

## Research Article

# Real time Road Traffic Light Signalling Using Image Processing With Emergency Vehicle Detection and Traffic Signal Violation Detection

Sreemana Datta

Heritage Institute of Technology, Chowbaga Road, Anandapur, P.O.East Kolkata Township, Kolkata – 700107, West Bengal, India

\*Corresponding author

SreemanaDatta

Email: [sreemanadatta@gmail.com](mailto:sreemanadatta@gmail.com)

**Abstract:** As the problem of traffic congestion increases, an optimal method to control traffic signalling has become important. We propose a system where traffic signalling is done based on the density of traffic on a particular lane. For a particular lane, a camera will be installed alongside the traffic lights. Live video feed from these cameras will be broken down into image frames to determine congestion level in each lane. An array of image processing techniques will be implemented on each image frame to determine the traffic density in each lane. After the video is acquired and broken down into image frames, the image is converted from RGB to a grayscale image. Following that, power law transformation is applied as an enhancement technique. Canny edge detection is performed on the enhanced image and finally a window based image matching technique is applied to determine the congestion level of a particular lane. This is done for all the lanes and correspondingly the duration and order of signalling is scheduled. The proposed system also detects emergency vehicles such as ambulance and preferential clearance is given to them. The system is also capable of detecting traffic violation and identifying a violating vehicle.

**Keywords:** Edge detection, Adaptively updated background, Window based image matching, Traffic Control, Emergency vehicle detection, Traffic violation detection.

## INTRODUCTION

Various techniques are being employed to manage and signal road traffic with the help of Intelligent Traffic Systems (ITS). Traditional methods of traffic control include manual traffic control, which requires continuous manpower and becomes very difficult to maintain if there is continuous flow of high density traffic throughout the day. Semi-automatic control of road traffic based on statistical data or round robin scheduling based on fixed time slots have also been employed but this method is not self-adaptive and hence leads to sparsely populated lanes to get green signal whereas densely populated lanes may be signalled red. A few other techniques are used to implement intelligent traffic systems such as by using GPS, PLC, PIC, RFID tags, Infrared sensors, Inductive loops and Electromagnetic sensors [1-4]. Most of these techniques require some additional hardware to be installed on the vehicle which makes difficult to implement on a large scale. Also, some of these techniques involve use of costly and sophisticated instruments such as GPS sensors which makes them infeasible for use on a wide scale. Techniques employing infrared sensors, inductive loops and electromagnets have been observed to deteriorate in performance during harsh weather conditions such as rainfall, snow and storm.

We propose a system that uses image processing techniques to determine the density of vehicles on a particular road at a particular point of time

[5-7]. Several image processing techniques have been implemented already for this purpose such as background subtraction, mean and median filtration techniques, lane masking and object counting. Background subtraction tries to demarcate the roads as background from the cars in the forefront. But this technique often fails to detect a dark coloured car as a foreground and hence leads to erroneous results. This technique also fails when ambient light is insufficient. Mean and median filtration respectively calculates the mean and median values of the previous n-1 frames to arrive at a decision about the nth frame. This method is seen to falter in cases where the traffic density is not uniform or there are fluctuations in traffic congestion levels with time. Lane masking tries and separates the moving parts of a frame such as the cars from the static parts such as roads and then arrive at a decision. This method gives good results only in cases where all the cars are mobile. In case a car is standing at a crossing which is signalled red, then this method is prone to error. Object counting techniques involve huge amount of computation at each frame leading to a slow speed of computation. This method also suffers from the flaw that objects other than cars such as human or animals on the road may be counted as cars and signalling is done accordingly and turns out to be wrong.

The proposed system aims to achieve the following:

- i) Distinguish the presence and absence of vehicles on roads.

- ii) Determine the density or congestion level of a particular road.
- iii) Schedule the order of signalling based on the traffic density.
- iv) Ensure that the most congested road gets the largest share of green signal time duration
- v) Identify the presence of emergency vehicles such as ambulance and give them immediate clearance even if the road is empty.
- vi) Detect traffic signal violation by a vehicle and determine the vehicle which has violated traffic signal.

## EXPERIMENTAL

Components of the current project include

- i) Hardware module
- ii) Software module
- iii) Interfacing

### Hardware Module

Image sensors: In this project a USB based web camera has been used.

Computer: A general purpose PC as a central unit for various image processing tasks and decision making has been used.

Platform: Consists of a few toy vehicles and LEDs (prototype of the real world traffic light control system). Toy vehicles with LED mounted to top is used as a prototype for emergency vehicle with a beacon

### Software Module

The Software module has been implemented using JAVA comprising of specialized modules that perform specific tasks has been used.

### Interfacing

The interfacing between the hardware prototype and software module is done using parallel port of the personal computer. Parallel port driver has been installed in the PC for this purpose.

## METHODS

The following techniques have been implemented:

- i) Image acquisition
- ii) RGB to grayscale conversion
- iii) Image enhancement
- iv) Edge detection
- v) Window based image matching on adaptively updated background
- vi) Traffic light signalling based on the data found.

### Image acquisition:

Real time video signal from the cameras mounted at roadside traffic signals are obtained and

broken down into frames for image processing. Then image processing algorithm is applied on frames after certain intervals and decision is made on whether vehicle is present or not.

### RGB to grayscale conversion:

This is done to reduce the computation time while not having to compromise on computational efficiency. The lookup table of a RGB image consists of 24 bits (8 bits each for R,G and B since each may have intensity values from 0-255) whereas lookup table of grayscale image consists of only 8 bits (intensity value from 0-255). As edge detection algorithms provide similar results for both RGB and grayscale images, we perform this step to reduce time of computation.

### Image enhancement:

This is done to bring the image in contrast to background so that a proper threshold level may be selected while binary conversion is carried out and the result is more suitable than the original image for the specific application.

It is often seen that the enhanced image gives better results in edge detection than the original image.

Image enhancement simply means, transforming an image  $f$  into image  $g$  using  $T$ . (Where  $T$  is the transformation. The values of pixels in images  $f$  and  $g$  are denoted by  $r$  and  $s$ , respectively. As said, the pixel values  $r$  and  $s$  are related by the expression,

$$s = T(r)$$

Where  $T$  is a transformation that maps a pixel value  $r$  into a pixel values. The results of this transformation are mapped into the gray scale range as we are dealing here only with gray scale digital images. So, the results are mapped back into the range  $[0, L-1]$ , where  $L=2^k$ ,  $k$  being the number of bits in the image being considered. So, for instance, for an 8-bit image the range of pixel values will be  $[0, 255]$ .

### Popular image enhancement algorithms include:

- i) Linear (Negative And Identity Transformations):  $[s = c * (ar+b)]$
- ii) Logarithmic (Log & Inverse Log Transformations):  $[s = c * \log(1+r)]$
- iii) Power Law Transformations (Gamma Correction):  $[s = cr^\gamma]$
- iv) Gray Level Slicing: Similar to a band pass filter.
- v) Taking Negative Of An Image:  $N(r, c) = 255 - I(r, c)$  where  $0 \leq r \leq R$  and  $0 \leq c \leq C$ . [ $R$ ,  $C$ : Number of rows and columns in the image].

The point processing methods are most primitive, yet essential image processing operations and are used primarily for contrast enhancement. Image Negative is suited for enhancing white detail embedded in dark regions and has applications in medical imaging.

Power-law transformations are useful for general purpose contrast manipulation. For a dark image, an expansion of gray levels is accomplished using a power-law transformation with a fractional exponent.iii) Log Transformation is Useful for enhancing details in the darker regions of the image at the expense of detail in the brighter regions the higher-level values.

For an image having a washed-out appearance, a compression of gray levels is obtained using a power-law transformation with  $\gamma$  greater than 1. The histogram of an image (i.e., a plot of the gray level frequencies) provides important information regarding the contrast of an image. Histogram equalization is a transformation that stretches the contrast by redistributing the gray-level values uniformly. Only the global histogram equalization can be done completely automatically. As Power Law transformation works well for general contrast manipulation, we used power law transformation or gamma correction. For daytime, the value of  $\gamma$  was taken as 0.5 in order to darken the image and increase contrast so that cars are not lost or washed out due to reflection of excessive illumination. For night time, the value of  $\gamma$  was taken as 1.5 in order to enhance light in the image and ensure that cars are not lost in the darkness or due to less ambient light.

**Edge detection:**

The image consists of objects of interest displayed on a contrasting background; an edge is a transition from background to object or vice versa. The total change in intensity from background to foreground is called the strength of the edge or edge detection. After converting the acquired image into grayscale, edge detection can be performed on it using anyone of the following algorithms:

**i) Local Threshold and Boolean Function Based Edge Detector:**

This edge detector is fundamentally different than many of the modern edge detectors derived from Canny’s original paper. It does not rely on the gradient or Gaussian smoothing. It takes advantage of both local and global thresholding to find edges. Unlike other edge detectors, it converts a window of pixels into a binary pattern based on a local threshold, and then appliesiv) masks to determine if an edge exists at a certain point or not. By calculating the threshold on a per pixel basis, the edge detector should be less sensitive to variations in lighting throughout the picture. It does not rely on blurring to reduce noise in the image. It instead looks at the variance on a local level.

**ii) Marr - Hildreth Edge Detector**

The Marr-Hildreth edge detector was a very popular edge operator before Canny released his paper. It is a gradient based operator which uses the Laplacian to take the second derivative of an image. The idea is that if there is a step difference in the intensity of thev)   
vi)

image, it will be represented by in the second derivative by a zero crossing

**Sobel Operator**

The operator consists of a pair of 3x3 convolution kernels as shown in Figure 1. One kernel is simply the other rotated by 90°.

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

**Gx**

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

**Gy**

These kernels are designed to respond maximally to edges running vertically and horizontally relative 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 (call these  $G_x$  and  $G_y$ ). 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:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(G_y/G_x)$$

**Prewitt’s Operator**

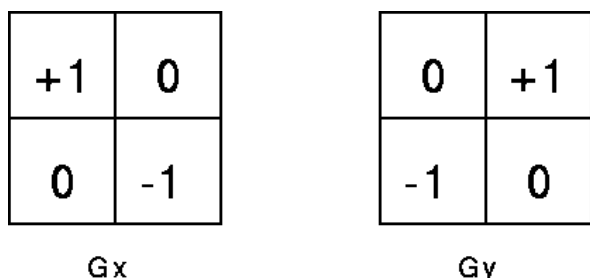
Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images.

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

vii) **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 2x2 convolution kernels as shown in Figure. 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 (call these *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:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

although typically, an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

which is much faster to compute.

The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by:

$$\theta = \arctan(Gy/Gx) - 3\pi/4$$

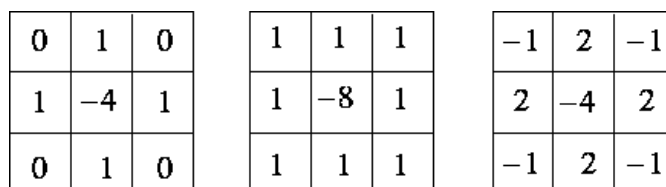
viii) **Laplacian of Gaussian (LoG)**

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 graylevel image as input and produces another graylevel image as output.

The Laplacian  $L(x,y)$  of an image with pixel intensity values  $I(x,y)$  is given by:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

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. Three commonly used small kernels are shown in



Edge detection of all four types was performed and Canny yielded the best results. This was expected as Canny edge detection accounts for regions in an image. Canny yields thin lines for its edges by using non-maximal suppression. Canny also utilizes hysteresis when thresholding.

As Canny Edge Detection Algorithm was found to be most efficient, we used it for edge detection.

**Steps Of Canny Edge Detection Algorithm Includes:**

- i) Smooth the image with a two dimensional Gaussian. In most cases the computation of a two dimensional Gaussian is costly, so it is approximated by two one dimensional Gaussians, one in the x direction and the other in the y direction.
- ii) Take the gradient of the image. This shows changes in intensity, which indicates the presence of edges. This actually gives two results, the gradient in the x direction and the gradient in the y direction.
- iii) Non-maximal suppression. Edges will occur at points where the gradient is at a maximum.
- iv) Therefore, all points not at a maximum should be suppressed. In order to do this, the magnitude and direction of the gradient is computed at each pixel. Then for each pixel check if the magnitude of the gradient is greater at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient. If the pixel is not greater than both, suppress it as shown below:

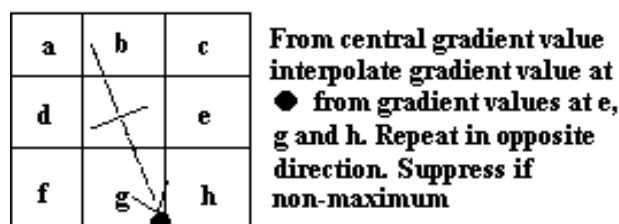


Fig.: Non-Maximal Suppression.

- v) Edge Thresholding: The method of thresholding used by the Canny Edge Detector is referred to as "hysteresis". It makes use of both a high threshold and a low threshold. If a pixel has a value above the high threshold, it is set as an edge pixel. If a pixel has a value above the low threshold and is the neighbor of an edge pixel, it is set as an edge pixel as well. If a pixel has a value above the low threshold but is not the neighbor of an edge pixel, it is not set as an edge pixel. If a pixel has a value below the low threshold, it is never set as an edge pixel.

#### Window based image matching on adaptively updated background:

Image matching is required in order to determine the congestion and traffic density of each road. For this, we applied the above mentioned steps on the image of an empty road and store it as the reference. We go on changing this background image adaptively. This means, at any instant, the background has contributions of the  $x$  times the  $n$ th frame and  $(1-x)$  times previous (say ' $n-1$ ') frames, where  $x$  is a constant between 0 and 1 and is defined as the contribution factor. We worked with a sequence of 10 frames per cycle and hence chose the value of  $x$  to be 0.9. At any instant of time, this adaptively updated background takes care of greater accuracy in cases where there are fluctuations in traffic density. Now at any instant, the operations mentioned earlier, that is, Image acquisition, RGB to grayscale conversion, Image enhancement and Edge detection are performed on each video frame. Then, a window based image comparison is done on each of these frames to compare them with the background image for that instant. Window based image matching refers to sub dividing the entire image into smaller windows and then iterating over the windows to compute the percentage of match. Rather than iterating pixel by pixel, this method provides better time complexity though the results are received at an acceptable level of approximation. Pixel by pixel comparison turned out to be time inefficient and hence, the technique of window based matching was employed.



Emergency Vehicle Detected

In this technique, the larger image is divided into several windows of the dimensions of the smaller image. The idea is that, if two images differ, each of these windows will differ and vice versa. Likewise, the decision is made.

Comparisons of the image of the empty road with that of image frames for each lane resulted in four distinct values of percentage match corresponding to each road with respect to the background image.

#### Traffic light signalling based on the data found:

After the percentage match of each lane is computed, it is determined that higher the percentage match for each lane as a result of image matching with the empty lane, less densely populated is the lane. A thickly congested lane was seen to show very less percentage match whereas an empty road showed high value of percentage match.

Hence, traffic signalling protocol was finalised as:

- Match Of 10 To 50% - Green Light On For 60 Seconds
- Match Of 50 To 70% - Green Light On For 30 Seconds
- Match Of 70 To 90% - Green Light On For 20 Seconds
- Match Of 90 To 100% - Red Light On For 60 Seconds

#### Emergency Vehicle Detection:

The video is also analysed for the detection of emergency vehicles through their flashing red lights. By specifying a threshold, we have isolated the areas with high intensity of red light and comparatively lesser intensity of blue and green colour. To distinguish from Head lights, the red light must satisfy the additional condition of blinking. This is achieved by taking account for the fact that the red light shall appear in every third frame only. The other lights do not appear in the image sequence with this frequency and hence are eliminated. This leads to the conclusion of the presence of an emergency vehicle. This comparison too, is done for each of the specified windows. On detection of an emergency vehicle on a particular lane, that lane is arbitrarily assigned a green signal for 30 seconds in order to allow the emergency vehicle to leave. Similar processing is done for blue beacons as well.



Emergency Vehicle Not Detected



**Traffic Signal Violation Detection:**

Traffic signal violation is detected by searching for a missing vehicle in consecutive video frames when the signal is red. The idea is that, whenever the signal is red, every vehicle which is present within the image in the previous frame, should be present in the current frame also. The whole image frame is sub divided into numerous pre-determined sub frames or windows and a region of mismatch is searched between them. This region corresponds to the vehicle that has moved out of the signal when the red light was on, thereby detection traffic signal violation. It is not accepted as violation if the position of a vehicle changes within a frame, but violation is detected only when the signal is red and a particular vehicle which was present in the previous frame is absent altogether in the current frame. The variations in position of a vehicle are taken care of by applying scale invariant feature transform or SIFT on each of the window or sub images.



**Traffic Violation Detected and Violating Car Identified**

When a violation has been detected, the window within which the percentage match has changed the most from the previous to the current frame is identified. The previous frame is stored until now and a circle is drawn in the previous frame at the position of the window where the highest percentage mismatch has occurred. This window corresponds to a vehicle moving out of it and hence that part of the road becomes empty and gives a higher percentage match with the reference image in the next frame.



**Previous Frame**



**Current Frame**



**Reference Image**

**RESULTS AND DISCUSSIONS**



Edge Detected Reference Image



Enhanced Image



Original Image



Edge Detected Image (36.256% Match)



Grayscale Image

It was seen that as the window size was increased, the time required for computation was decreased, but the accuracy of the result was decreased as well. However as the window size was decreased, the accuracy of computation increased, but time required increased as well. Pixel by pixel comparison can be viewed as a window based comparison where window size is 1x1. It provides the best result in terms of accuracy but is very time consuming.

As we had to perform all these steps on every frame of the image sequence, it was necessary to determine an optimum window size with an acceptable level of accuracy and time complexity. The following results were received.

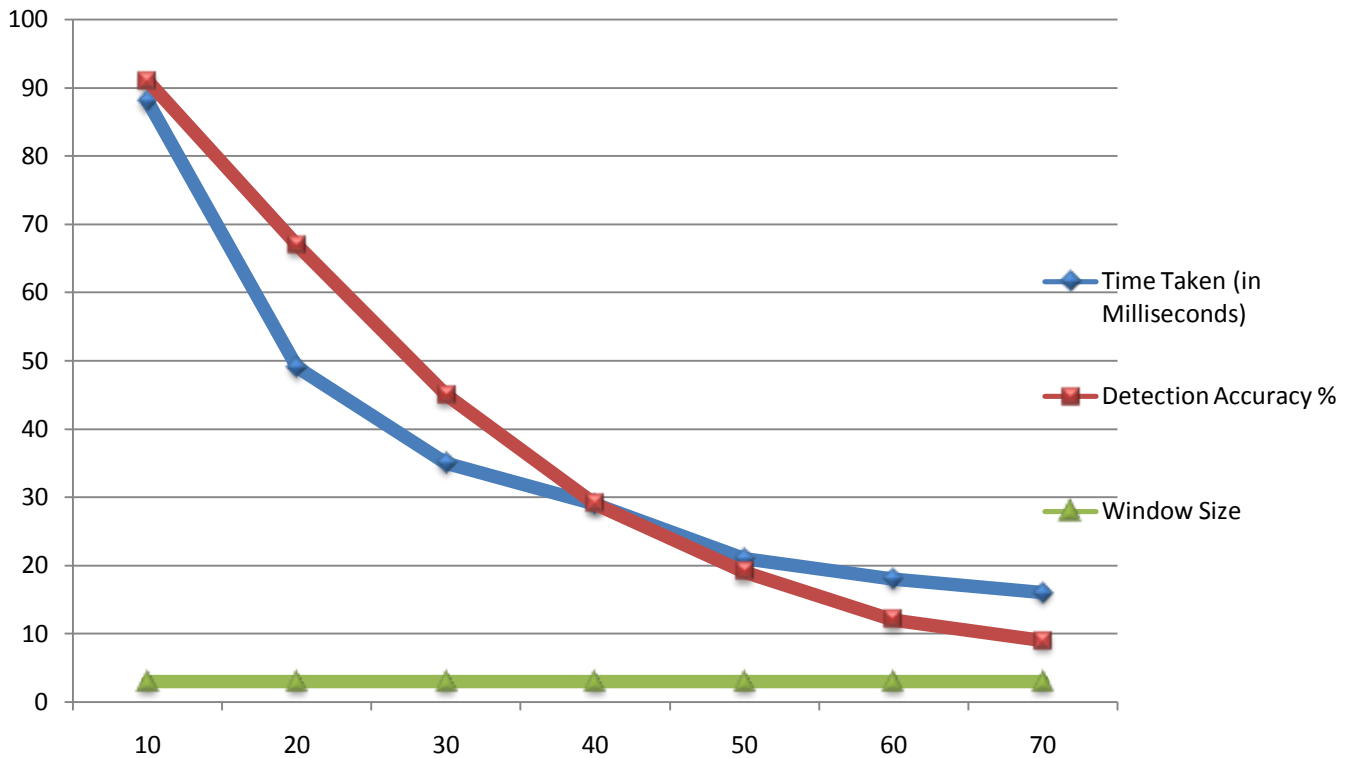


Fig: Optimum window size with an acceptable level of accuracy and time complexity.

It is also to be noted that among all the edge detection algorithms, Canny edge detection provided with the most accurate results although it is most time consuming in comparison with other established algorithms.

The proposed system has been tested and works well for up to four lanes and it has been observed that traffic signal violation and emergency vehicle detection can be performed alongside adaptive traffic light regulation. This system may be further expanded by adding number plate detection system which would be helpful to track down vehicles violating traffic signal.

Real time traffic light control using image processing with adaptively updated background may further be expanded by inclusion of more number of test parameters, larger set of test cases and more number of lanes from which video stream is received simultaneously.

**References**

1. Do H. Nama, Donald R. Drew, Automatic measurement of traffic variables for intelligent transportation systems applications.

2. Transportation Research Part B: Methodological, 1999; 33(6): 437-457.
2. Fei-Yue Wang, Daniel Zeng ; Liuqing Yang, Smart Cars on Smart Roads: An IEEE Intelligent Transportation Systems Society Update. Pervasive Computing, IEEE, 2006; 5(4):68-69.
3. Mirchandani P, Fei-Yue Wang, RHODES to intelligent transportation systems. Intelligent Systems, IEEE, 2005; 20(1):10-15.
4. Kastrinaki V, Zervakis M, and Kalaitzakis K, A survey of video processing techniques for traffic applications,"Image and Vision Computing, 2003; 21:359-381.
5. Hummel R. Histogram modification techniques, Computer Graphics and Image Processing, 1975;4:209-224.
6. Fathy M and Siyal MY, An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis,"Pattern Recognition Letters, 1995; 16:1321-1330.
7. Rita Cucchiara, Massimo Piccardi and Paola Mello, Image analysis and rule-based reasoning for a traffic monitoring system," IEEE Trans. on Intelligent Transportation Systems, 2000; 1 (2):119-130.