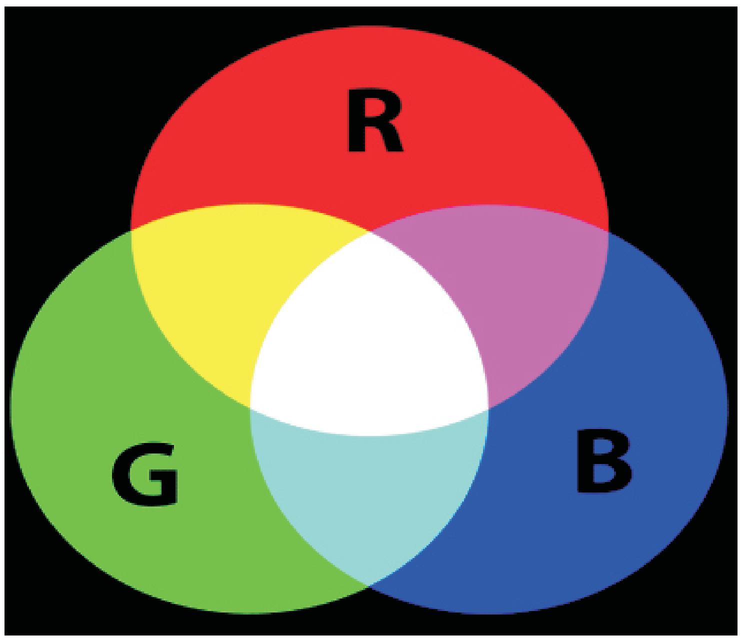


Figure 3.5 RGB color model

To form a color with RGB, three colored light beams (one red, one green, and one blue) must be superimposed (for example by emission from a black screen, or by reflection from a white screen). Each of the three beams is called a component of that color, and each of them can have an arbitrary intensity, from fully off to fully on, in the mixture. The RGB color model is additive in the sense that the three light beams are added together, and their light spectra add, wavelength for wavelength, to make the final color's spectrum.

Zero intensity for each component gives the darkest color (no light, considered the black), and full intensity of each gives a white; the quality of this white depends on the nature of the primary light sources, but if they are properly balanced, the result is a neutral white matching the system's white point. When the intensities for all the components are the same, the result is a shade of gray, darker or lighter depending on the intensity. When the intensities are different, the result is a colorized hue, more or less saturated depending on the difference of the strongest and weakest of the intensities of the primary colors employed.

When one of the components has the strongest intensity, the color is a hue near this primary color (reddish, greenish, or bluish), and when two components have the same strongest intensity, then the color is a hue of a secondary color (a shade of cyan, magenta or yellow). A secondary color is formed by the sum of two primary colors of equal intensity: cyan is green+blue, magenta is red+blue, and yellow is red+green. Every secondary color is the complement of one primary color; when a primary and its complementary secondary color are added together, the result is white: cyan complements red, magenta complements green and yellow complements blue.

3.3.4 Color to gray scale conversion

Conversion of a color image to grayscale is not unique; different weighting of the color channels effectively represents the effect of shooting black-and-white film with different-colored photographic filters on the cameras. A common strategy is to match the luminance of the grayscale image to the luminance of the color image.

To convert any color to a grayscale representation of its luminance, first one must obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding, by gamma expansion. Then, add together 30% of the red value, 59% of the green value, and 11% of the blue value (these weights depend on the exact choice of the RGB primaries, but are typical). Regardless of the scale employed (0.0 to 1.0, 0 to 255, 0% to 100%, etc.), the resultant number is the desired linear luminance value; it typically needs to be gamma compressed to get back to a conventional grayscale representation. To convert a gray intensity value to RGB, simply set all the three primary color components red, green and blue to the gray value, correcting to a different gamma if necessary. The reverse is also possible to build a full color image from their separate grayscale channels. By managing channels, using offsets,rotating and other manipulations, artistic effects can be achieved instead of accurately reproducing the original image.

The purpose of a color model is to facilitate the specification of colors in some standard, generally accepted way. In essence, a color model is a specification of a coordinate system and a subspace within that system where each color is represented by a single point.

3.4 Image Segmentation

The main objective of image segmentation is to extract various features of the image which can be merged or split in order to build objects of interest on which analysis and interpretation can be performed; it represents the first step in image analysis and pattern recognition.

3.4.1 Classification of Image Segmentation Techniques

Image segmentation can be broadly classified into two types
(i) Local segmentation
(ii) Global segmentation
(i) Local Segmentation

This deals with segmenting sub-images which are small windows on a whole image. The number of pixels available to local segmentation is much lower than global segmentation. Local segmentation must be frugal in its demands of pixel data.

(ii) Global Segmentation

It is concerned with segmenting a whole image. Global segmentation deals with segments consisting of a relatively large number of pixels. This makes estimated parameters values of global segments more robust.

3.4.2 Region Growing Approach to Image Segmentation

Region growing is an approach to image segmentation ion which neighbouring pixels are examined an added to region class if no edges are detected. This process is iterated for each boundary pixel in the region. if adjacent regions are found, a region –merging algorithm is used in which weak edges are dissolved and strong edges are left intact.

This requires a seed to begin with. Ideally the seed would be a region, but it could be a single pixel. A New segment is grown from the seed by assimilating as many neighbouring pixels as possible that meet the homogeneity criterion. The resultant is then removed from the process. A new seed is chosen from the remaining pixels, this continues until all pixels have been allocated to a segment.

These algorithms vary depending upon the criteria used to decide whether the pixel should be included in the region or not, the connectivity type used to determine neighbour, and the strategy used to visit neighbouring pixels. Region splitting and merging is a segmenting technique that takes spatial information into consideration.

(i) Splitting

1. Let R Represents the entire image .select a predicate p.
2. Split or subdivide the image successively into smaller and smaller quadrant regions.

(ii) Merging

Merge any adjacent regions are similar. The procedure for split and merge algorithm is given below:

1. Start with the whole image.
2. If the variance is too large, break it into quadrants.
3. Merge any adjacent regions that are similar.
4. Repeat steps (2) and (3) iteratively until no more splitting or merging occurs.

3.4.3 Image Segmentation Based on Thresholding

Thresholding techniques produce segments having pixels with similar intensities. Thresholding is a useful technique for establishing boundaries in images that contain solid objects resting on a contrasting background. There exist a large number of gray level based segmentation methods using either global or local image information. The thresholding technique requires that an object has homogeneous intensity and a background with a different intensity level. Such an image can be segmented into two regions by simple thresholding.

(i) Global Thresholding

It is the simplest and most widely used of all possible segmentation methods .in global Thresholding threshold value is chosen .

(ii) Adaptive Thresholding

Global thresholding works well if the objects of the interest have a reasonably uniform interior gray level and rest on a background of unequal but a relatively uniform gray level. In many cases, the background gray level isnot a constant, and object contrast varies within a image, in such cases, a threshold that works well in one area might not work well in other areas of the image.

(iii) Histogram based threshold selection

An image containing an object on a contrasting background has a bimodal gray level value. The two peaks correspond to the relatively large number of points inside and outside of the object. The dip between the peaks corresponds to relatively few points around edge of object. The dip is commonly used to establish n the threshold gray level

$$H(d) = -d/Dd\,(A/D) \qquad (3.2)$$

D Represents gray level value. A (D) is area of object obtained by thresholding at gray level d, h (d) is histogram.

3.5 Edge Based Segmentation

Edge detection is a fundamental tool in image processing and computer vision, particularly in the areas of feature detection and feature extraction, which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, if discontinuities present. Edge detection is a well-developed field on its own within image processing. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries.

Edge detection techniques have therefore been used as the base of another segmentation technique. The edges identified by edge detection are often disconnected. To segment an object from an image however, one needs closed region boundaries.

Image thresholding is one of the most widely used techniques for segmenting an image due to its simplicity. The basic approach is to select an intensity which is our threshold value, and any pixel which has an intensity value above this value is considered to be part of one region and anything below is considered to be part of other region. The key parameter in thresholding is obviously the choice of the threshold. Several different methods for choosing a threshold exist. The simplest method would be to choose the mean or median value, the rationale being that if the object pixels are brighter than the background, they should also be brighter than the average. In a noiseless image with uniform background and object values, the mean or median will work beautifully as the threshold, however generally speaking; this will not be the case.

A more sophisticated approach might be to create a histogram of the image pixel intensities and use the valley point as the threshold. The histogram approach assumes that there is some average value for the background and object pixels, but that the actual pixel values have some variation around these average values. However, computationally this is not as simple and many image histograms do not have clearly defined valley points. It is better to choose a method with threshold as it is simple, does not require too much prior knowledge of the image, and works well for noisy images.

Variables involved in the selection of an edge detection operator include:

(i) Edge orientation:

The geometry of the operator determines a characteristic direction in which it is most sensitive to edges. Operators can be optimized to look for horizontal, vertical, or diagonal edges.

(ii) Noise environment:

Edge detection is difficult in noisy images, since both the noise and the edges contain high-frequency content. Attempts to reduce the noise result in blurred and distorted edges. Operators used on noisy images are typically larger in scope, so they can average enough data to discount localized noisy pixels. This results in less accurate localization of the detected edges.

(iii) Edge Structure:

Not all edges involve a step change in intensity. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity. The operator needs to be chosen to be responsive to such a gradual change in those cases. Newer wavelet-based techniques actually characterize the nature of the transition for each edge in order to distinguish. For example, edges associated with hair from edges associated with a face.

The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to

1. discontinuities in depth,
2. discontinuities in surface orientation,
3. changes in material properties and
4. Variations in scene illumination.

In the ideal case, the result of applying an edge detector to an image may lead to a set of connected curves that indicate the boundaries of objects, the boundaries of surface markings as well as curves that correspond to discontinuities in surface orientation. Thus, applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. If the edge detection step is successful, then the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified. But, it is not always possible to obtain such ideal edges from real life images of moderate complexity. Edges extracted from non-trivial images are often hampered by fragmentation. It means that the edge curves are not connected, missing edge segments as well as false edges not corresponding to interesting phenomena in the image.

3.5.1 Edge Properties

The edges extracted from a two-dimensional image of a three-dimensional scene can be classified as either viewpoint dependent or viewpoint independent. A viewpoint independent edge typically reflects inherent properties of the three-dimensional objects, such as surface markings and surface shape. A viewpoint dependent edge may change as the viewpoint changes, and typically reflects the geometry of the scene, such as objects occluding one another. If the intensity differences were smaller between the 4th & 5th pixels than that of between the adjacent neighbouring pixels which were higher, it would not be as easy to say that there should be an edge in the corresponding region.

So to state a specific threshold on how large the intensity change between two neighbouring pixels must be for us to say that there should be an edge between these pixels is not always simple. Indeed, this is one of the reasons why edge detection may be a non-trivial problem unless the objects in 33 the scene are particularly simple and the illumination conditions can be well controlled.

3.5.2 Edge Detection techniques

The Canny edge detector including its variations is still a state of the art edge detector. Unless the preconditions are particularly suitable, it is hard to find an edge detector that performs significantly better than the canny edge detector.Canny-Deriche detector was derived from similar mathematical criteria as the canny edge detector, although starting from a discrete viewpoint and then leading to a set of recursive filters for image smoothing instead ofexponential filters or Gaussian filters.

3.4(a) RGB image edge detected image **3.4(b) Canny edge detected image**

There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories.
(i) Gradient: The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.
(ii) Laplacian: The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location. Fig 3.8 shows the input signal applied to Laplacian method.

A pixel location is declared an edge location if the value of the gradient exceeds some threshold. As mentioned before, edges will have higher pixel intensity values than those surrounding it. So once a threshold is set, you can compare the gradient value to the threshold value and detect an edge whenever the threshold is exceeded. Furthermore, when the first derivative is at a maximum, the second derivative is zero. As a result, another alternative to finding the location of an edge is to locate the zeros in the second derivative. Fig 3.5 shows the second derivative of f(t).

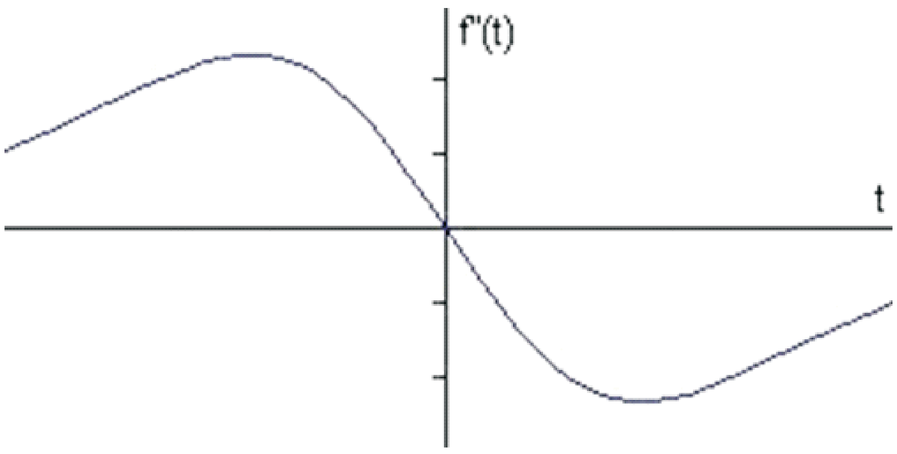


Figure 3.5 Second derivative of f(t)

(iii) Laplacian of Gaussian: The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single gray level image as input and produces another gray level image as output. The Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given by Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian.

Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian Smoothed before applying the Laplacian filter. This pre processing step reduces the high frequency noise components prior to the differentiation step. Since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages:

(i) Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations.

(ii) The LoG (Laplacian of Gaussian) kernel can be precalculated in advance so only one convolution needs to be performed at run-time on the image.

The 2-D LoG function cantered on zero and with Gaussian standard deviation has the form.

3.5.3 Sobel Operator

The operator consists of a pair of 3×3 convolution kernels as shown in below. One kernel is simply the other rotated by 90°. Fig 3.13 shows the Gradient Components of Sobel

+1	+2	+1
0	0	0
-1	- 2	-1

-1	0	+1
- 2	0	+2
-1	0	+1

GX GY

Figure 3.6 Gradient Components of Sobel

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid. One kernel is for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these as Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient.

3.5.4 Robert's cross operator

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point. The operator consists of a pair of 2×2 convolution kernels as shown in fig 3.14. One kernel is simply the other rotated by 90°. This is very similar to the Sobel operator.

These kernels are designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation and call these as GX and GY. These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient.

+1	0
0	-1

0	+1
-1	-1

Fig 3.6 Gradient Components of Roberts Cross operator

3.5.5 Prewitt's operator

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images. These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient

3.6 Performance of Edge Detection methods

The performance of the canny algorithm depends heavily on the adjustable parameters, σ, which is the standard deviation for the Gaussian filter, and the threshold values,T1‟ and T2‟. σ also controls the size of the Gaussian filter. The bigger the value for σ, the larger the size of the Gaussian filter becomes. This implies more blurring, necessary for noisy images, as well as detecting larger edges. However, the larger the scale of the Gaussian, the less accurate is the localization of the edge. A smaller value of σ implies a smaller Gaussian filter which limits the amount of blurring, maintaining finer edges in the image.

Canny‟s edge detection is computationally more expensive compared to Sobel, Prewitt and Robert‟s operators. The Canny‟s edge detection performs better than all these operators under almost all scenarios. Fig

3.7 shows comparision of edge detection methods.

3.7 Comparison of Edge detection methods

Fig 3.7(a) Original image

Fig 3.7(b) Sobel detected image

3.8 Morphological Operations

Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbours. By choosing the size and shape of the neighbourhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion. Before applying the morphological operations, structural element is created. The brief description of the structural elements is given below:

Structural Element

An essential part of the dilation and erosion operations is the structuring element used to probe the input image. A structuring element is a matrix consisting of only 0's and 1's that can have any arbitrary shape and size. The pixels with values of 1 define the neighbourhood.

Two-dimensional, or flat, structuring elements are typically much smaller than the image being processed. The centre pixel of the structuring element, called the origin, identifies the pixel of interest -- the pixel being processed. The pixels in the structuring element containing 1's define the neighbourhood of the structuring element. These pixels are also considered in dilation or erosion processing. Three-dimensional, or non-flat, structuring elements use 0's and 1's to define the extent of the structuring element in the x- and y-planes and add height values to define the third dimension.

3.8.1 Dilation

The value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1.

The following Fig. 3.7 illustrates the dilation of a binary image. Note how the structuring element defines the neighborhood of the pixel of interest, which is circled. The dilation function applies the appropriate rule to the pixels in the neighborhood and assigns a value to the corresponding pixel in the output image. In the Fig.3.7, the morphological dilation function sets the value of the output pixel to 1 because one of the elements in the neighborhood defined by the structuring element is on.

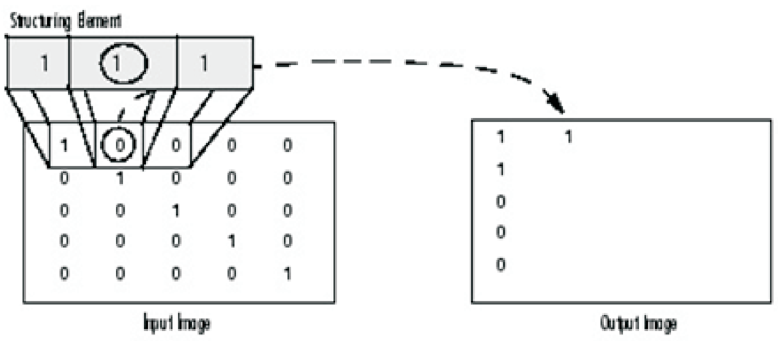


Figure. 3.8 Morphological Dilation of a Binary Image

The following Fig.3.8 illustrates morphological dilation processing for a gray scale image. The Fig.3.8 shows the processing of a particular pixel in the input image. Note how the function applies the rule to the input pixel's neighbourhood and uses the highest value of all the pixels in the neighbourhood as the value of the corresponding pixel in the output image

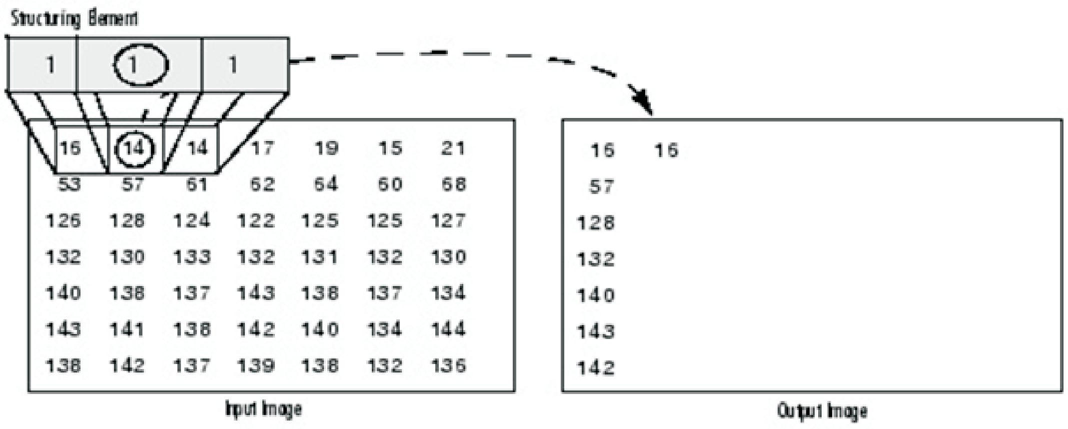


Figure. 3.9 Morphological Dilation of a Grayscale image

3.8.2 Erosion

The value of the output pixel is the minimum value of all the pixels in the input pixel's neighbourhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0. Binary erosion uses the following for its mask as its structural element.

Every pixel in the neighborhood must be 1 for the output pixel to be 1. Otherwise, the pixel will become 0. No matter what value the neighboring pixels have, if the central pixel is 0 the output pixel is 0. Just a single 0 pixel anywhere within the neighborhood will cause the output pixel to become 0. Erosion can be used to eliminate unwanted white noise pixels from an otherwise black area. The only condition in which a white pixel will remain white in the output image is if all of its neighbors are white. The effect on a binary image is to diminish, or erode, the edges of a white area of pixels from an image.

3.8.3 Opening

Opening is defined as an erosion followed by a dilation using the same structuring element for both operations. Opening is the dual of closing, i.e. opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element.

While erosion can be used to eliminate small clumps of undesirable foreground pixels, e.g. `salt noise'' quite effectively, it has the big disadvantage that it will affect all regions of foreground pixels indiscriminately. Opening gets around this by performing both an erosion and a dilation on the image. The effect of opening can be quite easily visualized. Imagine taking the structuring element and sliding it around inside each foreground region, without changing its orientation. All pixels which can be covered by the structuring element with the structuring element being entirely within the foreground region will be preserved. However, all foreground pixels which cannot be reached by the structuring element without parts of it moving out of the foreground region will be eroded away. After the opening has been carried out, the new boundaries of foreground regions will all be such that the structuring element fits inside them, and so further openings with the same element have no effect.

As with erosion and dilation, it is very common to use this 3×3 structuring element. The effect is rather subtle since the structuring element is quite compact and so it fits into the foreground boundaries quite well even before the opening operation. To increase the effect, multiple erosions are often performed with this element followed by the same number of dilations. This effectively performs an opening with a larger square structuring element.

3.8.4 Closing

Closing defined simply as a dilation followed by an erosion using the same structuring element for both operations. Closing is the dual of opening, i.e. closing the foreground pixels with a particular structuring element, is equivalent to closing the background with the same element.

One of the uses of dilation is to fill in small background color holes in images, e.g. `pepper noise'

One of the problems with doing this, however, is that the dilation will also distort all regions of pixels indiscriminately. By performing an erosion on the image after the dilation, i.e. closings reduce some of this effect. The effect of closing can be quite easily visualized. For any background boundary point, if the structuring element can be made to touch that point, without any part of the element being inside a foreground region, then that point remains background. If this is not possible, then the pixel is set to foreground. After the closing has been carried out the background region will be such that the structuring element can be made to cover any point in the background without any part of it also covering a foreground point, and so further closings will have no effect.

As with erosion and dilation, this particular 3×3 structuring element is the most commonly used, and in fact many implementations will have it hardwired into their code, in which case it is obviously not necessary to specify a separate structuring element. To achieve the effect of a closing with a larger structuring element, it is possible to perform multiple dilations followed by the same number of erosions.

Closing can sometimes be used to selectively fill in particular background regions of an image. Whether or not this can be done depends upon whether a suitable structuring element can be found that fits well inside regions that are to be preserved, but doesn't fit inside regions that are to be removed.

3.9 Image Processing in MATLAB

Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can perform image enhancement, image deblurring, feature detection, noise reduction, image segmentation, spatial transformations, and image registration [15]. Many functions in the toolbox are multithreaded to take advantage of multicore and multiprocessor computers.

Image Processing Toolbox supports a diverse set of image types, including high dynamic range, gigapixel resolution, ICC-compliant color, and tomographic images. Graphical tools let you explore an image, examine a region of pixels, adjust the contrast, create contours or histograms, and manipulate regions of interest (ROIs). With the toolbox algorithms you can restore degraded images, detect and measure features, analyze shapes and textures, and adjust the color balance of images.

Key Features:

1. Image enhancement, filtering, and deblurring
2. Image analysis, including segmentation, morphology, feature extraction, and measurement
3. Spatial transformations and image registration
4. Image transforms, including FFT, DCT, Radon, and fan-beam projection
5. Workflows for processing, displaying, and navigating arbitrarily large images
6. Modular interactive tools, including ROI selections, histograms, and distance measurements
7. ICC color management
8. Multidimensional image processing
9. Image-sequence and video display
10. DICOM import and export

3.9.1 Importing and Exporting Images

Image Processing Toolbox supports images generated by a wide range of devices, including digital cameras, satellite and airborne sensors, medical imaging devices, microscopes, telescopes, and other scientific instruments. You can visualize, analyze, and process these images in many data types, including single- and double-precision floating-point and signed and unsigned 8-, 16-, and 32-bit integers.

There are several ways to import and export images into and out of the MATLAB environment for processing. You can use Image Acquisition Toolbox to acquire live images from Web cameras, frame grabbers, DCAM-compatible cameras, and other devices. Using Database Toolbox, you can access images stored in ODBC/JDBC-compliant databases.

MATLAB supports standard data and image formats, including JPEG, JPEG-2000, TIFF, PNG, HDF, HDF-EOS, FITS, Microsoft Excel, ASCII, and binary files. It also supports the multiband image formats BIP and BIL, as used by LANDSAT for example. Low-level I/O and memory mapping functions enable you to develop custom routines for working with any data format. Image Processing Toolbox supports a number of specialized image file formats [15]. For medical images, it supports the DICOM file format, including associated metadata, as well as the Analyze 7.5 and Interfile formats. The toolbox can also read geospatial images in the NITF format and high dynamic range images in the HDR format. Fig 3.9 shows visualization and exploration of medical images in MATLAB.

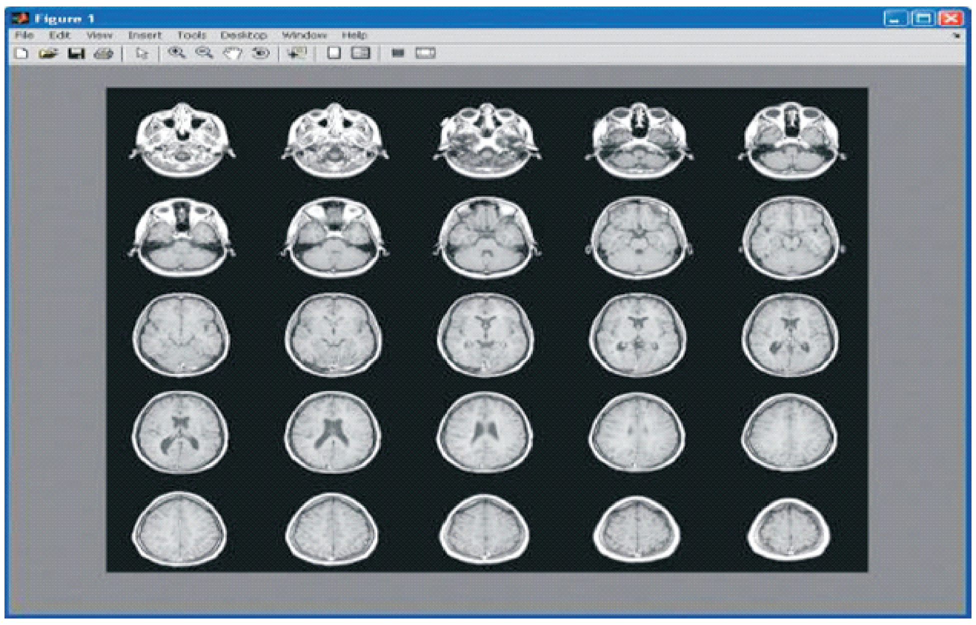


Figure3.10 Visualization and exploration of medical images in MATLAB

3.9.2 Analyzing Images

Image Processing Toolbox provides a comprehensive suite of reference-standard algorithms and graphical tools for image analysis tasks such as statistical analysis, feature extraction, and property measurement. Statistical functions let you analyze the general characteristics of an image by:

1. Computing the mean or standard deviation
2. Determining the intensity values along a line segment
3. Displaying an image histogram
4. Plotting a profile of intensity values

Edge-detection algorithms let you identify object boundaries in an image. These algorithms include the Sobel, Prewitt, Roberts, Canny, and Laplacian of Gaussian methods. The powerful canny method can detect true weak edges without being "fooled" by noise.

Image segmentation algorithms determine region boundaries in an image. You can explore many different approaches to image segmentation, including automatic thresholding, edge-based methods, and morphology-based methods such as the watershed transform, often used to segment touching objects.

3.9.3 Morphological operators

Morphological operators enable you to detect edges, enhance contrast, remove noise, segment an image into regions, thin regions, or perform skeletonization on regions. Image Processing Toolbox also contains advanced image analysis functions that let you:

1.Measure the properties of a specified image region, such as the area, center of mass, and bounding box.
2.Detect lines and extract line segments from an image using the Hough transform.
3.Measure properties, such as surface roughness or color variation, using texture analysis functions.

4 DETECTION AND CLASSIFICATION OF VEHICLES USING DIFFERENT METHODS

4.1 Vehicle Detection Methods:

Numerous research projects aiming to detect vehicles in traffic flow have been carried out in terms of measuring traffic performance during the past decades. There are

Motion-estimation-based vehicle detection techniques
optical flow estimation method
Gaussian scale mixture model method
background subtraction method

4.1.1 Motion estimation based vehicle detection techniques:

Motion estimation (ME) to replace the sum of absolute difference (SAD) measure. NCC is a more suitable similarity measure than SAD for reducing the temporal redundancy in video comparison since we can obtain flatter residual after motion compensation by using the NCC as the similarity measure in the motion estimation. The flat residual results in large DC term and smaller AC term, which is identified.

4.2 Single time spatial images:

In this approach, a time-spatial image (TSI) is generated using the luminance value of pixels of the moving objects that pass the virtual line. Each of the moving objects that passes the VDL creates a blob in a TSI, and the total number of vehicles is counted by detecting these blobs. One of the major causes of error in counting vehicles from a single TSI is due to the challenges of identifying moving objects that are occluded with each other. Counting errors may happen because of not only occlusion caused by the limitation of camera angle but also the morphological operations used for the generation of segmented blobs corresponding to moving objects that are close to each other. To reduce counting error, the method in uses an approximate width of a vehicle, and the method in uses the Hough transform for making a decision on whether the blobs are merged or not in a TSI. The merging of blobs is also identified by analyzing the motion fields of the merged regions and their trajectories. One of the major causes of error in counting vehicles from a single TSI is due to the challenges of identifying moving objects that are occluded with each other. Counting errors may happen because of not only occlusion caused by the limitation of camera angle but also the morphological operations used for the generation

A TSI of a video sequence is of segmented blobs corresponding to moving objects that are close to each other.

4.3 Multiple Time spatial images Generation:

generated by using a VDL on a frame through which a vehicle passes. The VDL is, in fact, a set of line indexes whose position is usually perpendicular to the motion of the vehicles and is independent of the frames. The TSI is obtained by placing the pixel strips of the frames on the VDL in chronological order. In these methods, a time-spatial image (TSI) is generated using the luminance value of pixels of the moving objects that pass the virtual line. Each of the moving objects that passes the VDL creates a blob in a TSI, and the total number of vehicles is counted by detecting these blobs.

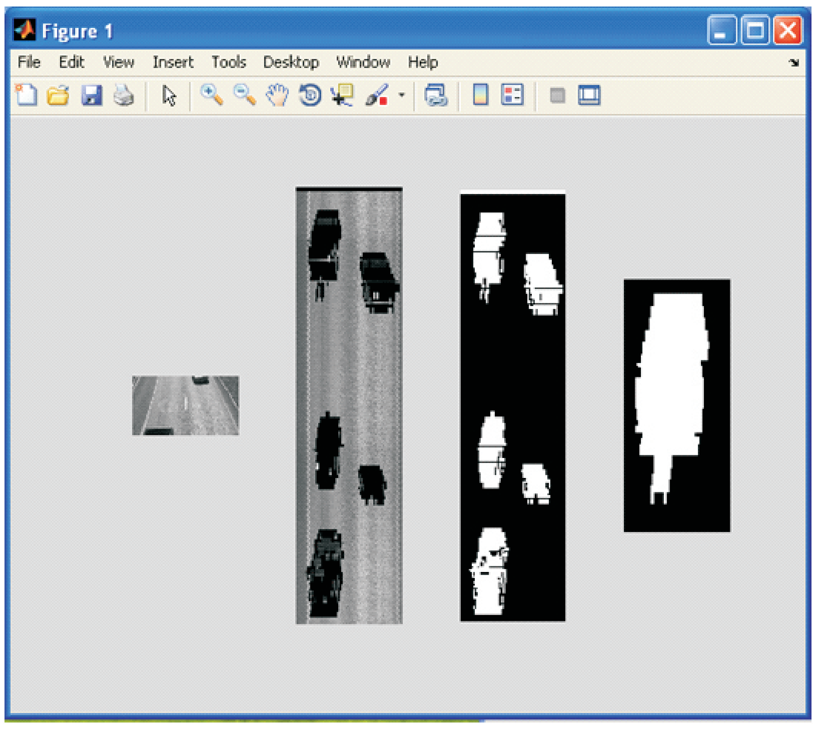


Figure :4.1 Multiple time spatial image obtained from multiple virtual detection lines.

In consecutive frame technique, a time-spatial image (TSI) is generated using the luminance value of pixels of the moving objects that pass the virtual line. Each of the moving objects that passes the VDL creates a blob in a TSI, and the total number of vehicles is counted by detecting these blobs. One of the major causes of error in counting vehicles from a single TSI is due to the challenges of identifying moving objects that are occluded with each other. Counting errors may happen because of not only occlusion caused by the limitation of camera angle but also the morphological operations used for the generation of segmented blobs corresponding to moving objects that are close to each other. To reduce counting error, the method in uses an approximate width of a vehicle, and the method in uses the Hough transform for making a decision on whether the blobs are merged or not in a TSI. The merging of blobs is also identified by analyzing the motion fields of the merged regions and their trajectories.

5. VEHICLE DETECTION

This section, describes how to detect the moving vehicles on the highway. Firstly, the moving regions are segmented from the background by using neighbouring frame difference method Then, the geometric properties of the segmented regions are used to filter out the false regions. In order to improve the accuracy of segmentation results, the merging blobs are also separated and finally the results are compared with the trained data set then finally the vehicle is classified by using KNN classifier and FCM clustering methods.

5. 1 K Nearest Neighborhood Classification

The consecutive neighbouring frame difference method. An Algorithm, whose computational complexity is sufficiently low and, at the same time, provides appreciably good classification performance. In this classification method, the popular kNN algorithm is used in two consecutive steps. In the first step, vehicles are grouped into one of the four broad classes, viz., 2W, 3W, 4W, and 6W. These classes roughly indicate the relative size of the vehicles. Next, each of these broad classes is further grouped into a particular type of the vehicle among those available in the traffic. A simple block diagram of this two-step classification algorithm for each of the TOBs and the corresponding KVFs . In both steps, classification is done by estimating the feature vectors from the TOBs and KVBs obtained from the TSIs and KVFs, respectively, and comparing with those kept in a trained set of feature data. To prepare the training database, individual data sets of every class and type are clustered using the fuzzy C-means (FCM) algorithm [8] for reducing the memory requirement and the searching time for the classifier.

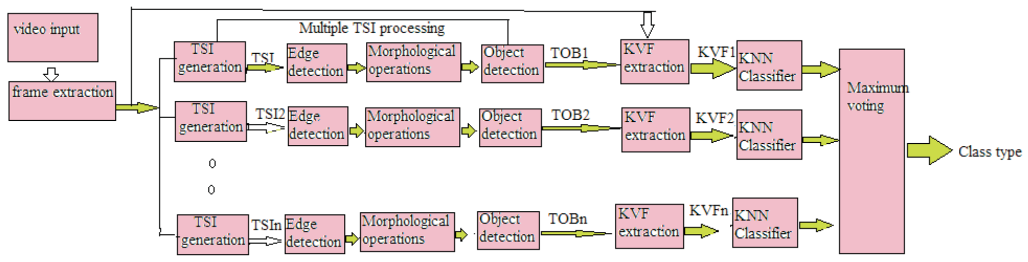


Figure: 5.1 Block diagram of the KNN classifier

The TSI-based vehicle counting method, background subtraction-based vehicle classification method , and SVDL method. It is to be noted that it is found that an increased number of VDLs does not significantly improve detection or classification performance. The frame difference α for the extraction of silhouette from the KVF is chosen as 2, and the values of parameters λ and μ are chosen as 5 and 7 for the segmentation of the TSI, respectively, and 4 and 5 for the KVF, respectively. To train the feature vectors, we have picked 6 vehicles from each of the vehicle types and performed FCM clustering. These trained feature vectors have been used to identify the class and type of a particular vehicle that provides a disjoint TOB for a given VDL using the two step kNN classifier. In both steps of the kNN classifier, the same number of neighboring data samples, i.e., the value of k, is used for measurement of closeness. A suitable value of k is empirically obtained by computing the percentage error in classification for each of the four classes, viz., 2W, 3W,4W, and 6W. The variations of such error with respect to the values of k . In consecutive frame difference method the value of k between 4 and 6 is a good choice to obtain good classification performance.

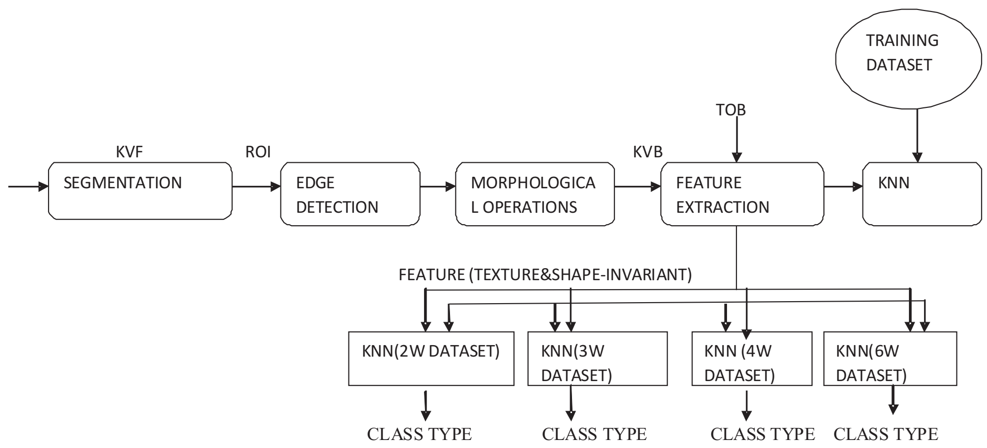


Figure: 5.2 Two step KNN classifier

In KNN classification of vehicles, with the use of two-step classifier, the probability of misclassification of the proposed method considerably reduces. Apparently, it may appear that higher classification accuracy in the proposed MVDL-based method is obtained at an increased computational load due to the two-step classification algorithm. However, it is worth mentioning that the processing time of the proposed MVDL-based method is approximately three times lower than that of the background-subtraction-based vehicle classification method.

5.2 Feature Extraction:

Inconsecutive frame difference technique, the features are chosen in such a way that they are sufficiently non redundant, distinctive, less sensitive to varying environment, and computationally efficient. Both geometric features such as shape-based, shape-invariant, and texture-based features are considered to construct the feature vector for each class of the vehicles. These features are obtained from the TOB of the TSI and the KVB of the KVF. We prefer to use six features to describe the shape of a vehicle

1) Width: A reliable approximation of the width of a vehicle is found from Wy estimated from the TOB.
2) Area: This is actually the number of pixels of the KVB.
3) Compactness: This feature, which determines the closeness to a circular shape of the vehicle, is evaluated as the ratio of the area and square of the perimeter of the KVB.
4) Length–width ratio: The length of the vehicle is obtained from the KVB and width as Wy.
5) Major-axis to minor-axis ratio: The lengths of the major axis and minor axis are obtained from the ellipse that fits the KVB best.
6) Rectangularity: This is the ratio between the number of pixels of the best-fitted rectangular box and that of the entire region of the KVB.

Because the height of truck and bus is larger than the one of car, it will have larger aspect ratio. Accordingly, the detected vehicle with aspect ratio smaller than a threshold is firstly labelled as car. For further classifying between bus and truck, the system analyzes the foreground masks of bus and truck. Since the bus is general a convex object, the segmented foreground will be more compact than the truck. Accordingly, the region with compactness larger than a threshold is classified as the bus.

5.3 Fuzzy C-Means Clustering

The consecutive neighbouring frame difference method works by assigning membership to each data point corresponding to each cluster center on the basis of distance between the cluster center and the data point. More the data is near to the cluster center more is its membership towards the particular cluster center. Clearly, summation of membership of each data point should be equal to one.This training set of data is clustered in desired number of vehicle classes using fuzzy C-means (FCM) algorithm, and thus reduces the memory requirement of the trained data set and speeds up the searching time for the classifier. Finally, a decision is made that a vehicle belongs to a certain class by a process of majority voting among the FCM-based training data records in the neighborhood using a weighted distance measurement.To prepare the training database, individual data sets of every class and type are clustered using the fuzzy C-means (FCM) algorithm for reducing the memory requirement and the searching time for the classifier.

Clustering of numerical data forms the basis of many classification and system modeling algorithms. The purpose of clustering is to identify natural groupings of data from a large data set to produce a concise representation of a system's behavior. Fuzzy Logic tools allow you to find clusters in input-output training data. You can use the cluster information to generate a Sugeno-type fuzzy inference system that best models the data behavior using a minimum number of rules. The rules partition themselves according to the fuzzy qualities associated with each of the data clusters. Use the command-line function, genfis2 to automatically accomplish this type of FIS generation. It provides a method that shows how to group data points that populate some multidimensional space into a specific number of different clusters.

This training set of data is clustered in desired number of vehicle classes using fuzzy C-means (FCM) algorithm, and thus reduces the memory requirement of the trained data set and speeds up the searching time for the classifier. Finally, a decision is made that a vehicle belongs to a certain class by a process of majority voting among the FCM-based training data records in the neighborhood using a weighted distance measurement. The FCM algorithm works by assigning membership to each data point corresponding to each cluster center on the basis of distance between the cluster center and the data point. More the data is near to the cluster center more is its membership towards the particular cluster center. Clearly, summation of membership of each data point should be equal to one. This training set of data is clustered in desired number of vehicle classes using fuzzy C-means (FCM) algorithm, and thus reduces the memory requirement of the trained data set and speeds up the searching time for the classifier.

In this system, the detected vehicle regions will be classified as car and truck or bus. After vehicle tracking, the system considers two parameters for classifying the vehicles. Those four parameters are aspect ratio and compactness C which are respectively defined as :

1.Centroid
2.Area
3.Rectangularity
4.Ratio.

CENTROID:

$$Y_1 : \frac{|c_\varepsilon^y - c_n^y|}{|c_n^y|} > T_C^y \quad (5.1)$$

WIDTH INY AXIS:

$$Y_2 : \frac{|w_\varepsilon^y - w_n^y|}{|w_n^y|} > T_W^y \quad (5.2)$$

WIDTH IN X AXIS

$$Y_3 : \frac{|w_\varepsilon^x - w_n^x|}{|c_n^y|} > T_W^x \quad (5.3)$$

AREA

$$Y_4 : \frac{|A_\varepsilon - A_n|}{|A_n|} > T_A \quad (5.4)$$

If the condision 5 satisfies then only TOBs of vehiclesare detected

$$Y : Y_1 \wedge (Y_2 \vee Y_3) \wedge Y_4 \quad (5.5)$$

6. IMPLEMENTATION OF KNN ALGORITHM TO MULTIPL TIME SPATIAL IMAGES

The fig. 5.1 shows the complete flow chart for detection of vehicles from consecutive neighbouring system. Initially a video is acquired by using a digital camera. The acquired video file is taken and then read all the frames in the video by the Matlab function aviread.

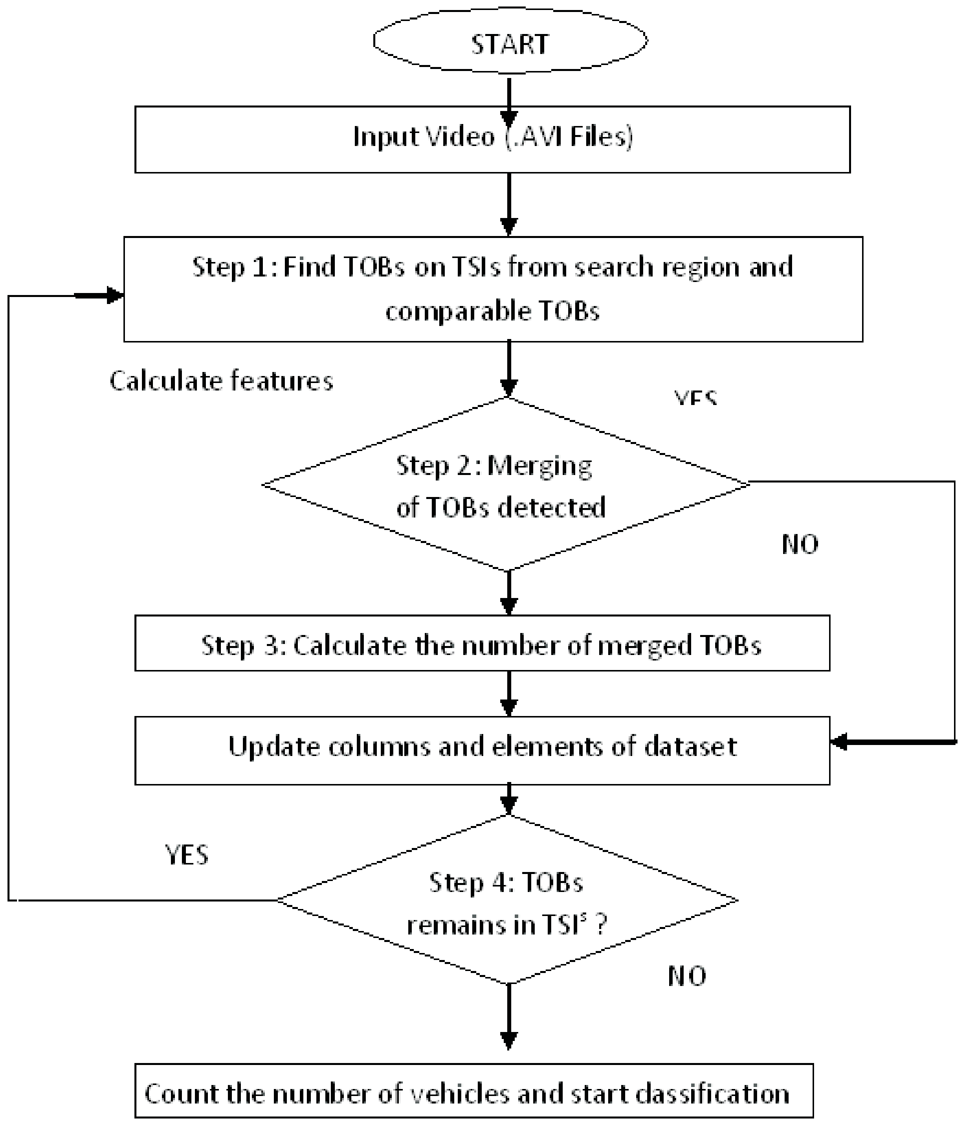


Figure: 6 Flow chart for detection of vehicle using two step KNN method

6. 1 Implementation

1. Firstly a video will be taken and then read all the frames in the video by thMatlab function Aviread.
2. Next the KNN frame difference method is used to detect the moving vehicles from the highway scene. Unfortunately, due to the merging of multiple TSIs; several vehicles will be occluded together and cannot be well separated. This makes the further vehicle detection and classification incorrect.
3. To avoid this problem some morphological operations are used to remove the merging of TSI object blobs and to detect the moving vehicle correctly.
4. After vehicle detection, an improved KNN frame difference method is used for building the correspondence between vehicles detected at different time instants. After detecting vehicle. The number of blobs are sufficient to count determines the number of vehicles present.
5. Finally, four parameters are considered such as aspect ratio and compactness to classify the vehicles. By using FCM clustering thus reduces the memory requirement of the trained data set and speeds up the searching time for the classifier.

The procedure of reading the video using Matlab software is given below.

(a) Access Video with MMREADER

The mmreader function constructs a multimedia reader object that can read video data from Multimedia file. To read the videos following functions are used. This function is very useful as it can

read the video file with different formats which are not been read by simple Function like AVIREAD. Mmreader supports the following formats: AVI, MPG, MPEG, WMV, ASF, and ASX
Video File = mmreader (PATH);

(b) Reading of the frames

The mmreader function is used for reading the video file. The frames from the video file can be extracted by using the cdata" function. It takes two images as input and produces as output a third image whose pixel values are simply those of the first image minus the corresponding pixel values from the second image. The subtraction of two images is performed straightforwardly in a single pass. Detecting moving objects in video sequences is very important in visual surveillance. The existence of cast-shadows would change the shape and size of the moving objects. Because the shadows usually move along the moving objects so that they may cause error in classification, which can cause various unwanted behaviour such as object shape distortion and object merging when the video images are captured with a fixed camera, KNN classification is a commonly used technique to segment moving objects [25]. The foreground objects are identified if they differ significantly from the previous frame. However, the detecting results of moving objects are usually under the influence of merging of TOBs in TSIs. By using The multiple TSIs method alone cannot effectively give the accurate results of vehicle detection because vehicle and background gray levels are too similar. For these reasons, it is critical to detect and the vehicles are occluded with each other in order to describe moving object correctly in visual surveillance and monitoring systems.

Some morphological operations are used for identifying and merging of the TSIs object blobs and to detect the moving object correctly. Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbours. By choosing the size and shape of the neighbourhood, you can construct a morphological operation that is sensitive to specific shapes in the input image. We need exact boundary of the objects and their TOBs in multiple TSIs. For this we applied consecutive KNN frame difference on the multiple TSIs of the each frame of the video.
rgb2gray - convert RGB image to gray scale

The above function is used to convert the rgb image into the gray scale so that we can apply the frame difference on it. For this process converted the previous frame as well as the current frame to gray scale and then performed the operation on the gray scale image of the video sequence.
F1 = rgb2gray (x{i})
x{i} = current frame of video sequence
F1 = Image in gray scale
F2 = rgb2gray (x{i -1})
x{i -1} = previous frame of video sequence
F2 = frame in gray scale
Imfill with „holes" is used to fill out the holes in a bounded area of an image.
After this we will subtract the current frame from the previous frame (F2)
F3 = im2double (imabsdiff (F1, F2))

Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. After the above step we will do dilation on the above frame and well use imfill to fill out the holes in a bound area of an image.
Imdilate (,) –This function is used for the dilation of image.
Merging: Multiple vehicles are detected at t-1.detected as a single region as t.

6.2 Vehicle Classification:

In this system, the detected vehicle regions will be classified as car and truck or bus. After vehicle tracking, the system considers four parameters for classifying the vehicles. Those four parameters are aspect ratio and compactness C which are respectively defined as :

1.Centroid
2.Area
3.Rectangularity
4.Ratio.

Because the height of truck and bus is larger than the one of car, it will have larger aspect ratio. Accordingly, the detected vehicle with aspect ratio smaller than a threshold is firstly labelled as car. For further classifying between bus and truck, the system analyzes the foreground masks of bus and truck.

7. RESULTS AND DISCUSSIONS

The proposed algorithm is simulated using MATLAB 7.0.4. The simulated outputs of vehicle detection and classification are shown below.

Fig 6.1 shows the results of vehicle detection using improved consecutive neighbouring frame difference method for two different videos. Fig 6.1(a) and 6.1(b) shows the frame difference at two instances. In figure 6.1, each frame has 3 sub images of which the first image shows the frame taken from the input video feed, the second image shows the TSI object blob of the first image and the third image shows the presence or absence of vehicles after implementing consecutive neighbouring frame difference method.

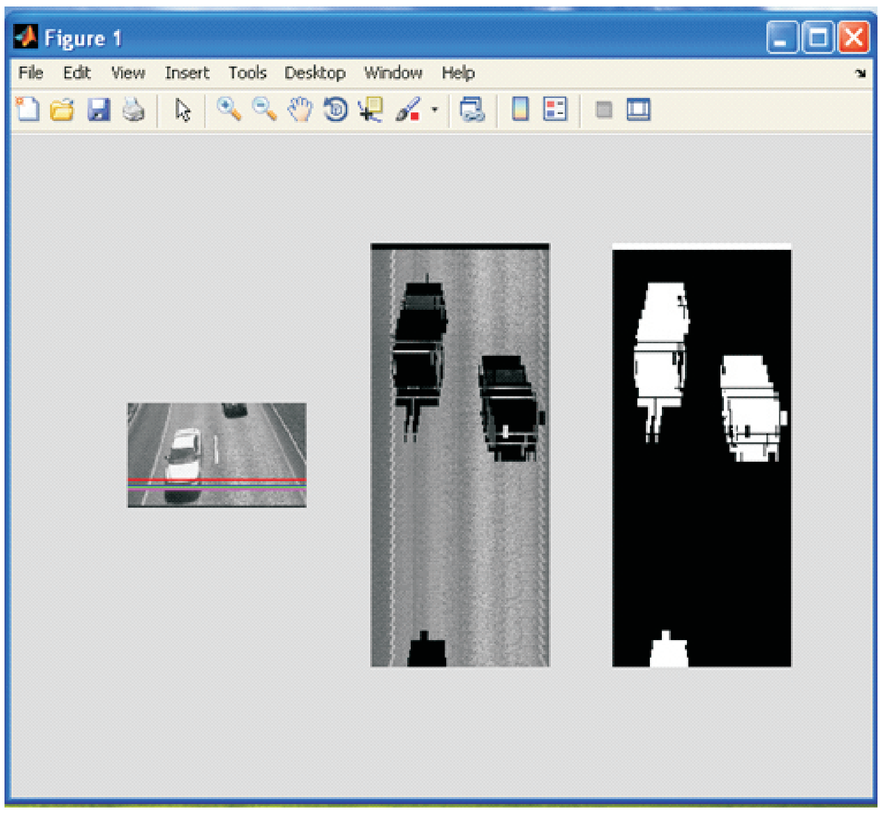


Figure :7.1 The TOBs are obtained from MVDLs after binary masking

A suitable value of k is empirically obtained by computing the percentage error in classification for each of the four classes, viz., 2W, 3W,4W, and 6W.

The variations of such error with respect to the values of k . In consecutive frame difference method the value of k between 4 and 6 is a good choice to obtain good classification performance.

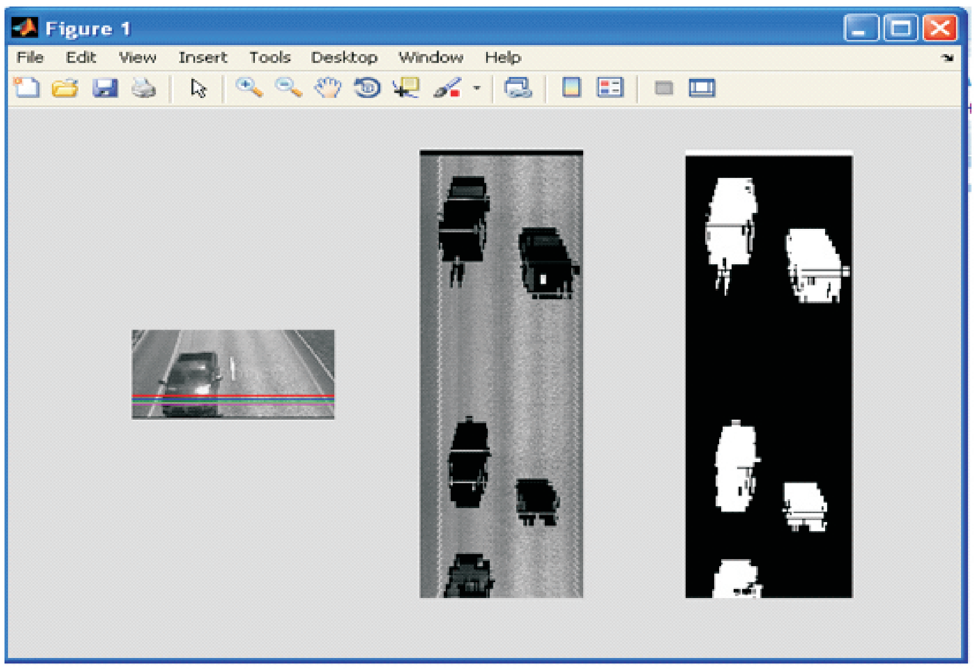


Figure :7.2 Cropping of TOBs for classification of vehicles

Matlab simulated output showing vehicle detection using consecutive neighbouring frame difference method is shown in figure

In this system, the detected vehicle regions will be classified as car and truck or bus. After vehicle tracking, the system considers four parameters for classifying the vehicles. Those four parameters are aspect ratio and compactness C which are respectively defined as

A suitable value of k is empirically obtained by computing the percentage error in classification for each of the four classes, viz., 2W, 3W,4W, and 6W.

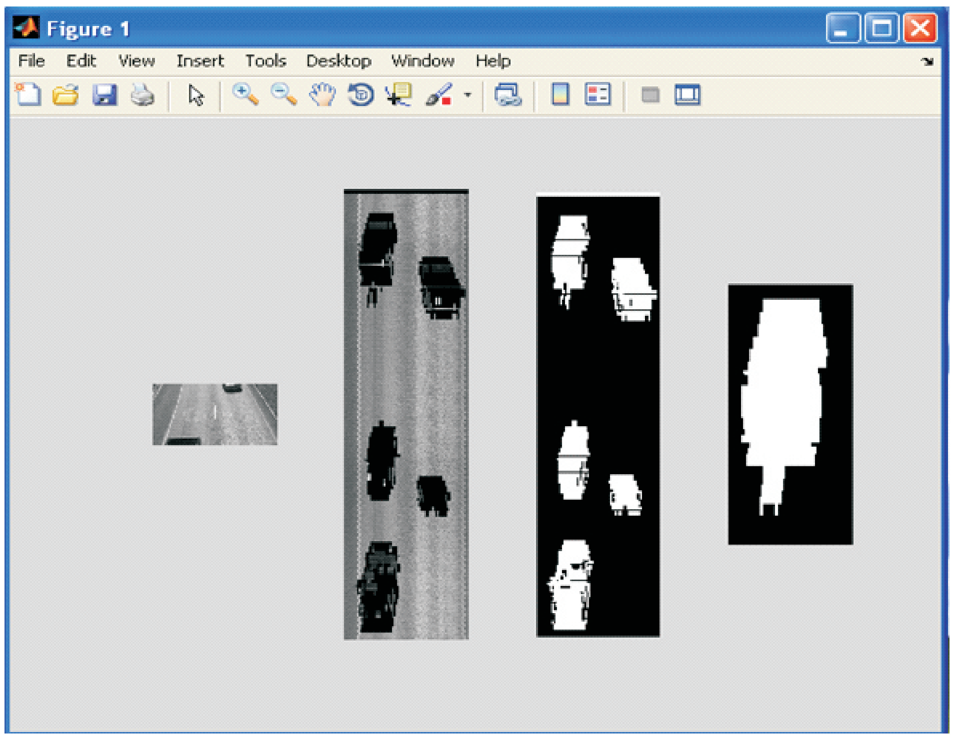


Figure : 7.3 Matlab simulated output showing vehicle detection using neighbouring frame difference method

The TSIs object blob obtained from multiple TSIs of a video. Then calculate the features of TSI object blob of area, centroid, rectangularity, ratio. After obtain the values are compared with the trained data set values then finally by using the maximum voting scheme the type of the classifier is obtain.

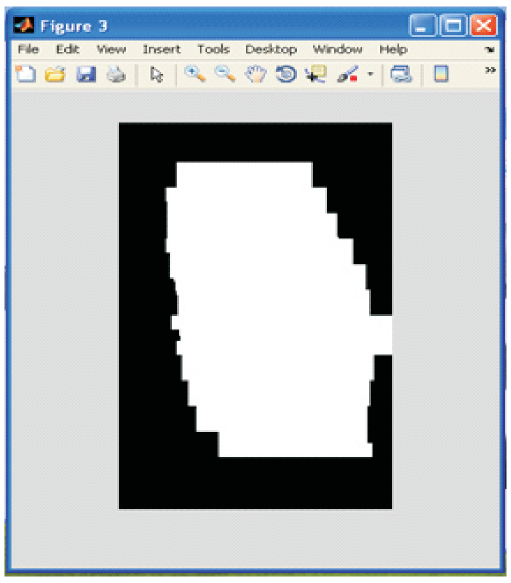


Figure (a):TOB of TSI

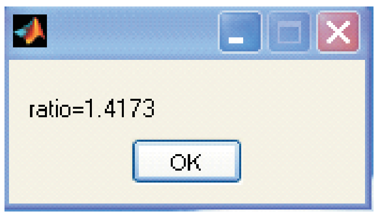


Figure (b) :Ratio

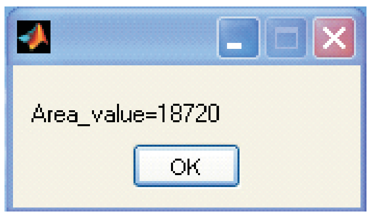


Figure (c):Area

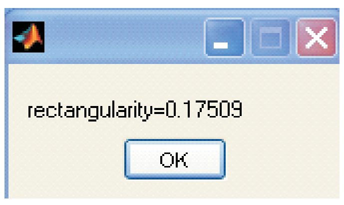


Figure (d):Rectangularity

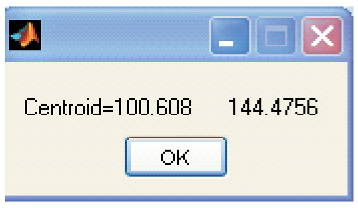


Figure (e):Centroid

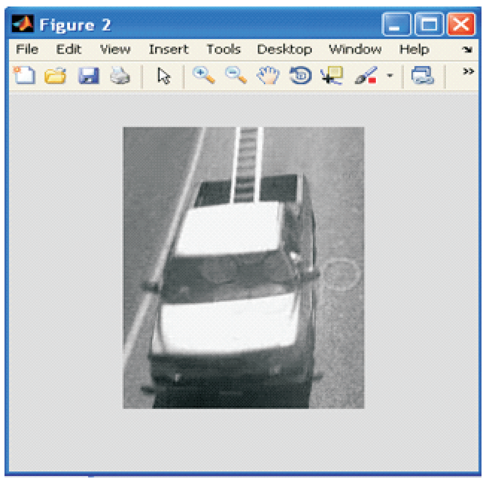


Figure (f):classified vehicle is truck

Figure :7.4 The classifier is truck obtained after consecutive neighbouring frame difference method.

The TSIs object blob obtained from multiple TSIs of a video. Then calculate the features of TSI object blob of area, centroid, rectangularity, ratio. After obtain the values are compared with the trained data set values then finally by using the maximum voting scheme the type of the classifier is obtain. The resulting classifier is car.The result specifies the vehicle is car belongs to class (4w) four wheeler.

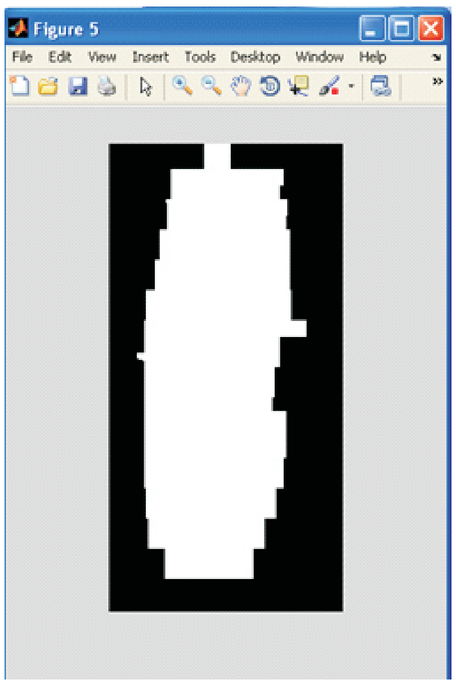
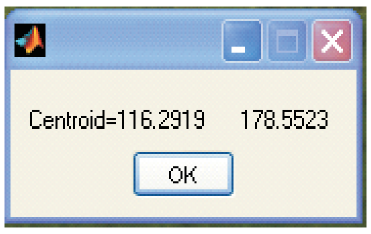


Figure (g):TOB of TSI



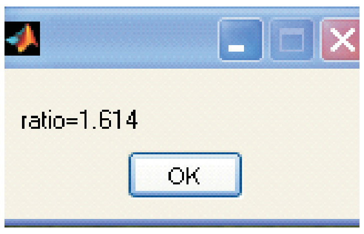
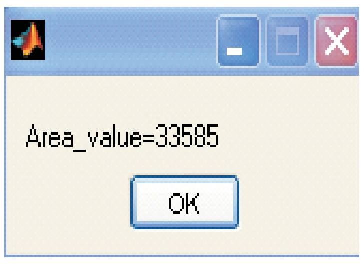


Figure (1):Centroid Figure (2):Ratio



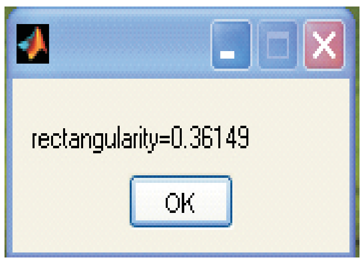


Figure (3):Area Figure (4):Rectangularity

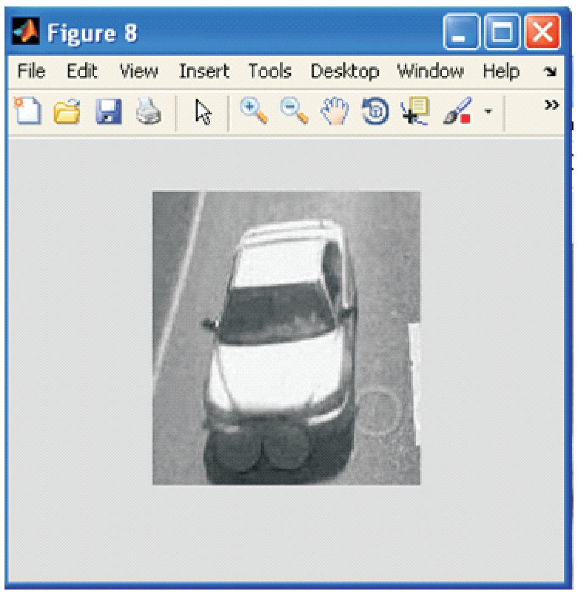


Figure (f):classified vehicle is car
Figure : 7.5Matlab simulated output showing vehicle detection using consecutive neighbouring difference method.

The TSIs object blob obtained from multiple TSIs of a video. After obtain the values are compared with the trained data set values then finally by using the maximum voting scheme the type of the classifier is obtain. The resulting classifier is van. . The resulting classifier is car.The result specifies the vehicle is car belongs to class (4w) four wheeler.

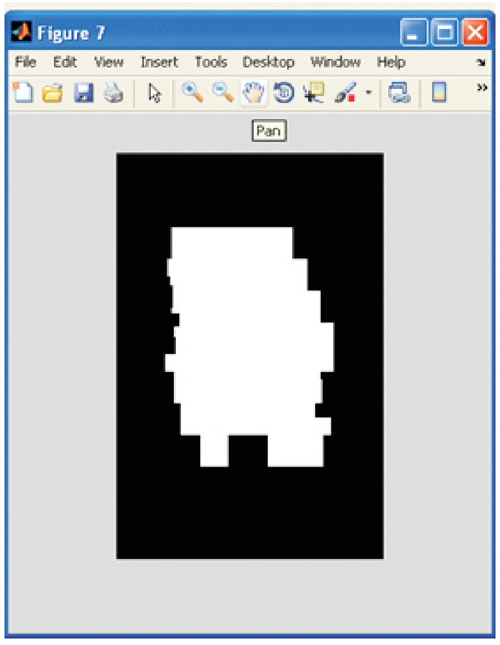


Figure (5):TOB of TSI

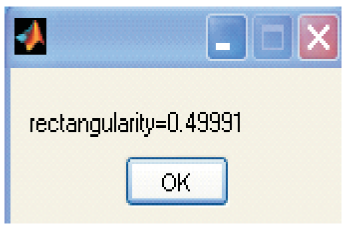


Figure (1):Rectangularity

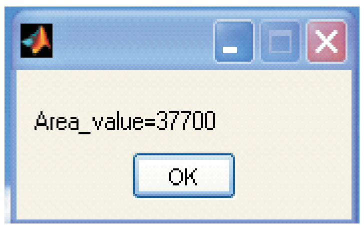


Figure (2):Area

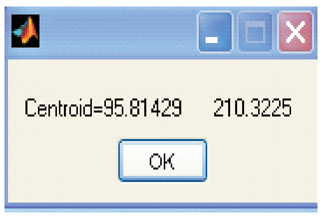


Figure (3):Centroid

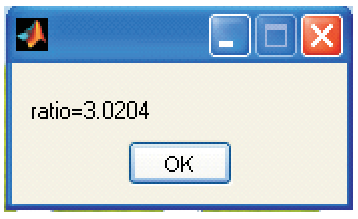


Figure (4):Rectangularity

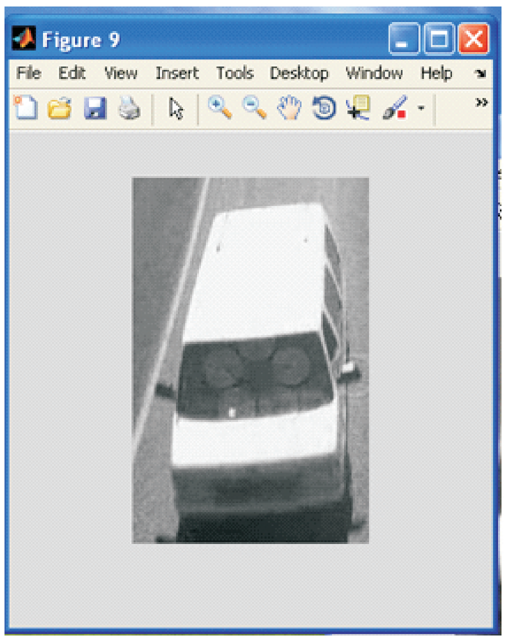


Figure (5):classified vehicle is van

The TSIs object blob obtained from multiple TSIs of a video. After obtain the values are compared with the trained data set values then finally by using the maximum voting scheme the type of the classifier is obtain. The resulting classifier is van. The resulting classifier is car.The result specifies the vehicle is car belongs to class (4w) four wheeler.

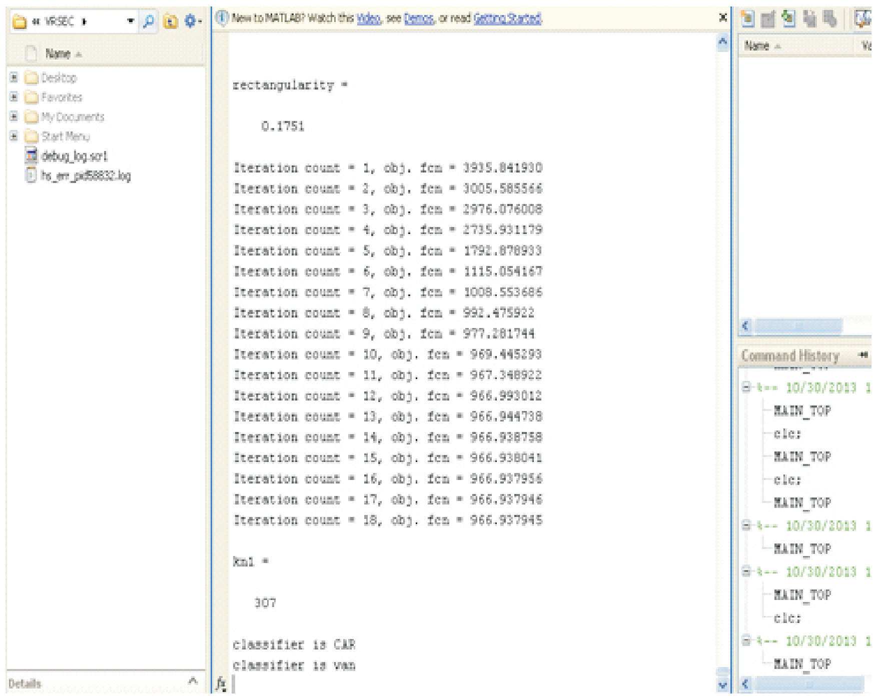


Figure : 7.7 Matlab simulated output showing vehicle detection using consecutive neighbouring difference method

The results obtained after using consecutive neighbouring frame difference method to the second video is shown in figure

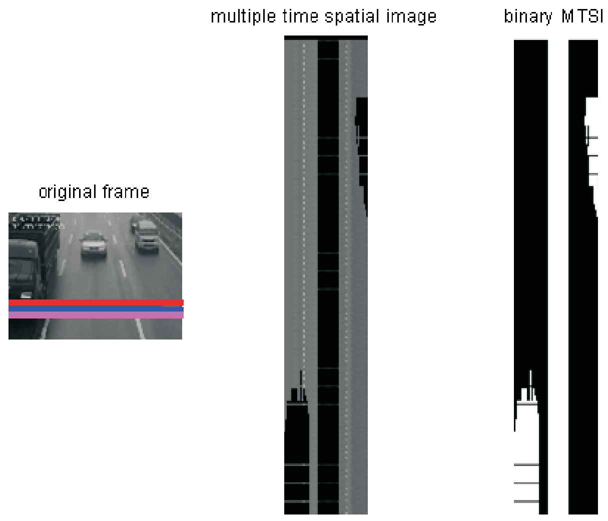


Figure: 7.8 The Matlab simulated output showing vehicle detection using consecutive neighbouring difference method for the second video.

The TSIs object blob obtained from multiple TSIs of a video. Then calculate the features of TSI object blob of area, centroid, rectangularity, ratio. After obtain the values are compared with the trained data set values then finally by using the maximum voting scheme the type of the classifier is obtain. The resulting classifier is car.The result specifies the vehicle is car belongs to class (4w) four wheeler.

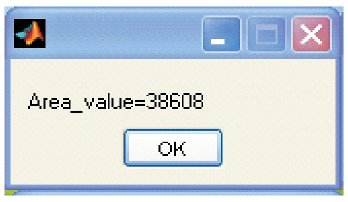
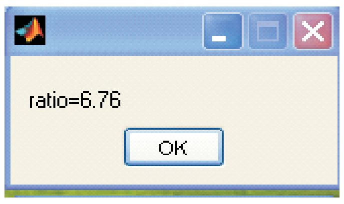
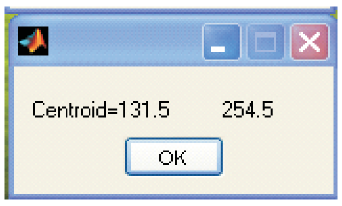


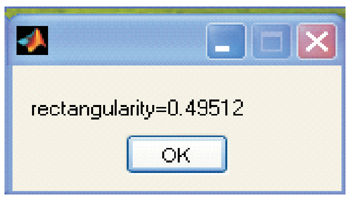
Figure (a):Area;



Figure(b):Ratio



Figure(c):Centroid



Figure(d):Rectangularity

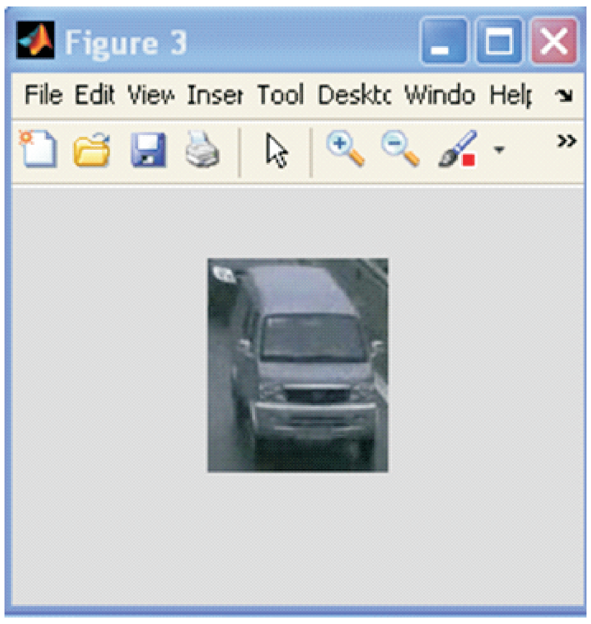


Figure: 7.9 The Matlab simulated output showing vehicle detection using CNFD

The TSIs object blob obtained from multiple TSIs of a video. Then calculate the features of TSI object blob of area, centroid, rectangularity, ratio..The result specifies the vehicle is car belongs to class (4w) four wheeler.

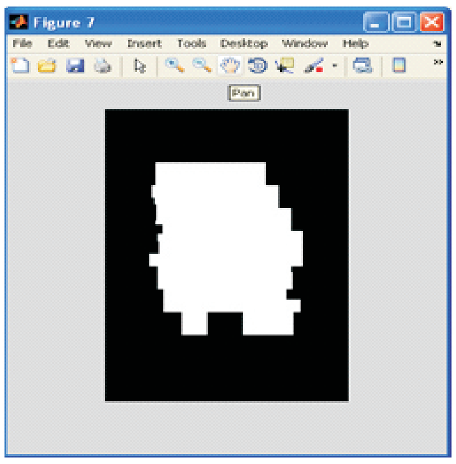
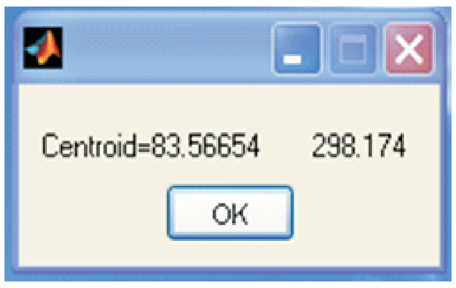
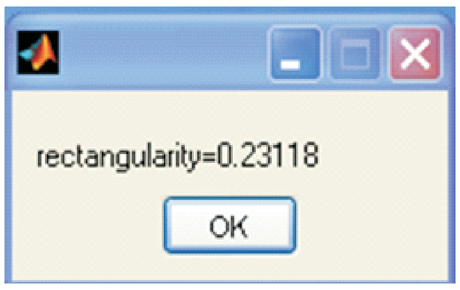


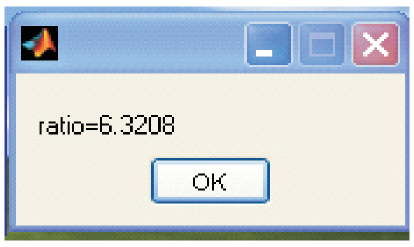
Figure (a):TOB of TSI



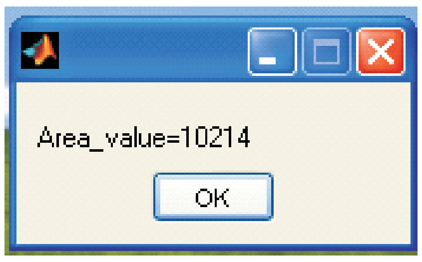
Figure(7.10.1):Centroid



Figure(7.10.2):Rectangularity



Figure(7.10.3):Ratio



Figure(7.10.4):Area

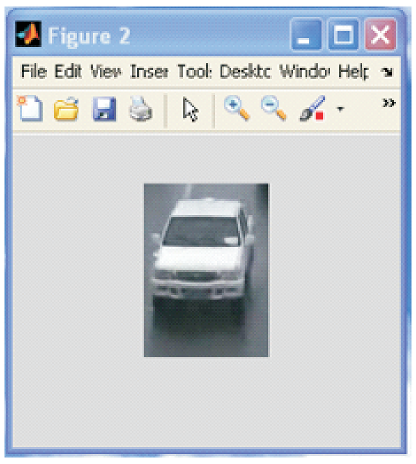


Figure (f):classified vehicle is car

8 CONCLUSION AND FUTURE SCOPE

Vehicle classification is used in ETC for collecting the appropriate toll and even to reduce the time at toll plazas. This project presents a robust traffic surveillance system for vehicle detection and classification. A new method has been implemented for detecting and classifying vehicles from different video sequences. The algorithm implemented in this project classifies the vehicles as per the shape based features like aspect ratio and compactness. The usage of consecutive neighbouring frame difference method ensures the reduction in complexity while detecting the moving objects. A merging of TSIs object blobs from multiple virtual detection lines implemented using morphological operation reduces the occlusions among the vehicles and reduction of time to minimum level. The results obtained after implementing the developed technique during different weather conditions are of same accuracy. This technique is sensitive to variations in illumination.

However the performance of this system is significantly affected by the selected thresholds. In the near future, Gaussian mixture methods can be used to detect the video object detected and find the vehicle parameter estimation.

REFERENCES

[1] K. Park, D. Lee, and Y. Park, "Video-based detection of street-parking violation," in Proc. Int. Conf. Image Process., Comput. Vis., PatternRecognit., Las Vegas, NV, 2007, vol. 1, pp. 152–156.
[2] H. Yalcin, M. Herbert, R. Collins, and M. J. Black, "A flow-based approach to vehicle detection and background mosaicking in airborne video," in Proc. CVPR, San Diego, CA, 2005, vol. 2, p. 1202.
[3] H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos, "Vision-based monitoring of intersections," in Proc. IEEE 5th Int. Conf. Intell. Transp. Syst., Singapore, 2002, pp. 7–12.
[4] Z. Zhang, Y. Cai, K. Huang, and T. Tan, "Real-time moving object classification with automatic scene division," in Proc. IEEE Int. Conf. Image Process., San Antonio, TX, 2007, vol. 5, pp. V-149–V-152.
[5] P. G. Michalopoulos, "Vehicle detection video through image processing: The autoscope system," IEEE Trans. Veh. Technol., vol. 40, no. 1, pp. 21–29, Feb. 1991.
[6] E. Rivlin, M. Rudzsky, M. Goldenberg, U. Bogomolov, and S. Lapchev, "A real-time system for classification of moving objects," in Proc. Int. Conf. Pattern Recognit., Quebec City, QC, Canada, 2002, vol. 3, pp. 688–691.
[7] L. Xie, G. Zhu, Y. Wang, H. Xu, and Z. Zhang, "Robust vehicles extraction in a video-based intelligent transportation system," in Proc. IEEE Int. Conf. Commun., Circuits Syst., Hong Kong, China, 2005, vol. 2,pp. 887–890.
[8] J. Cao and L. Li, "Vehicle objects detection of video images based on gray-scale characteristics," in Proc. IEEE Int. Conf. Educ. Technol. Comput.Sci., Wuhan, Hubei, China, 2009, pp. 936–940
[9] B. T.Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," IEEE Trans. Intell. Transp. Syst., vol. 9, no. 3, pp. 425–437, Sep. 2008
[10] Z. Zhang, W. Dong, K. Huang, and T. Tan, "EDA approach for model based localization and recognition of vehicles," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Minneapolis, MN, 2007, pp. 1–8.
[11] A. Wedel, T. Schoenemann, T. Brox, and D. Cremers, "WarpCut—Fast obstacle segmentation in monocular video," in Proc. 29th DAGM Conf. Pattern Recognit., Berlin, Heidelberg, Germany, 2007, pp. 264–273.
[12] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," IEEE Trans. Intell. Transp. Syst., vol. 3, no. 1, pp. 37–47, Mar. 2002.
[13] S. Wender and K. Dietmayer, "3D vehicle detection using a laser scanner and a video camera," IET Intell. Transp. Syst., vol. 2, no. 2, pp. 105–112, Jun. 2008.
[14] L. Anan, Y. Zhaoxuan, and L. Jintao, "Video vehicle detection algorithm based on virtual-line group," in Proc. IEEE Asia Pacific Conf. Circuits Syst., Singapore, 2006, pp. 1148–1151.
[15] J.Wu, Z. Yang, J.Wu, and A. Liu, "Virtual line group based video vehicle detection algorithm utilizing both luminance and chrominance," in Proc. IEEE Conf. Ind. Electron. Appl., Harbin, China, 2007, pp. 2854–2858.
[16] L. Li, L. Chen, X. Huang, and J. Huang, "A traffic congestion estimation approach from video using time-spatial imagery," in Proc. Int. Conf. Intell. Netw. Intell. Syst., Wuhan, China, 2008, pp. 465–469.
[17] Y. Hue, "A traffic-flow parameters evaluation approach based on urban road video," Int. J. Intell. Eng. Syst., vol. 2, no. 1, pp. 33–39, 2009.
[18] C. L. Huang and W. C. Liao, "A vision-based vehicle identification system," in Proc. 17th Int. Conf. Pattern Recognit., Cambridge, U.K., 2004, pp. 364–367.
[19] A. J. Lipton, H. Fujioshi, and R. Patil, "Moving target classification and tracking for real time video," in Proc. IEEE Workshop Appl. Comput. Vis., Princeton, NJ, 1998, pp. 8–14.
[20] L. Eikvil, L. Aurdal, and H. Koren, "Classification-based vehicle detection in high-resolution satellite images," ISPRS J. Photogramm. Remote Sens., vol. 64, no. 1, pp. 65–72, Jan. 2009.
[21] D. J. Dailey, F. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," IEEE Trans. Intell. Transp. Syst., vol. 1, no. 2, pp. 98–107, Jun. 2000.
[22] E. M. Kornaropoulos and P. Tsakalides, "A novel KNN classifier for acoustic vehicle classification based on alpha-stable statistical modeling," in Proc. IEEE Workshop Stat. Signal Process., Cardiff, U.K., 2009, pp. 1–4.
[23] Z. Chen, N. Pears, M. Freeman, and J. Austin, "Road vehicle classification using support vector machines," in Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst., Shanghai, China, 2009, pp. 214–218.

[24] A. Goyal and B. Verma, “A neural network based approach for the vehicle classification,” in Proc. IEEE Symp. C

[25] N. U. Rashid, N. C. Mithun, B. R. Joy, and S. M. M. Rahman, “Detection and classification of vehicles from a video using time-spatial image,” in Proc. 6th Int. Conf. Elect. Comput. Eng., Dhaka, Bangladesh, 2010, pp. 502–505

[26] Niluthpol Chowdhury Mithun, Nafi Ur Rashid, and S. M. Mahbubur Rahman,“Detection and classification of vehicles from a video using multiple time-spatial image,” in Proc. 6th Int. Conf. Elect. Comput. Eng., Dhaka, Bangladesh, 2012, pp. 502–505