**CHAPTER 1**

**INTRODUCTION**

**1.1 SCOPE OF THE PROJECT**

Traffic congestion has become a significant problem in recent years. The main reason behind it is the increase in the population in large cities and respective increase in number of vehicles. The traffic jams not only affect the human routine lives but also lead to a rise in the cost of transportation. Therefore an automated traffic system is required to manage the traffic congestion problem smoothly.

To unravel this problem, the government should encourage people to use public transport or vehicles with small size such as bicycles or make tax on personal vehicles. Particularly, in some Asian countries such as Viet Nam, the local authorities passed law limiting to the number of vehicles for each family. The methods mentioned above is really efficient in fact. That the inadequate infrastructure cannot handle the issue of traffic is also a decisive reason. The public conveyance is available and its quality is very bad, mostly in the establishing countries. Besides, the highway and roads are incapable of meeting the requirement of increasing number of vehicle. Instead of working on roads to accommodate the growing traffic various techniques have been devised to control the traffic on roads like embedded controllers that are installed at the junction. These techniques are briefly described in next section.

* + 1. **Manual Controlling**

Manual controlling the name instance it require man power to control the traffic. Depending on the countries and states the traffic polices are allotted for a required area or city to control traffic. The traffic polices will carry sign board, sign light and whistle to control the traffic. They will be instructed to wear specific uniforms in order to control the traffic.

**1.1.2 Automatic Controlling**

Automatic traffic light is controlled by timers and electrical sensors. In traffic light each phase a constant numerical value loaded in the timer. The lights are automatically getting ON and OFF depending on the timer value changes. While using electrical sensors it will capture the availability of the vehicle and signals on each phase, depending on the signal the lights automatically switch ON and OFF.

**1.1.3 Drawbacks:**

In the manual controlling system we need more man power. As we have poor strength of traffic police we cannot control traffic manually in all area of a city or town. So we need a better solution to control the traffic. On the other side, automatic traffic controlling a traffic light uses timer for every phase. Using electronic sensors is another way in order to detect vehicles, and produce signal that to this method the time is being wasted by a green light on an empty road. Traffic congestion also occurred while using the electronic sensors for controlling the traffic. All these drawbacks are supposed to be eliminated by using image processing.

* 1. **OBJECTIVE OF THE SYSTEM**

We propose a system for controlling the traffic light by image processing. The vehicles are detected by the system through images instead of using electronic sensors embedded in the pavement. A camera will be placed alongside the traffic light. It will capture image sequences. Image processing is a better technique to control the state change of the traffic light. It shows that it can decrease the traffic congestion and avoids the time being wasted by a green light on an empty road. It is also more reliable in estimating vehicle presence because it uses actual traffic images. It visualizes the practicality, so it functions much better than those systems that rely on the detection of the vehicles’ metal content.

**1.3 THESIS ORGANIZATION**

Many techniques have been developed in Image Processing during the last four to five decades. Most of the methods are developed for enhancing images obtained from unmanned space probes, spacecraft and military reconnaissance flights. Image Processing systems are becoming widely popular due to easy availability of powerful personnel computers, large memory devices, graphics software and many more. Image processing involves issues related to image representation, compression techniques and various complex operations, which can be carried out on the image data. The operations that come under image processing are image enhancement operations such as sharpening, blurring, and brightening, edge enhancement. Traffic density of lanes is calculated using image processing which is done of images of lanes that are captured using digital camera. We have chosen image processing for calculation of traffic density as cameras are very much cheaper than other devises such as sensors. Making use of the above mentioned virtues of image processing we propose a technique that can be used for traffic control.

Traffic control system is based on four main steps.

1. Image Acquisition

2. Image Pre-Processing

3. Density Calculation

4. Traffic Control

**1. Image Acquisition**

Generally an image is a two-dimensional function f(x,y)(here x and y are plane coordinates).The amplitude of image at any point say f is called intensity of the image. It is also called the gray level of image at that point. We need to convert these x and y values to finite discrete values to form a digital image. The input image is a fundus taken from stare data base and drive data base. The image of the retina is taken for processing and to check the condition of the person. We need to convert the analog image to digital image to process it through digital computer. Each digital image composed of a finite elements and each finite element is called a pixel.

**1.1 Formation of Image**

We have some conditions for forming an image f(x,y) as values of image are proportional to energy radiated by a physical source. So f(x,y) must be nonzero and finite. i.e. 0< f(x,y) < ∞.

**2. Image Pre-Processing**

Pre-processing is done to get a clear image. Since the images are extracted from real time video frames the images can be distorted or blurred or dark etc. like images can be blurred when the weather is foggy or rainy. Similarly, images can be darker when captured at night time conditions or can be too bright when it’s very sunny (like in afternoon).Therefore different pre-processing methods are applied on the images to improve the quality of the image, according to the objective of the user.

**2.1 Image Resizing/Scaling**

Image scaling occurs in all digital photos at some stage whether this be in Bayer demo sizing or in photo enlargement. It happens anytime you resize your image from one pixel grid to another. Image resizing is necessary when you need to increase or decrease the total number of pixels. Even if the same image resize is performed, the result can vary significantly depending on the algorithm.

Images are resized because of number of reasons but one of them is very important in our project. Every camera has its resolution, so when a system is designed for some camera specifications it will not run correctly for any other camera depending on specification similarities, so it is necessary to make the resolution constant for the application and hence perform image resizing.

**2.2 RGB to GRAY Conversion**

Humans perceive color through wavelength-sensitive sensory cells called cones. There are three different varieties of cones, each has a different sensitivity to electromagnetic radiation (light) of different wavelength. One cone is mainly sensitive to green light, one to red light, and one to blue light. By emitting a restricted combination of these three colors (red, green and blue), and hence stimulate the three types of cones at will, we are able to generate almost any detectable color. This is the reason behind why color images are often stored as three separate image matrices; one storing the amount of red (R) in each pixel, one the amount of green (G) and one the amount of blue (B). We call such color images as stored in an RGB format. In grayscale images, however, we do not differentiate how much we emit of different colors, we emit the same amount in every channel.

We will be able to differentiate the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels. When converting an RGB image to grayscale, we have to consider the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel. One of the approaches is to take the average of the contribution from each channel: (R+B+C)/3.

However, since the perceived brightness is often dominated by the green component, a different, more "human-oriented", method is to consider a weighted average, e.g.: 0.3R + 0.59G + 0.11B.

Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further analysis. For example, we can eliminate noise, which will make it easier to identify the key characteristics.

In poor contrast images, the adjacent characters merge during binarization. We have to reduce the spread of the characters before applying a threshold to the word image.

Hence, we introduce “**POWER- LAW TRANSFORMATION**” which increases the contrast of the characters and helps in better segmentation. The basic form of power-law transformation is

Where **r** and **s** are the input and output intensities, respectively; c and γ are positive constants. A variety of devices used for image capture, printing, and display respond according to a Power Law. By convention, the exponent in the power-law equation is referred to as gamma. Hence, the process used to correct these power-law response phenomena is called gamma correction. Gamma correction is important, if displaying an image accurately on a computer screen is of concern.

In our experimentation, γ is varied in the range of 1 to 5. If c is not equal to ’1’, then the dynamic range of the pixel values will be significantly affected by scaling. Thus, to avoid another stage of rescaling after power-law transformation, we fix the value of c = 1.With γ = 1, if the power-law transformed image is passed through binarization, there will be no change in the result compared to simple binarization. When γ > 1, there will be a change in the histogram plot, since there is an increase of samples in the bins towards the gray value of zero. Gamma correction is important if displaying an image accurately on computer screen is of concern.

**3. Density Calculation**

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more technically, has discontinuities or noise. The points at which image brightness alters sharply are typically organized into a set of curved line segments termed edges.

The same problem of detecting discontinuities in 1D signal is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a basic tool in image processing, machine vision and computer envisage, particularly in the areas of feature reveal and feature extraction.

**3.1 Edge detection techniques**

Different colors has different brightness values of particular color. Green image has more bright than red and blue image or blue image is blurred image and red image is the high noise image. Following are list of various edge-detection methods:

• Sobel Edge Detection Technique

• Prewitt Edge Detection

• Roberts Edge Detection Technique

• Zero cross Threshold Edge Detection Technique

• Canny Edge Detection Technique

In our project we use “**CANNY EDGE DETECTION TECHNIQUE**” because of its various advantages over other edge detection techniques.

**3.1.1 Canny Edge Detection**

The Canny Edge Detector is one of the most commonly used image processing tools detecting edges in a very robust manner. It is a multi-step process, which can be implemented on the GPU as a sequence of filters. Canny edge detection technique is based on three basic objectives.

**I. Low error rate:** All edges should be found, and there should be no spurious responses. That is, the edges must be as close as possible to the true edges.

**II. Edge point should be well localized:** The edges located must be as close as possible to the true edges. That is, the distance between a point marked as an edge by the detector and the center of the true edge should be minimum.

**III. Single edge point response:** The detector should return only one point for each true edge point. That is, the number of local maxima around the true edge should be minimum. This means that the detector should not identify multiple edge pixels where only a single edge point exist.

The essence of Canny’s work was in expressing the preceding three criteria mathematically and then attempting to find optimal solution to these formulations, in general, it is difficult to find a close form solution that satisfies all the preceding objectives. However, using numerical optimization with 1-D step edges corrupted by additive while Gaussian noise led to the conclusion that a good approximation to the optimal step edge detector is the first derivative of Gaussian:

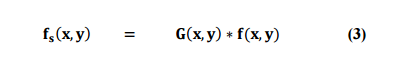
****

Generalizing this result to 2-D involves recognizing that the 1-D approach still applies in the direction of the edge normal. Because the direction of the normal is unknown beforehand, this would require applying the 1-D edge detector in all possible directions. This task can be approximated by first smoothing the image with circular 2-D Gaussian function, computing the gradient of the result, and then using the gradient magnitude and direction to estimate edge strength and direction at every point.

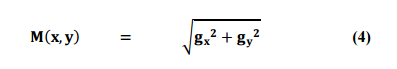
Let f(x,y) denote the input image and G(x,y) denote the Gaussian function:

****

We form a smoothed image, fs(x, y), by convolving G and f:



This operation is followed by computing the gradient and direction (angle)



And

****

With gx = 𝛛𝐟𝐬 𝛛𝐱 and gy = 𝝏𝒇𝒔 𝝏𝒚

Equation (2) is implemented using an n x n Gaussian mask. Keep in mind that M(x, y) and 𝛼(x, y) are arrays of the same size as the image from which they are computed. Because it is generated using the gradient M(x, y) typically contains wide ridges around local maxima.

The next step is to thin those ridges. One approach is to use non maxima suppression. This can be done in several ways, but the essence of the approach is to specify a no. of discrete orientations of edge normal (gradient vector). For example, in 3x3 region we can define four orientation for an edge passing through the center point of the region: horizontal, vertical, +45° and -45°.

The final operation is to threshold gN(x, y) to reduce false edge point. We do it by using a single threshold, in which all value below the threshold were set to 0. If we set the threshold too low, there will still be some false edge (called false positives). If the threshold id set too high, then actual valid edge points will be eliminated (false negatives). Canny’s algorithm attempts to improve on this situation by using hysteresis threshold. We use two threshold,

A low threshold, TL, and a high threshold, TH. Canny suggested that the ratio of high to low threshold should be two or three to one.

We visualize the thresholding operation as creating two additional images

𝐠𝐍(𝐱, 𝐲) = 𝐠𝐍(𝐱, 𝐲) ≥ **𝐓𝐇 (6)**

And 𝐠𝐍(𝐱, 𝐲) = 𝐠𝐍(𝐱, 𝐲) ≥ 𝐓𝑳 **(7)**

Where, initially, both gNH(x, y) 𝑎𝑛𝑑 gNL(x, y) are set to 0. After thresholding, gNH(x, y) will have fewer nonzero pixels than gNL(x, y) in general, but all the nonzero pixels in gNH(x, y) will be contained in gNL(x, y) because the latter image is formed with lower threshold. We eliminate from gNL(x, y) all the nonzero from gNH(x, y) by letting

𝐠𝐍(𝐱, 𝐲) = 𝐠𝐍𝐋(𝐱, 𝐲) − 𝐠𝐍𝐇(𝐱, 𝐲) **(8)**

The nonzero pixels in gNL(x, y) and gNH(x, y) may be viewed as being “strong”. And “weak” edge pixels, respectively. After the thresholding operations, all strong pixels in gNH(x, y) are assumed to be valid edge pixels and are so marked immediately.

Depending on the value of TH, the edges in gNH(x, y) typically have gaps. Longer edges are formed using the following procedure

a. Locate the next unvisited pixel p in gNH(x, y).

b. Mark as valid edge pixels all the weak pixels in gNL(x, y) that are connected to p using say 8 connectivity.

c. If all nonzero pixels in gNH(x, y) have been visited go to step d. else return to step a.

d. Set to zero all pixels in gNL(x, y) that were not marked as valid edge pixels. At the end of this procedure, the final image output by the Canny is formed by appending to gNH(x, y) all the nonzero pixels from gNl(x, y).

We use two additional images, gNH(x, y) and gNl(x, y), to simplify the discussion. In practice, hysteresis threshold can be implemented directly during non-maxima suppression, and thresholding can be implemented directly on gNL(x, y) by forming a list of strong pixels and the weak pixels connected to them.

Summarizing, the canny edge detection algorithm consist of the following basic steps;

1. Smooth the input image with Gaussian filter.
2. Compute the gradient magnitude and angle images.
3. Apply non-maxima suppression to the gradient magnitude image.
4. Use double thresholding and connectivity analysis to detect and link edges.

Later, the number of vehicles are counted, also known as **Density** of the road.

**4. Traffic Control**

The traffic density/traffic count calculation helps in automatic switching of traffic signals, based on the number of vehicles present at any particular lane at any instance of time.

**USEFUL DEFINITIONS**

An **image** is a picture representing visual information. A **digital image** is an image that can be stored in digital form.

**Digital image processing** is a field of study that seeks to analyze, process, or enhance a digital image to achieve some desired outcome.

**Biometrics** is the study of automated identification by use of physical and behavioral traits.

**Pixel** is the smallest element of an image. Each pixel correspond to any one value. In an 8-bit gray scale image, the value of the pixel between 0 and 255. The value of a pixel at any point correspond to the intensity of the light photons striking at that point. Each pixel store a value proportional to the light intensity at that particular location.

The **binary image** as it name states, contain only two pixel values. 0 and 1.

In **pixel resolution**, the term resolution refers to the total number of count of pixels in a digital image. For example. If an image has M rows and N columns, then its resolution can be defined as M X N.

**Aspect ratio** is the ratio between width of an image and the height of an image. It is commonly explained as two numbers separated by a colon (8:9). This ratio differs in different images, and in different screens.

**Zooming** simply means enlarging a picture in a sense that the details in the image became more visible and clear. Zooming an image has many wide applications ranging from zooming through a camera lens, to zoom an image on internet etc.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 IMAGE PROCESSING BASICS**

**Image processing** is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

 1. Importing the image with optical scanner or by digital photography.

 2. Analysing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.

 3. Output is the last stage in which result can be altered image or report that is based on image analysis.

The purpose of image processing is divided into 5 groups. They are:

1.      Visualization - Observe the objects that are not visible.

2.      Image sharpening and restoration - To create a better image.

3.      Image retrieval - Seek for the image of interest.

4.      Measurement of pattern – Measures various objects in an image.

5.      Image Recognition – Distinguish the objects in an image

The two types of methods used for Image Processing are Analog and Digital Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

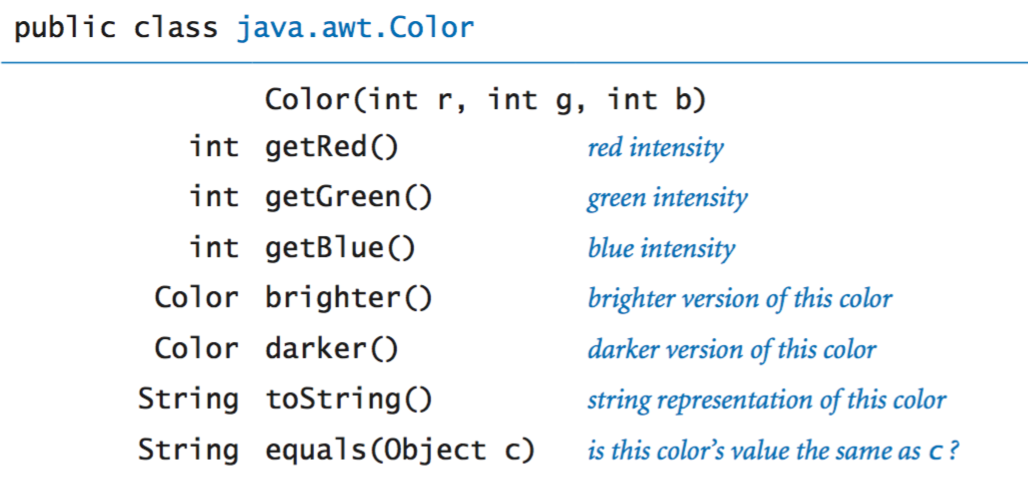
## COLOR

Java's [Color](https://docs.oracle.com/javase/8/docs/api/java/awt/Color.html) data type represents color values using the [RGB color model](https://en.wikipedia.org/wiki/RGB_color_model) where a color is defined by three integers (each between 0 and 255) that represent the intensity of the red, green, and blue components of the color. Other color values are obtained by mixing the red, blue and green components.

The Color data type has a constructor that takes three integer arguments. For example, you can write

|  |  |
| --- | --- |
| Color red = new Color(255, 0, 0);  Color blue = new Color( 9, 90, 166); |  |

The following table summarizes the methods in the [Color](https://docs.oracle.com/javase/8/docs/api/java/awt/Color.html) API



**BUFFEREDIMAGE**

Java BufferedImage class is a subclass of Image class. It is used to handle and manipulate the image data. A BufferedImage is made of ColorModel of image data. All BufferedImage objects have an upper left corner coordinate of (0, 0).

This class supports three types of constructors.

The first constructor constructs a new BufferedImage with a specified ColorModel and Raster.

BufferedImage(ColorModel cm, WritableRaster raster, boolean isRasterPremultiplied, Hashtable<?,?> properties)

The second constructor constructs a BufferedImage of one of the predefined image types.

BufferedImage(int width, int height, int imageType)

The third constructor constructs a BufferedImage of one of the predefined image types: TYPE\_BYTE\_BINARY or TYPE\_BYTE\_INDEXED.

BufferedImage(int width, int height, int imageType, IndexColorModel cm)

Methods and Description.

|  |  |
| --- | --- |
| S.No | Methods |
| 1 | copyData(WritableRaster outRaster)  It computes an arbitrary rectangular region of the BufferedImageand copies it into a specified WritableRaster. |
| 2 | getColorModel()  It returns object of class ColorModel of an image. |
| 3 | getData()  It returns the image as one large tile. |
| 4 | getData(Rectangle rect)  It computes and returns an arbitrary region of the BufferedImage. |
| 5 | getGraphics()  This method returns a Graphics2D, retains backwards compatibility. |
| 6 | getHeight()  It returns the height of the BufferedImage. |
| 7 | getMinX()  It returns the minimum x coordinate of this BufferedImage. |
| 8 | getMinY()  It returns the minimum y coordinate of this BufferedImage. |
| 9 | getRGB(int x, int y)  It returns an integer pixel in the default RGB color model (TYPE\_INT\_ARGB) and default sRGB colorspace. |
| 10 | getType()  It returns the image type. |

**2.2 Traffic Image Pre-processing Techniques**

Pre-processing is done to get a clear image. Since the images are extracted from real time video frames the images can be distorted or blurred or dark etc. like images can be blurred when the weather is foggy or rainy. Similarly, images can be darker when captured at night time conditions or can be too bright when it’s very sunny (like in afternoon).Therefore different pre-processing methods are applied on the images to improve the quality of the image, according to the objective of the user. In static background subtraction method the images captured are first converted into grayscale. Later the grayscale is converted into binary. After that erosion and dilation are applied according to requirement for a clearer image. On the other hand, in canny edge detection method first the image is converted to grayscale and then every frame is applied with background subtraction for object detection.

In a self-proposed algorithm in first the image is converted to grayscale and then further processed for density calculation.

Another method involving dual technique combination proposes to use grayscale. Conversion along with background subtraction to detect foreground objects on a fixed background.

In the gradient based image enhancement method the author makes use of gamma correction for image enhancement.

Pre-processing plays an important role as the images are captured from the live videos so they can be affected by the surrounding conditions of the road. The images can be blurred, distorted, very bright or very dark etc. So pre-processing helps to improve the quality of the image that further helps in better analysis of the image and traffic density calculation also. Following figures show some pre-processing outputs:

****

Figure1: Image of road after applying background

Subtraction and wiener filter

****

Figure1: Image of road after applying closing

Operation

**2.3 Density calculation/Vehicle count**

In the background subtraction technique a combination of motion detection and vehicle detection is used. For motion detection, analysis of two consecutive frames is taken into account, in which the histogram of key region parts of the frames is analysed. The histogram is then compared with the determined threshold.

A constraint stated with this method is that the key region should be at least 3-pixel wide profile of the image along the road.

The difference between these profiles then shows the displacement or motion of the object. For vehicle detection, the image of the road is divided into subparts. Then background subtraction technique is used.

In the Canny edge detection method an adaptive background subtraction is used. After that, canny edge detection method is applied for edge detection of the vehicle which will detect all the edges of the vehicles present in the image. Canny edge detector may prove to be effective as it considers all neighbourhood pixels while detecting edges. For object detection, Moore neighbourhood algorithm is used along with the Jacob’s criterion.

This method is supposed to give better results as compared to static background subtraction. In another self-proposed method the author has proposed to calculate the density of vehicle traffic rather than calculation the number of vehicles. This means, for instance, the vehicle density of a truck could be equivalent to two medium sized cars. This method proposes to be better than counting number of vehicles.

The reason is that, counting number of vehicles may be a problem when image has different types of vehicles, like car, bicycle etc.

But calculating density of each vehicle will consider all types of vehicles in traffic.

The author also proposed a formula for the same

***C = h\*no. of rows in subtracted***

***Image\*no. of columns in subtracted***

***Image\*no. of frames per second***

Where **h=height of camera from road**

In the Dual method technique the author uses a combination of gradient magnitude and direct subtraction techniques to detect vehicles present on the lanes. The reason for using these two techniques simultaneously is that each of them overcomes the disadvantage of the other.

In direct subtraction method colour of vehicle can be problematic in finding density. This problem is resolved using gradient magnitude. While in gradient magnitude method, there can be situations where detected edges may not form closed contour. This problem is resolved using direct subtraction.

In the Gradient method technique the author proposes to use edge detection for making vehicle count. Edge detection in this method is done using canny edge detector and gradient based edge detection.

According to the review done, the self-proposed method gives the most explanatory and proving results as compared to all other methods discussed in this report.

Following are the comparison of all methods that can be used in this system.

|  |  |  |  |
| --- | --- | --- | --- |
| **Methods** | **Image Acquisition** | **Pre-processing** | **Density Calculation** |
| Backgroun  d  Subtraction  technique | Uses Cameras | Grayscale  conversion, Binary conversion,  Erosion,  Dilation | Motion  detection using  Consecutive  frame  comparison  based on  histogram key  region and  Vehicle  detection using  background |
| Canny  Edge  Detection  Technique | Uses Cameras | Grayscale  conversion,  Background  subtraction | Canny edge  detection for  vehicle edge  detection, Moore  neighbourhood  algorithm for  object count |
| Self  proposed  algorithm  by author | Uses Cameras | Grayscale  conversion | Self-proposed  algorithm and  formula for  vehicle density  calculation |
| Dual  method  technique | Uses Cameras | Grayscale  conversion | using a  combination of  gradient  magnitude and  direct  subtraction  techniques to  detect vehicles |
| Gradient  Method | Uses  cameras | Grayscale conv,  Gamma | using canny  edge detector |

## 2.4 Traffic Control

The calculation of vehicle count/density is utilized for further traffic control for different purposes in different methods. In the first method, the vehicle count is used to develop an android app that will give the user details about the traffic jam conditions at any particular location.

In the second method, the density calculation helped in automatic traffic lights switching for better traffic management. It contributed to a special feature, i.e. detection of presence of emergency vehicle on the lane.

When this happens, then that lane is given preference over others and the traffic lights are switched accordingly.

Similarly, in the third, fourth and fifth method the traffic density/traffic count calculation helps in automatic switching of traffic signals, based on the number of vehicles present at any particular lane at any instance of time.

The comparison of these algorithms shows that vehicle density calculation can be achieved with the help of various algorithms. Table I briefs the whole comparison of the five methods discussed.

**CHAPTER 3**

**TECHNOLOGIES USED**

**3.1 NetBeans IDE**

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++and HTML5.

NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM.

The NetBeans Team actively support the product and seeks feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback.

The NetBeans Platform is a framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

• User interface management (e.g. menus and toolbars)

• User settings management

• Storage management (saving and loading any kind of data)

• Window management

• Wizard framework (supports step-by-step dialogs)

• NetBeans Visual Library

• Integrated development tools

**3.3 Software Environment**

## Java Technology

Java technology is both a programming language and a platform.

### The Java Programming Language

### The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

* + - Simple
    - Architecture neutral
    - Object oriented
    - Portable
    - Distributed
    - High performance
    - Interpreted
    - Multithreaded
    - Robust
    - Dynamic
    - Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



### The Java Platform

A platform is the hardware or software environment in which a program runs. We’ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it’s a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

* The Java Virtual Machine (Java VM)
* The Java Application Programming Interface (Java API)

We’ve already been introduced to the Java VM. It’s the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that’s running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

## What Can Java Technology Do?

The most common types of programs written in the Java programming language are applets and applications. If you’ve surfed the Web, you’re probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

* **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
* **Applets**: The set of conventions used by applets.
* **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
* **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
* **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
* **Software components**: Known as JavaBeansTM, can plug into existing component architectures.
* **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
* **Java Database Connectivity (JDBCTM)**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.

**CHAPTER 4**

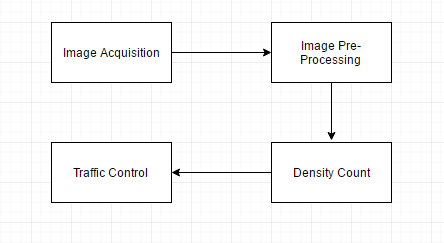
**DESIGN AND IMPLEMENTATION**

**4.1 INTRODUCTION**

Design is the first step in the development phase for any engineered product or system. The designer’s goal is to purchase a model or representation of an entity that will later be built.

**4.2 System Level Design**

An intelligent transportation system also known as traffic control system constitutes of Image acquired from a camera for generating digital image of traffic. Traffic lights can be controlled from the block diagram shown below:



The algorithm behind the block diagram consists of following steps

1. We have a reference image and the image to be matched is continuously captured using a camera that is installed at the junction.

2. The images are pre-processed in two steps as follows

a. Images are rescaled to 300x300 pixels.

b. Then the above rescaled images are converted from RGB to gray.

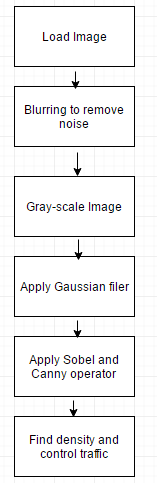
3. Edge detection of pre-processed images is carried out using Canny edge detection technique.

4. Number of cars on the road are counted, alias density is calculated.

5. Traffic signals are controlled accordingly.

**4.3 Algorithm Level Design**

To implement control over traffic signal, a six-stage approach is widely used by researchers. These stages are preprocessing, edge detection and traffic signal control. The steps of the complete process used in the work are show below.



The first two stages after image acquisition, i.e. blurring and RGB to grayscale conversion comes under pre-processing stage and the next stages are edge detection operations. The last stage help controlling traffic lights according to the density of the road.

**4.4** **SYSTEM REQUIREMENTS**

**4.4.1 HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.

**4.4.2 SOFTWARE REQUIREMENTS:**

* Operating system : Windows XP/7.
* Coding Language : JAVA/J2EE

**4.4 IMPLEMENTING SYSTEM**

As per the proposed system we capture the image from a file using JFileChooser in JAVA programming language. We implement the system mainly on two processes namely,

(a) Density Calculation

(b) Traffic Control System

**Density Calculation**

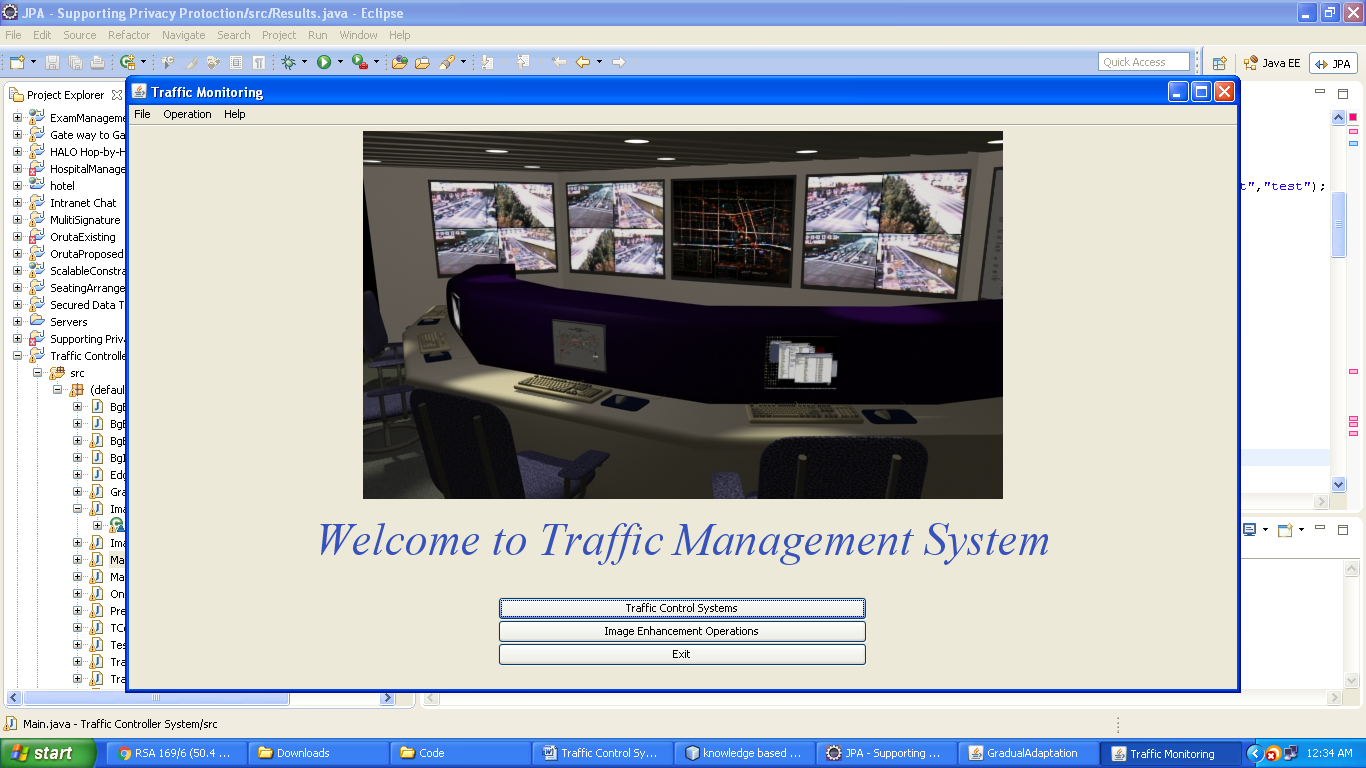
Density of the road is calculated after pre-processing the image.

**Traffic Control System.**

Density is adjusted on the density progress bar according to which traffic signals on the four way lane are adjusted.

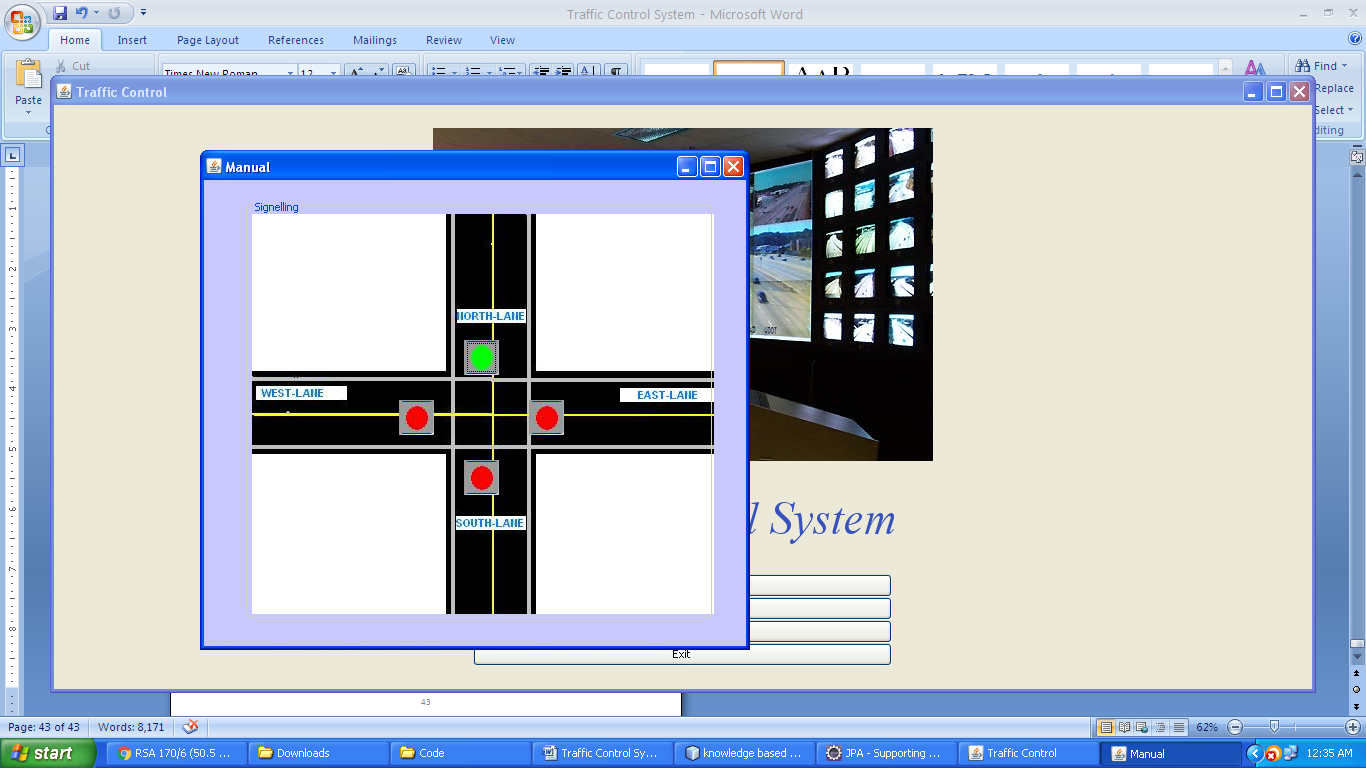
**CHAPTER 5**

1. **User Interface**

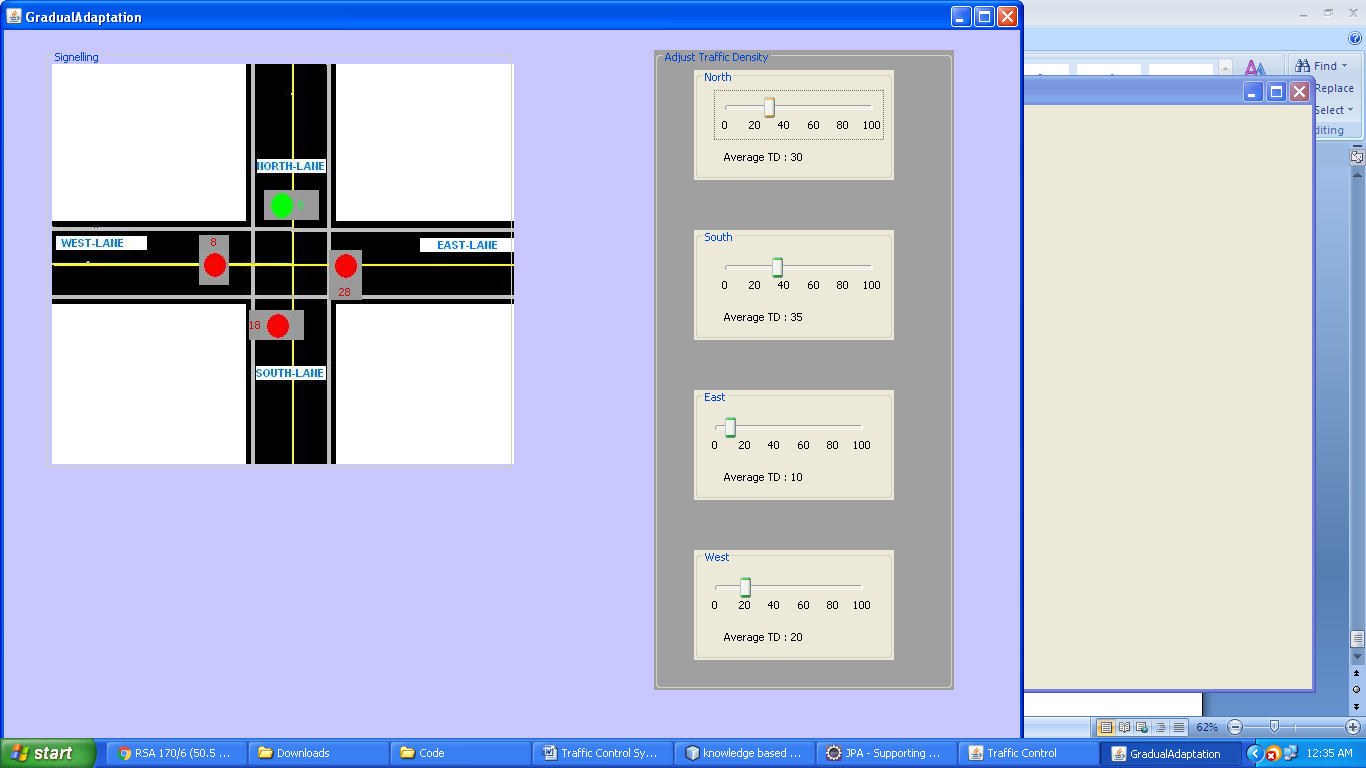


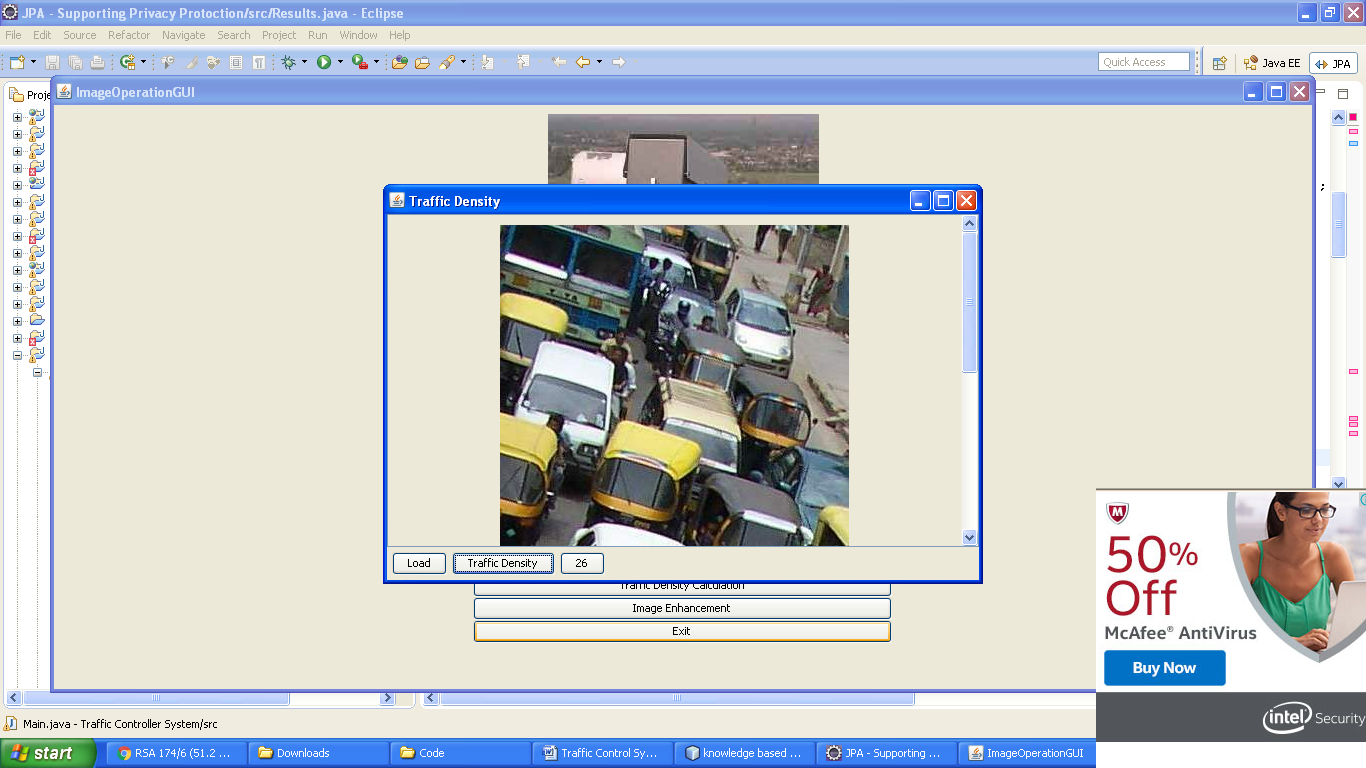
1. **Types of available systems**



**3. Controlling traffic manually**

**4. Controlling traffic with traffic density**



**5. Finding Traffic Density**

**CONCLUSION**

“**Traffic control using image processing**” technique that we propose overcomes all the limitations of the earlier (in use) techniques used for controlling the traffic. Earlier in automatic traffic control use of timer had a drawback that the time is being wasted by green light on the empty. This technique avoids this problem. Upon comparison of various edge detection algorithms, it was inferred that Canny Edge Detector technique is the most efficient one. The project demonstrates that image processing is a far more efficient method of traffic control as compared to traditional techniques. The use of our technique removes the need for extra hardware such as sound sensors. The increased response time for these vehicles is crucial for the prevention of loss of life. Major advantage is the variation in signal time which control appropriate traffic density using Density Calculation. The accuracy in calculation of time due to single moving camera depends on the registration position while facing road every time.

For object counting, Canny edge detector has proved to be the most efficient according to the literature survey done as it is not susceptible to noise interference and it also detects true weak edges Introducing such changes can help getting better results and thus making image processing better method for traffic density calculation than any other method.

**BIBLIOGRAPHY**

* http://www.tutorialspoint.com/jdbc/
* http://www.tutorialspoint.com/java\_dip/
* Digital image processing by Rafael C. Gonzalez and Richard E. Woods.
* Ahmed S. Salama, Bahaa K. Saleh, Mohamad M. Eassa, ”Intelligent Cross Road Traffic Management System (ICRTMS),” 2nd Int. Conf. on Computer Technology and Development, Cairo, Nov 2010, pp. 27-31.
* B. Fazenda, H. Atmoko, F.Gu, L. Guan1 and A. Ball” Acoustic Based Safety Emergency Vehicle Detection for Intelligent Transport Systems” ICCAS-SICE, Fukuoka, Aug 2009, pp.4250-4255.
* Z. Jinglei, L. Zhengguang, and T. Univ, “A vision-based road surveillance system using improved background subtraction and region growing approach,” Eighth ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Qingdao, August 2007, pp. 819-822.
* M. Siyal, and J. Ahmed, “A novel morphological edge detection and window based approach for real-time road data control and management,” Fifth IEEE Int. Conf. on Information, Communications and Signal Processing, Bangkok, July 2005, pp. 324-328.
* Y. Wu, F. Lian, and T. Chang, “Traffic monitoring and vehicle tracking using roadside camera,” IEEE Int. Conf. on Robotics and Automation, Taipei, Oct 2006, pp. 4631– 4636.
* Reulke, S. Bauer, T. D’oring, F. Meysel, “Traffic surveillance using multi-camera detection and multi-target tracking”, Proceedings of Image and Vision Computing New Zealand 2007, pp. 175–180, Hamilton, New Zealand, December
* Traffic light control system simulation through vehicle detection using image processing By Mac Michael B. Reyes and Dr Eliezer A. Albaccea