Gray Eye Traffic Surveillance

Ayesha Butalia[#1], Durgesh Maru[*2], Kuldeep Baheti[*3], Avais Mohammad[*4], Ankit Bagdiya[*5]
[#]*Professor, Department of Computer Engineering, MIT College of Engineering, Paud Road, Kothrud, Pune-411 038*
[*]*Department of Computer Engineering MIT College of Engineering, Paud Road, Kothrud,, Pune-411 038.*

***Abstract*— Gray Eye Traffic Surveillance (GETS) is a system based on image processing technology and database processing for the purpose of Traffic Surveillance. This system is built specifically for India, where standard fonts and scripts for the number plates are rarely followed and corruption is a menace.**

This paper attempts to present a system to automate the entire process of traffic surveillance and curb corruption at the same time. A two way approach of developing a desktop and a web based mobile application is proposed. The desktop based application uses image processing technology to recognize the license number from image of vehicles captured by surveillance cameras and maintain a track log. The web based mobile application can be used by authorized traffic police to file complaints. The general public can log into this application to view their log and file complaints in case of any discrepancies. Different database schemas are used for various purpose like authentication, crime details, tracking system.

Key Words: Image Processing, Traffic Surveillance, Feature Extraction.

1. INTRODUCTION

GETS is a system, which provides mobile and desktop based applications for Traffic Surveillance. It addresses various problems faced in the Indian context, where standard fonts and scripts for the License plates are rarely followed and people indulge in corrupt practices after violating the traffic rules.
The system helps for automated traffic surveillance and tracking system, automated high-way/parking toll collection systems, automation of petrol stations, journey time monitoring, fine collection by the police and also provides an interface to general public to register complaints and view their log.
The process of Traffic Surveillance is made convenient, time efficient and unbiased by the use of GETS.

1.1 Existing System

Automatic Traffic Surveillance is already being done in various countries like USA, Australia, Korea, etc. It is mainly being done using the ANPR (Automatic Number Plate Recognition) system. Since standard fonts and scripts are used for the number plates and strict judicial enforcement is practiced, the implementation of ANPR is comparatively easier than the Indian circumstances. These existing systems use standard features of the license plates such as: dimensions of plate, border for the plate, color and font of characters, etc, which help to localize the number plate easily and identify the license number of the vehicle.

In India, number plate standards are rarely followed. Wide variations are found in terms of font types, script, size, placement and color of the number plates. In few cases, other unwanted decorations are present on the number plate. Also, unlike other countries, no special features are available on Indian number plates to ease their recognition process. Hence, currently only manual recording systems are used and ANPR has not been commercially implemented in India.

1.2 Proposed System

In designing this system (fig. 1), a website is made, which maintains a log of all vehicles registered with the RTO (Regional Transport Office). A two way approach: Mobile based and Desktop based is used. The mobile will be manually operated by the traffic police. They have to log into their domain and register complain of the vehicle violating the traffic rules and confirm payment of fine for the same. General public can log into the public web URL, designed specifically for them, view their log and file complaints, if any.
The desktop based system makes use of image processing technology, which extracts the License number from the vehicle image and makes an entry into the track log of the database.

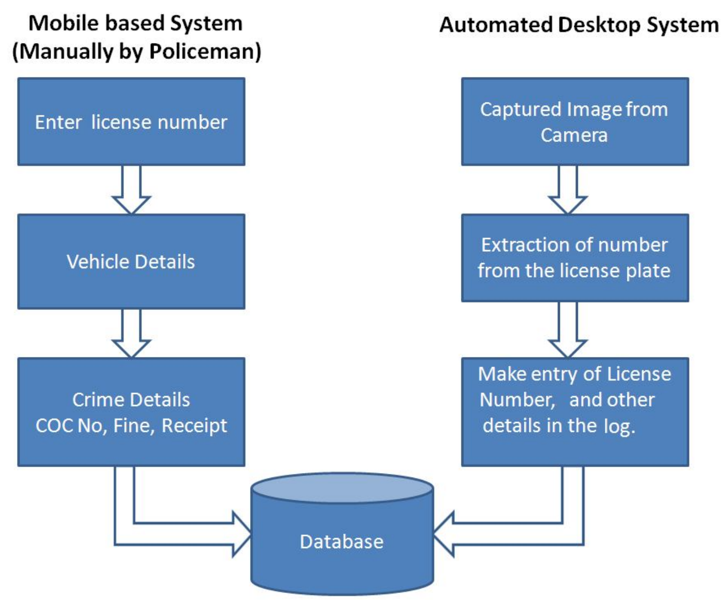

Figure 1: Block Diagram of proposed system.

2. METHODOLOGY

The system is being implemented by using a two way approach of a desktop and web based mobile application. A common

database is being used for both of these approaches. Different database schemas- track log, crime log, RTO database and the Police login database are maintained. They help in the process of authentication, entering the crime details, vehicle tracking, journey time monitoring, etc.
The working of these applications has been explained as under:

2.1 Desktop Based Application

The desktop based system uses image processing technology to recognize the license number of a vehicle from its image captured by the surveillance cameras. A number of algorithms have been used in various ANPR systems. They vary in the efficiency with which they detect the correct license number. The algorithm shown in Figure 2 has a greater efficiency as per the results shown in Table 1 and is suitable for the Indian scenario as discussed under the existing system.

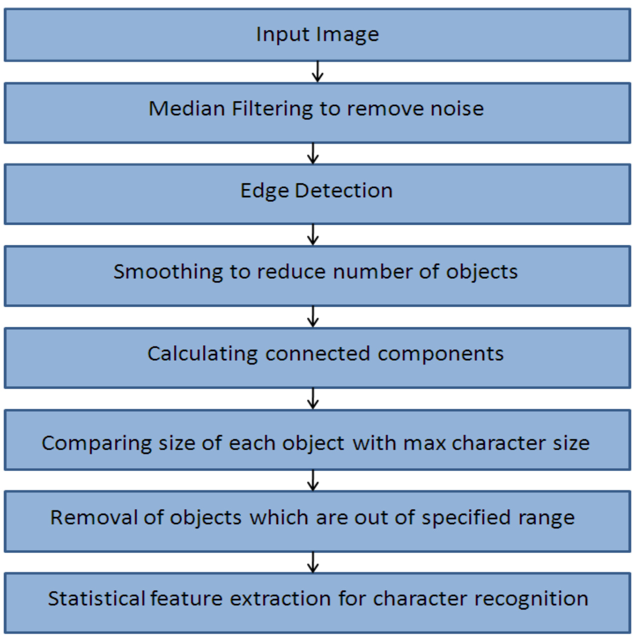

Figure 2: Image Processing Algorithm.

Figure 3: Original Image

The original image of the vehicle (fig. 3), is captured by Surveillance Cameras and is given as input to the system. It is then converted to a grayscale image (fig. 4). Then median filtering is performed on the image to remove noise.

Figure 4: Gray Scale Image

2.1.1Median Filtering:

The main idea of median filter is to run through the signal, entry by entry, replacing each entry with the median of neighboring entries (fig. 5). Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image) [13].

Figure 5: Median Filtered Image

The median filter is also a sliding-window spatial filter, but it replaces the center value in the window with the median of all the pixel values in the window. As for the mean filter, the kernel is usually square but can be of any shape. An example of median filtering of a single 3x3 window of values is shown below.

unfiltered values

6	2	0
3	97	4
19	3	10

median filtered

*	*	*
*	4	*
*	*	*

In ascending order of values :

0, 2, 3, 3, **4**, 6, 10, 15, 97.

Center value (previously 97) is replaced by the median of all nine values (4). Note that for the above example, the median filter would also return a value of 5, since the ordered values are 1, 2, 3, 4, **5**, 6, 7, 8, 9. For the second (bottom) example, though, the mean filter returns the value 16 since the sum of the nine values in the window is 144 and 144 / 9 = 16. This illustrates one of the celebrated features of the median filter:

its ability to remove 'impulse' noise (outlying values, either high or low). The median filter is also widely claimed to be 'edge-preserving' since it theoretically preserves step edges without blurring. However, in the presence of noise it does blur edges in images slightly.

Neighborhood averaging can suppress isolated out-of-range noise, but the side effect is that it also blurs sudden changes (corresponding to high spatial frequencies) such as sharp edges.

The *median filter* is an effective method that can suppress isolated noise without blurring sharp edges. Specifically, the median filter replaces a pixel by the median of all pixels in the neighborhood:

$$y[m,n] = median\{x[i,j],(i,j) \in w\}$$

where w represents a neighborhood centered around location *(mm, n)* in the image.

1D median filter:

Consider a 1x5 window sliding over a 1D array (either horizontal or vertical) of pixels. Assume the five pixels currently inside the windows are:

80	90	200	110	120

where the middle pixel with value 200 is an isolated out-of-range noise. The median of these five values can be found by sorting the values (in either ascending or descending order). The middle value is the median:

80	90	110	120	200

The original pixel value 200 is replaced by the median 110.

2D median filter:

The window of a 2D median filter can be of any central symmetric shape, a round disc, a square, a rectangle, or a cross. The pixel at the center will be replaced by the median of all pixel values inside the window.

Programming issues:

Sorting is necessary for finding the median of a set of values. There exit various sorting algorithm with complexity of *O(n log$_2$ n)* . However, in this case, as the number of pixels is quite limited, a simple sorting method with complexity O(n^2) can be used. The code segment below sorts an array of k elements:

$$
\begin{array}{l}
for(i = 0;\ \ i < k-1;\ \ i++) \\
for(j = i+1;\ \ j < k;\ \ j++) \\
\ \ if(bin[i]\ > \ bin[j]) \\
\ \ \ \ \{w = bin[i]; bin[i] = bin[j]; bin[j] = w;\}
\end{array}
$$

2.1.2Edge Detection:

Next, Edge detection is performed on the given image, which aims at identifying points in digital image at which image brightness changes sharply or, more formally, has discontinuities (fig. 6) (in surface orientation, in scene illumination, etc.)

The **Sobel operator** is used in image processing, particularly within edge detection algorithms [11]. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image. Mathematically, the operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define **A** as the source image, and $\mathbf{G}_x$ and $\mathbf{G}_y$ are two images which at each point contain the horizontal and vertical derivative approximations, the computations are as follows:

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A}$$

and

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}$$

where * here denotes the 2-dimensional convolution operation.

The *x*-coordinate is here defined as increasing in the "right"-direction, and the *y*-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$

Using this information, we can also calculate the gradient's direction:

$$\mathbf{\Theta} = \operatorname{arctan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

where, for example, $\mathbf{\Theta}$ is 0 for a vertical edge which is darker on the left side.

Comparisons with other operators

Figure 6: Output of Sobel Operator

Figure 7: Output of Prewitt Opeartor

Figure 8: Output of Roberts Operator

Figure 9: Output of Log Operaor

Figure 10: Output of Zero Cross Operator

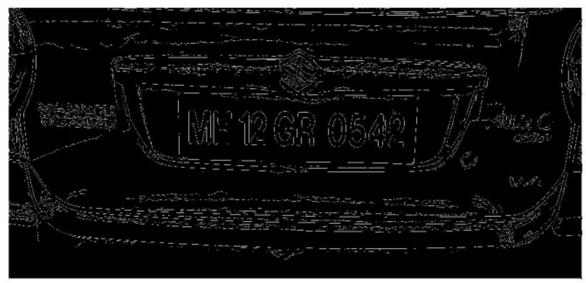

Figure 11: Output of Canny Operator

2.1.3Smoothing:

After edge detection, when the image converted to binary form, smoothing of the image (fig. 7) is performed by finding convolution of original image and the masking array. This process helps to merge smaller objects together, so that number of objects can be reduced. These objects are the connected components. BWlabel command used to label these connected components.

Reduction of components and character segmentation:

Each object is compared with maximum character size. For fixed distance, the possible range of character height and width is calculated. If it is not within the specified range of size, it is removed (fig. 11).

Figure 12: Smoothened Image

Figure 13: Reduced Components

2.1.4 Object Scissoring :

Object scissoring is a procedure to separate out object out of actual image and send it to character recognition step (for feature extraction).

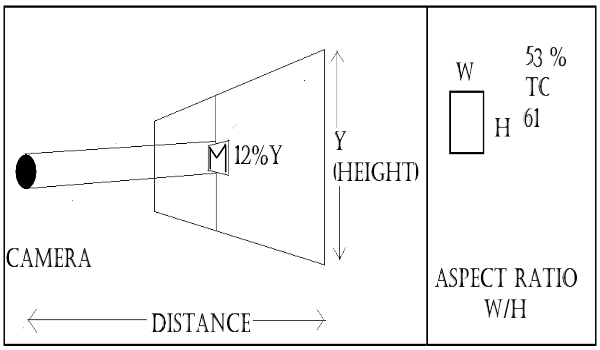

Figure 14: Range of size of maximum possible character

In Fig. 14,
D = Fixed Distance between Camera and Screen or Number plate
R [X Y] = Resolution of Camera X stands for Width and Y for Height
C [W H] = Character width and height [5].

Derived Attributes:
1) Specific range of height to check object scissoring from observing resolutions H is 12% of Y
2) Specific range of width to check object scissoring (Assumed Basic proportion in fonts) aspect ratio
8 X 13 = 61%
8 X 15 = 53%

Problem in object scissoring is about enclosed object as shown below. The dotted components are not shown in figure after object scissoring. For this, all labeled objects in range of shortlisted objects are also considered to be valid.

Figure 15: Dotted Components not shown

After this filling of inside region of each object is done

Figure 16: Filling of region

After this step, one can directly go and pass these objects to next step for pattern matching with given template. This method saves computation of horizontal and vertical scissoring of objects [1].

2.1.6 Feature extraction:
For extracting the feature, the zone based hybrid approach is proposed. The most important aspect of handwriting recognition scheme is the selection of good feature set, which is reasonably invariant with respect to shape variations caused by various writing styles [8]. The major advantage of this approach stems from its robustness to small variation, ease of implementation and provides good recognition rate. Zone based feature extraction method provides good result even when certain preprocessing steps like filtering, smoothing and slant removing are not considered. In this section, we explain the concept of feature extraction method used for extracting features for efficient classification and recognition. The following paragraph explains in detail about the feature extraction methodology.
The character centroid is computed and the image (character/numeral) is further divided in to fifty equal parts. Average distance from the character centroid to the each pixel present in the zone is to be computed. Similarly zone centroid is computed and average distance from the zone centroid to each pixel present in the zone is to be computed. We repeated this procedure for all the zones/grids/boxes present in the numeral image. There could be some zones that are empty, and then the value of that particular zone image value in the feature vector is zero. Finally 100 such features are used for feature extraction.
For classification and recognition, nearest neighbor classifier and feed forward back propagation neural network classifiers are used. Algorithm 1 provides Image centroid and Zone (ICZ) based distance metric feature extraction system. Algorithm 2 provides Zone Centroid and Zone (ZCZ) based Distance metric feature
extraction system. Algorithm 3 provides the proposed hybrid feature extraction system. (ICZ + ZCZ). The following are the algorithms to show theworking procedure of our feature extraction methods.

Algorithm 1: Image Centroid and Zone (ICZ) based Distance metric feature extraction system.

Input: Preprocessed numeral image

Output: Features for Classification and Recognition

Method Begins
Step 1: Compute the input image centroid
Step 2: Divide the input image in to **n** equal zones.

Step 3: Compute the distance between the image centroid to each pixel present in the zone.
Step 4: Repeat step 3 for the entire pixel present in the zone.
Step 5: Compute average distance between these points.
Step 6: Repeat this procedure sequentially for the entire zone.
Step 7: Finally, **n** such features will be obtained for classification and recognition.
Method Ends

Algorithm 2: Zone Centroid and Zone (ZCZ) based Distance metric feature extraction system.
Input: Preprocessed numeral image
Output: Features for Classification and Recognition.
Method Begins
Step 1: Divide the input image in to **n** equal zones.
Step 2: Compute the zone centroid
Step 3: Compute the distance between the zone centroid to each pixel present in the zone.
Step 4: Repeat step 3 for the entire pixel present in the zone.
Step 5: Compute average distance between these points.
Step 6: Repeat this procedure sequentially forthe entire zone.
Step 7: Finally, **n** such features will be obtained for classification and recognition.
Method Ends

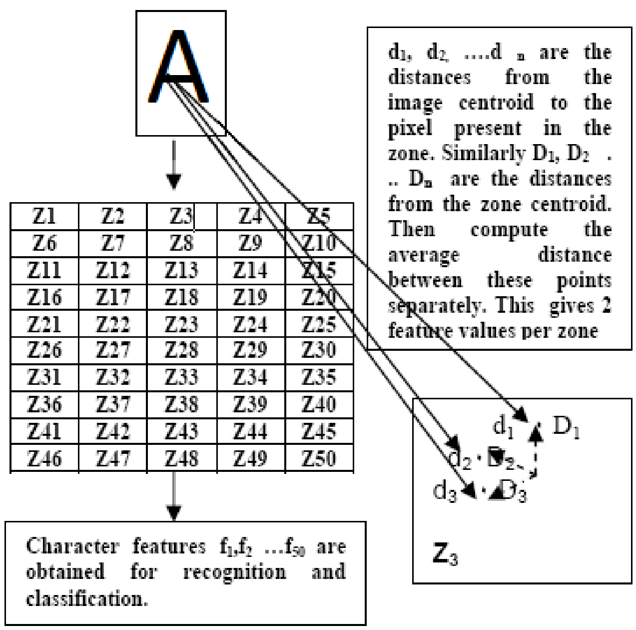

Figure 17: Procedure for extracting feature from image

Proposed Algorithm 3: ICZ + ZCZ based Distance metric feature extraction system.
Input: Preprocessed numeral image
Output: Features for Classification and Recognition
Method Begins
Step 1: Compute the input image centroid
Step 2: Divide the input image into **n** equal zones.
Step 3: Compute the distance between the image centroid to each pixel present in the zone.
Step 4: Repeat step 3 for the entire pixel present in the zone.
Step 5: Compute average distance between these points.
Step 6: Compute the zone centroid
Step 7: Compute the distance between the zone centroid to each pixel present in the zone.
Step 8: Repeat step 7 for the entire pixel present in the zone
Step 9: Compute average distance between these points.
Step 10: Repeat the steps 3-9 sequentially for the entire zone.
Step 11: Finally, **2*n** such features will be obtained for classification and recognition.
Method Ends

Classification and recognition

For large-scale pattern matching, a long employed approach is the nearest- neighbor classifier. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the actual classification phase, the same features as before are computed for the test samples. Distances from the new vector to all stored vectors are computed. Then Classification and recognition is achieved on the basis of similarity measurement.

An Artificial Neural Network (ANN) is a computational model widely used in pattern recognition [2]. It has been used extensively both for the recognition of Indian scripts. Feed forward back propagation network is used for subsequent recognition and classification of image.

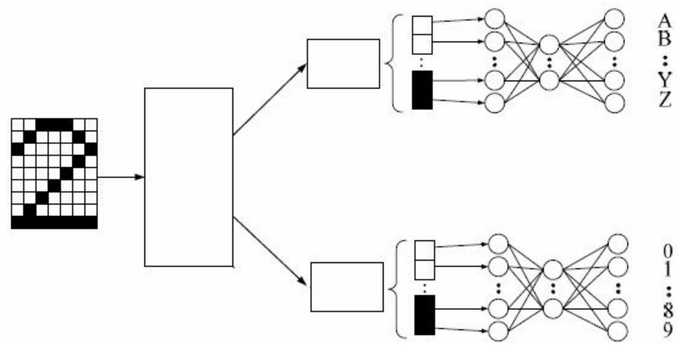

Figure 18: Design of neural network

Network structure and number of hidden layer nodes:

The recognition performance of Back Propagation network will highly depend on the structure of the network and training algorithm. Feed forward back propagation neural network has been selected to train the network. The number of nodes in input, hidden and output layers will determine the network structure. All the neurons of one layer are fully interconnected with all neurons of its just preceding and just succeeding layers (if any). The network consists of 50 nodes in the input layer (corresponding to one feature in each of the 50 zones), 80 neurons in the hidden layer. The output layer has 10 neurons corresponding to 10 numerals. Therefore only the number of hidden nodes needs to be determined. The number of hidden nodes will heavily influence the network performance. Insufficient hidden nodes will cause under fitting where the network cannot recognition the numeral because there are not sufficient adjustable parameters to model the input-output relationship. Excessive hidden nodes will cause over fitting where the network fails to generalize. There is no theoretical development based on which, the optimal number of neurons in the hidden layer can be determined.

There are several rules of thumb for deciding the number of neurons in the hidden layer.

- The number of hidden neuron should be less than twice the input layer size.
- The number of hidden neuron should be in the range between the size of the input layer and the size of the output layer.
- Finding the minimum number of epochs taken to recognize a character and recognition efficiency of training as well as testing samples.
- The number of hidden neurons should be 2/3 of the input layer size, plus the size of the output layer. We followed these rules to arrive at 80 neurons or the hidden layer.

Transfer function and performance function:

Since our desired outputs must be ranged between 0 to 1, so we have selected log sigmoid as the transfer function for both hidden and output layer. We have used 'Mean Squared Error' (MSE) as performance parameter function. MSE is the average squared error between the network outputs and the target outputs. During training, the weights of the network are iteratively adjusted to minimize the function. We adopt 'Gradient descent back propagation' as a learning algorithm. The algorithm updates weights according to gradient descent momentum and adaptive learning rate. The values we have used to set training parameters are learning rate to 0.1, momentum factor to 0.8 and performance goal to 0.01.

Other algorithms for future use:

a. Projection Histograms:

The basic idea behind using projections is that character images, which are 2-D signals, can be represented as 1-D signal [9]. These features, although independent to noise and deformation, depend on rotation. Projection histograms count the number of pixels in each column and row of a character image. Projection histograms can separate characters such as "m" and "n".

Figure 19: Projection of Character

b. Profiles

The profile counts the number of pixels (distance) between the bounding box of the character image and the edge of the character [9]. The profiles describe well the external shapes of characters and allow to distinguish between a great number of letters, such as "p" and "q".

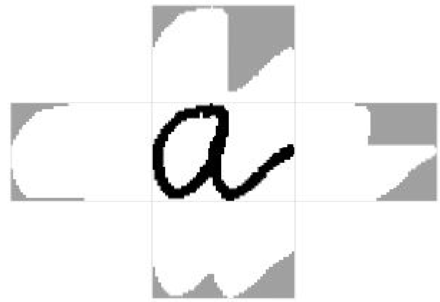

Figure 20: Profile of Character

Profiles can also be used to the contour of the character image.

- Extract the contour of the character
- Locate the uppermost and the lowermost points of the contour
- Calculate the in and out profiles of the contour

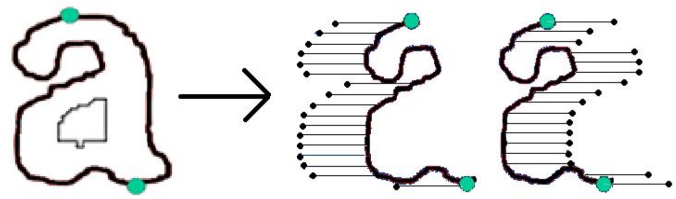

Figure 21: Contour of Character

c. Crossing and Distance

Crossings count the number of transitions from background to foreground pixels along vertical and horizontal lines through the character image and **Distances** calculate the distances of the first image pixel detected from the upper and lower boundaries, of the image, along vertical lines and from the left and right boundaries along horizontal lines

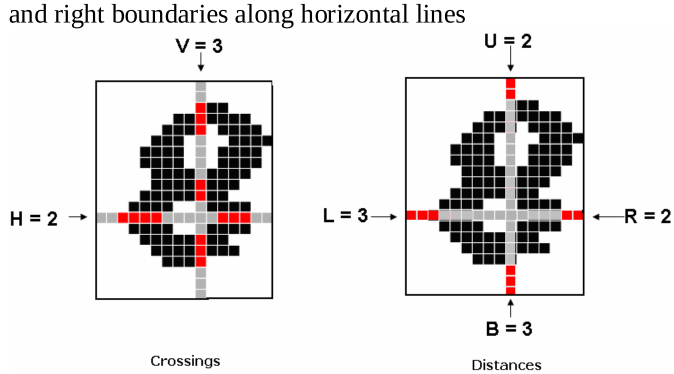

Figure 22: Crossings and Distances

d. Object enhancement:

The quality of plate images varies much in different capture conditions [5]. Illumination variance and noise make it difficult for character segmentation. Then some image enhancement should be adopted to improve the quality of images. The image enhancement methods of histogram equalization and gray level scaling have some side effects. They may have the noise enhanced as well. For character segmentation, only the character pixels need to be enhanced and the background pixels should be weakened at the same time. In fact, a license plate image contains about 20%

character pixels. So these 20% character pixels need to be enhanced and the rest pixels need to be weakened. It is called object enhancement.

The object enhancement algorithm consists of two steps.

- Firstly, gray level of all pixels is scaled into the range of 0 to 100 and compared with the original range 0 to 255, the character pixels and the background pixels are both weakened.
- Secondly, sorting all pixels by gray level in descending order and multiply the gray level of the top 20% pixels by 2.55. Then most characters pixels are enhanced while background pixels keep weakened.

The following figure shows the result of object enhancement [7]. It can be seen from the figure that after object enhancement the contrast of peaks and valleys of the projection is more significant than the original.

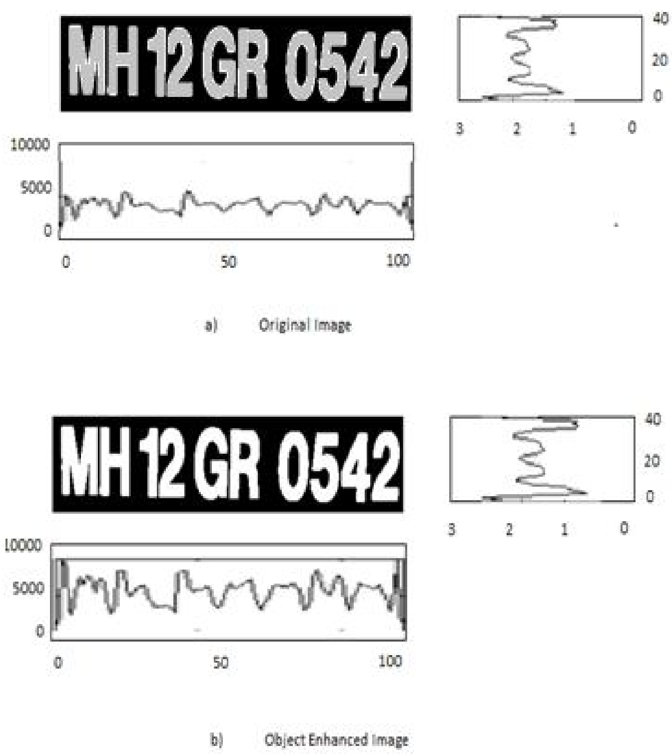

Figure 23: Object Enhancement

2.2 WEB BASED MOBILE APPLICATION

Today, Mobiles are the most easily available communication devices. Traffic violation enforcement and fine collection system can be automated by using the mobile based system. The advantage of this system is faster processing and increased transparency to tackle corruption. This application is web based and can be accessed by policemen by their registered mobiles only. To use this web based facility, the MAC addresses of authorized users are registered in the Police log database to provide additional secured connection to prevent hacking or sniffing attacks.

First of all, user (police) has to log into the system by entering the required fields (fig. 24a). If he wants to lodge a complaint against a vehicle owner, then he must enter License number of the vehicle and search the vehicle details (fig 24b). After verification of identity, he can enter the crime details (fig. 24c).

As a part of this web based service, a separate URL is provided for general public, where one can view his/her own log and file complaints, if any. This system can also contribute to curb corruption. As a real time scenario, if a policeman catches a traffic offender, he has to make an entry in the mobile based application. The higher judicial authorities can also keep a tab on the surveillance process by logging into special accounts created for them and viewing the database.

In this way, the entire traffic surveillance process can be monitored in a systematic manner by the traffic police, general public and higher authorities by using their own systems.

(a) **(b)** **(c)**

Figure 24: User Interface for mobile based system

Technology Used:

As the Web based system is to be used on mobile devices, it needs to me made compatible with them. The DOCTYPE is thus therefore focused. XHTML MP (Mobile Profile) is used to build the web pages.

Front End :

Language used at the front end in mobile phone is XHTML. Mostly all browsers can process a page written in XHTML. The Syntax used in XHTML is similar to HTML but syntax is more emphasized in XHTML. As Syntax is more emphasized the output is clear and error free.

Back End :

Language used for processing at the server i.e at back end is ASP (Active Server Pages). Since it is independent of language used at front end, it makes the system detachable.

Security :

Since the password of each member is stored in the database, the database is vulnerable to attacks. In order to secure the members profile, passwords are encrypted before storing into the database. The encryption/decryption algorithm used here is our own developed and hence difficult to crack unless the algorithm is exposed.

For example a password as ‘getslpassword’ is encrypted to ‘xv…„}□r„„ˆ€ƒu’

Consider a scenario, where a traffic police catches a person for violating traffic rules. He accesses the system, enters his ID and Password. On authentication, he will enter the License Number of the Vehicle to search for Registration details. After verification of identity of the person, he will enter the Code of crime along with the Receipt Number and amount of fine collected from the person. After submitting the details, entries are stored into the database and he can log out. Further while login the IP address of the remote client is also saved for providing the proof. Another page is hosted for those people who want to complain against some corrupt police officers who tried to bribe them, the name of the police officers are tracked using the same database and some intelligent techniques in order to make the country corrupt free.

Since the Web based system is to worked out on mobile hence data received/sent is another issue that is to be minimized in order to reduce the browsing charges charged by respective service providers. Following is the per session transaction in bytes

DETAILS OF	DIGITS ENTERED
BadgeID	4
Location	12
Password	4
Vehicle License Number	10
Code Of Crime	4
Fine Collected	4
Receipt Number	5

The total bytes Sent from mobile device to the server is around 318 bytes and received from server to the mobile is around 4.7 kb, hence a total of approximately 5 kb of browsing is done in an authentic access. Since this amount is too less so most of the service providers usually won’t charge, hence access is free anytime.

2.3 DATABASE SCHEMAS

2.3.1Track Log

The database is common to desktop and web based applications [6]. The desktop based application makes entry of all the vehicles passing by in a database called the track log. The attributes of track log are Vehicle license number, location, time and image. The track log can further be used by authorities for carrying out the regular surveillance activities.

2.3.2Search Log

The search log consists of entries searched by the traffic police. This log can further be used in case of any complaints filed by a vehicle owner claiming the charges of bribery or other corrupt practices against the policemen.

2.3.3Crime Log

The Crime log contains crime details of the traffic offenders, as entered by the concerned traffic police. The attributes of this schema are License Number, Code of Crime, Receipt Number and the Fine Amount.

2.3.4Original RTO database

This database contains details of all the registered vehicle owners, as maintained by the concerned Regional Transport Office (RTO).

2.3.5Police Login Database

This database contains details of the police, who is authorized to file complaints using the web based mobile system. The attributes of this database are: badge number, location and password. The MAC address of the associated mobile is registered with the Police Login Database to provide secured connection.

As a large number of traffic crimes are committed every day, it is difficult to store and maintain such a large amount of data. A threshold period is decided, which is the maximum duration for which to maintain the data of a particular date. A few summarizing attributes are stored in the database. The details will be removed from the database after the threshold period. This Summarized database file is useful for data mining and statistical analysis of data. Also original details in that duration will be compressed and stored in permanent storage drives so that it may be used after longer time.

3. TESTS AND FINAL CONSIDERATIONS

The position of the camera significantly affects the quality of captured images, and the success of the recognition process. A suitable position of the camera towards the lane can lead to a better set of snapshots. In some situations, we can avoid getting skewed images by suitable positioning of it. Let *S* be a representative set of all snapshots, which can be captured by a concrete instance of a camera. Some of the snapshots in this set can be blurred, some too small, too big, too skewed or too deformed. Because of this, the whole set is divided as follows:

$$S = S_c \cup S_b \cup S_s \cup S_e \cup S_l$$

Here, S_c is a subset of “clear” plates, S_b is a subset of blurred plates, S_s is a subset of skewed plates, S_e is a subset of plates, which has a difficult surrounding environment, and S_l is a subset of plates with little characters.

Weighted score

If $\mathbf{P}(r)$ is a plate number recognized by machine, and $\mathbf{P}(c)$ is the correct one, then weighted score S_w of plate $\mathbf{P}(r)$ is given as:

$$s_w\left(\mathbf{P}^{(r)}\right) = \frac{\left|\left\{p_i^{(r)} \,\middle|\, p_i^{(r)} = p_i^{(c)}\right\}\right|}{\left|\left\{p_i^{(r)}\right\}\right|} = \frac{m}{n}$$

where *m* is the number of correctly recognized characters, and *n* is the number of all characters in plate.
For example if the plate "MH12GR0542" has been recognized as "MH12GR8542", the Weighted Correctness score for this plate is 0.9.

4. RESULTS

Following table shows recognition rates, which have been achieved while testing on various sets of number plates. According to the results, this system gives good response to clear plates, because skewed plates and plates with difficult surrounding environment cause significant degradation of recognition abilities.

	Total number of plates	Total number of characters	Weighted score
Clear plates	68	470	87.2
Blurred plates	52	352	46.87
Skewed plates	40	279	51.64
Average plates	177	1254	73.02

Table 1: Recognition Rates

5. CONCLUSIONS AND FUTURE RESEARCH

GETS provides a holistic approach, wherein the entire traffic surveillance process can be monitored in a systematic manner by the traffic police, general public and higher authorities by using their own systems. The system automates traffic surveillance and addresses most of the problems faced in the Indian scenario. It is definitely a better alternative to the existing manual systems. It also attempts to curb corruption at various levels. Currently there are certain restrictions on parameters like speed of the vehicle, script on the number plate, skew on the image, etc. A lot of research needs to undergo to enhance the algorithms further. There should be a self learning system, which will improve its detecting accuracy with experience. In this way, a more trained system will be able to detect most kind of images.

6. REFERENCES

[1] Prathamesh Kulkarni, Kushal Shah, "*Automatic Number Plate Recognition System for Indian Condition*", IEEE 2009.

[2] C. Nelson Kennedy Babu, Krishnan Nallaperumal, "*An Efficient Geometric Feature Based License Plate Localization and Recognition*", IJISE, GA, USA, April 2008, Vol.2.No.2.

[3] Jaedokim, Youngjoon Han, Harnsoo Hahn, "*Character Segmentation Method for License Plate with Topological Transform*", World Academy of Science, Engineering and Technology 56, 2009.

[4] Hong Shan Neoh, Ashar Hazanchuk, "*Adaptive Edge Detection for Real-time Video Processing Using FPGAs*", Altera 200.

[5] Deepak Kumar Gupta, Siddharth Kandoi, "*Automatic Vehicle License Plate Recognition System*", Indian Institute of Technology, Kanpur, 2009-10, CS-676:Image Processing and Computer Vision.

[6] Charlotte Van Ooijen, "*Legitimacy Issues Regarding Citizen Surveillance: The Case of ANPR technology in Dutch Policing*",The Annual Conference of European Group on Public Administration, 2-5 September, 2009.

[7] Feng Wang, Lichun Man, "*Fuzzy Based Algorithm for Color Recognition of License Plate*", Pattern Recognition Letters 29 (2008) 1007-1020.

[8] Cheng-Lin Liu and Xiang-Dong Zhou, "*Online Japanese Character Recognition using Trajectory Based Normalization and Direction Feature Extraction*", National Laboratory of Pattern Recognition (NLPR), Institute of Automation, 2009.

[9] Giorgos Vamvakas, "*Optical Character Recognition for Handwritten characters*", National Center for Scientific Research "Demokritos", Athens, Greece- 2009.

[10] Darcy Bullock and Chris Hendrickson, "*Roadway Traffic Control Software*", IEEE transactions on Control, Systems Technology, Vol.2, No.3, September 1994.

[11] Ghassen Hamarneh, Karin Althoff, Rafeef Abu-Gharbieh, "*Automatic Line Detection using Hough Transform*", Image Analysis Group, Department of Signals and Systems, CHALMERS University of Technology, Lund, May 5-7 and August 19-20, 1999.

[12] Aamir Mohd YASIN, Mohd Rehan KARIN, Ahmad Saifisul ABDULLAH, "*Travel Time Measurement in Real-Time using Automatic Number Plate Recognition for Malaysian Environment*", Journal of the Eastern Asia Society for transportation Studies, Vol.8, 2009.

[13] Ziv Yaniv, "*Median Filtering*", School of Engineering and Computer Science, The Hebrew University, Israel, 2009.