Automatic License Plate Recognition in Real Time Videos using Visual Surveillance Techniques

Lucky Kodwani , Sukadev Meher
Department of Electronics & Communication
National Institute of Technology Rourkela, Rourkela, Odisha, India
lucky.kodwani27@gmail.com, s.mehar@nitrkl.ac.in

Abstract— In this paper we present full-featured vehicle detection, tracking and licence plate recognition system. It consists of vehicle detection, licence plate extraction and a character recognition module. Here first foreground estimation is done by Gaussian mixture model then proposing a real time and robust method of license plate extraction based on block variance technique. License plate extraction is an important stage in license plate recognition for automated transport system. The Extracted license plates are segmented into individual characters by using a region-based approach. The recognition scheme combines adaptive iterative thresholding with a template matching algorithm. The method is invariant to illumination and is robust to character size and thickness, skew and small character breaks. The major advantages of our system are its real-time capability and that it does not require any additional sensor input (e.g. from infrared sensors) except a video stream. We evaluate our system on a large number of vehicle images. Experimental results demonstrate the great robustness and efficiency of our method.

***Keywords* — Background estimation; License plate recognition; *Surveillance System;* Tracking; Vehicle detection; Video indexing;**

I. INTRODUCTION

There is a need for intelligent traffic management systems in order to cope with the constantly increasing traffic on today's roads. Here, traffic data may come from different kinds of sensors. The use of video cameras (many of which are already installed to survey road networks), coupled with computer vision techniques, offers an attractive alternative to other traffic sensors [1]. Video based traffic surveillance is an important part of such installations. Information about current situations can be automatically extracted by image processing algorithms. Vehicle detection and tracking, identification via license plate recognition is important for a variety of applications. Most license plate recognition systems in operation today use special hardware like high resolution cameras or infrared sensors to improve the input quality and they operate in controlled settings. A different solution, as proposed in this work, is the continuous analysis and consideration of subsequent frames. However, this implicates that enough frames are captured by the capturing device and are processed by the processing engine. We state that for the tasks of vehicle detection and subsequent license plate recognition real-time is a flexible term. As about 20 fps (frames per second) might be enough real time for this tasks when cars are driving at residential speeds, this is insufficient for country roads and highway traffic. In our terminology, real-time operation stands for fast enough operation in order to not miss a single object that moves through the scene, irrespective of the object speed. License Plate Recognition (LPR) systems have wide range of applications such as access control, tolling, border patrol, traffic control, finding stolen cars, etc. Additionally, this technology does not need any installation on car, such as transmitter or responder.

In general, a system consists of a camera, a frame grabber, a computer, and custom designed software for image processing, analysis and recognition. It should (1) operate indoors and outdoors, (2) operate in a wide range of illumination conditions, (3) be invariant to size, scale, and font boldness variations, (4) be robust to broken strokes, printing defects, and other kind of noise, (5) be insensitive to camera car relative position within a reasonable distance interval, (6) provide a real-time response, (7) work with alternative image acquisition equipment, as well as with pre-captured and archived images [2].

Experimental results demonstrate the great robustness and efficiency of our method.The rest of the paper is organized as follows: In the next section, an overview of the complete system is given. Section III introduces the vehicle detection and tracking part based on Gaussian mixture model. In Section IV license plate are extracted using block variance technique. In Section V, character of license plate are segmented using optical character recognition (OCR) technique .This approach is based on pattern recognition principles using template-matching algorithm. In Section V, a numerical evaluation of the performance and the computational efficiency of the algorithms are performed. Finally conclusions are drawn in Section VII.

II. Overview of the proposed model

We propose efficient real-time automatic license plate recognition (ALPR) framework, particularly designed to work on video footage. At present, vehicle detection, tracking and licence plate recognition are reasonably well-tackled problems with many successful commercial solutions being available. This paper proposes a novel approach for efficient localization of license plates and the use of a revised version of an existing technique for detection and tracking.

AUTOMATIC NUMBER PLATE RECOGNITION SYSTERM (ANPR) in dynamic scenes is used to detect, recognize and track vehicle from incoming video frames then extract the license plate from it. It has found numerous applications as wide as possible such as: access control in security sensitive areas, securities for communities and important buildings, detection of military target areas, traffic surveillance in cities and highways, detection of anomalies behavior and among many other applications.

The main task in most visual surveillance systems includes motion detection, object classification, tracking, activity understanding and semantic description. Every visual surveillance systems start with detecting moving object in video streams.

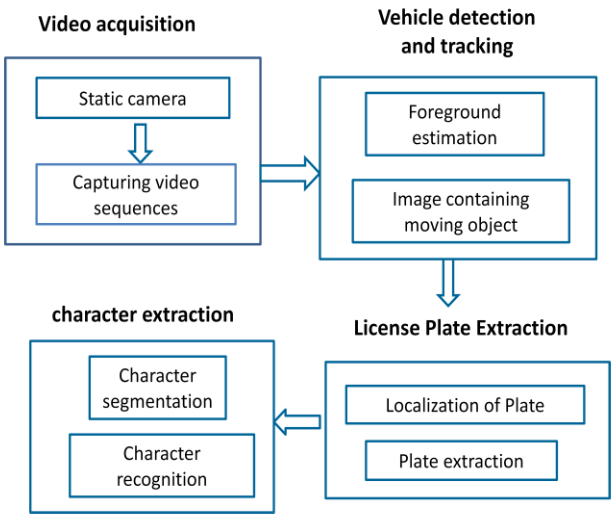


Fig. 1 Structure of ALPR system

Proposed ALPR system is carried out in following sequence as shown in Fig. 1.

- Video Acquisition – video are taken by the static camera situated at traffic scenario.
- *Vehicle Detection And Tracking* – foreground is estimated and moving object is tracked in incoming video frames.
- *License Plate Extraction* - licence plates are first located in current frame then they are extracted.
- *Character Extraction* - extracted plates are the input to this module, here characters are first segmented and then recognized.

Segmented and then recognized.

III. Vehicle Detection And Tracking

A large literature exists concerning moving object detection in video streams and to construct reliable background from incoming video frames. We are using Gaussian mixture model (GMM) for segmentation of moving foreground object from the background [3]-[5].

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

$$P(X_t) = \sum_{i=1}^{k} w_{i,t} * \varphi(X_t, \mu_{i,t}, C_{i,j}) \quad \text{.........} \quad (1)$$

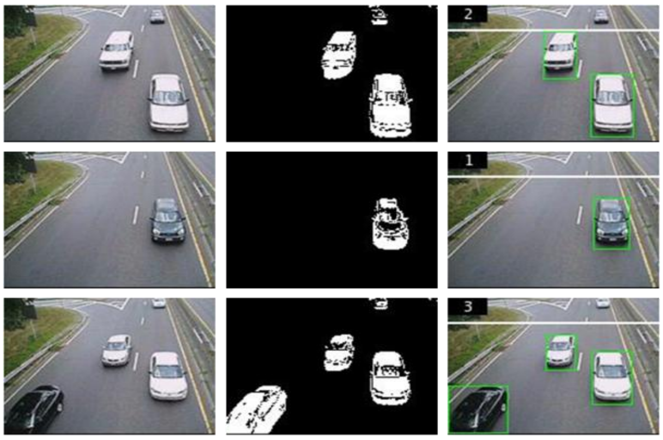
Where $w_{i,t}$, $\mu_{i,t}$ and $C_{i,j}$ are the estimate weight, mean value and covariance matrix of i_{th} Gaussian in the mixture at time t. $\phi(X_t, \mu_{i,t}, C_{i,j})$ is the Gaussian probability density function equation (2)

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

The vehicle detected and tracked by means of following step operations -

step 1: Initialize the required variables and System objects.

step 2: Read video from avi file.



(a) (b) (c)

Fig.2 Examples of moving foreground extraction (a) Original images, (b) Backgrounds Constructed, (c) vehicle tracking.

step 3: Convert the image from RGB to intensity format.

step 4: Detect foreground using Gaussian mixture models.

step 5: A blob analysis to segment cars in the video.

step 6: Drawing the bounding boxes around detected cars configure a System object to write the number of cars being tracked.

step 7: Display the results as shown in Fig.2.

IV. License Plate Extraction

The task “license plate extraction” is considered as the most crucial stage in the whole ALPR system. In the past, a number of techniques have been proposed for locating the plate through visual image processing. The major features used for license plate detection include colors [6], venial edges, symmetry [7], and corners and so on. For examples, K. K. Kim al. [6] used colors and neural networks to extract license plates from images. However color is not stable when lighting conditions change. The major problem in these approaches is the used features depend strongly on the intensity differences between the extracted license plate and car colors, which are not stable at different changes of lighting conditions and view orientations.

This paper proposed a real time and robust method of license plate extraction. License plate area contains rich edges and having highest variance changes as compare to their background so we are using this property to extract the plate region. We first extract out the edges of the car image using image enhancement and Sobel operator, then remove most of the background and noise edges by an effective algorithm, and finally search the plate region and segment the plate out from the original car image.

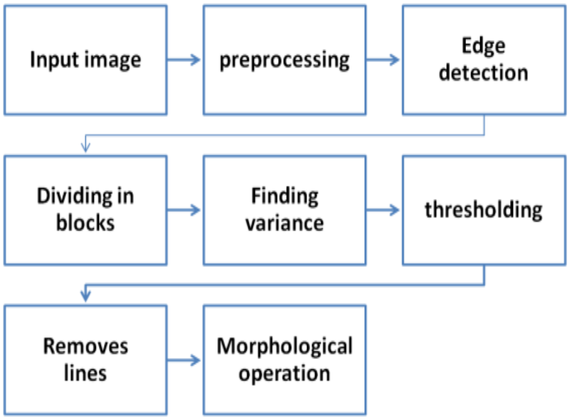


Fig.3 Flow chart of proposed block variance method.

Here we are using block variance technique for extraction of licence plate. As shown in Fig 3.

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

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

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

step 4: The license plate of the car consists of several characters (such as Latin letters, Arabic numerals, etc.), so the plate area contains rich edge information. We used Sobel edge detection Technique to detect the edges of the image.

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

step 6: Calculating threshold variance using (3).

$$V_{th} = \frac{V_{max}+V_{min}}{2} \quad .... (3)$$

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

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

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

step 9: Performing morphological operation and regionprops functions of MATLAB for getting licence plate. Results are shown in *Fig. 4*.

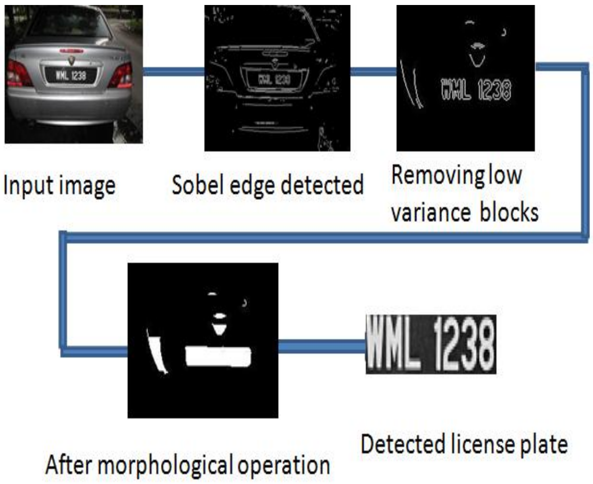


Fig. 4 Results of proposed block variance algorithm.

V. Character Extraction

A. Character Segmentation

Character segmentation is an essential preliminary step in most automatic pictorial pattern-recognition and scene analysis problems. Segmentation subdivides an image into its constituent regions or objects. Segmentation accuracy determines the eventual success or failure of computerized analysis procedures. For this reason, considerable care should be taken to improve the probability of rugged segmentation.

Character segmentation is an important step in License Plate Recognition system. There are many difficulties in this step, such as the influence of image noise, plate frame, rivet, the space mark, and so on.

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

In order to maximize the accuracy of the system format of the Registration number is considered.

Format of Registration Number

Because this paper deals with the Recognition of number plates in India, the general format for only the vehicles in this country is considered. The new format of the Registration, which is in action since 1980, is as shown below:

Case (i): SS DD AA NNNN

Where SS is the two letter state code; DD is the two digit district code; NNNN is the unique license plate number and AA are the optional alphabets if the 9999 numbers are used up. The National Capital Territory of Delhi has an additional code in the registration code:

Case (ii): SS DD C AA NNNN

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

Classification of Characters in License Plate

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

B. Optical Character Recognition(OCR)

The OCR technique is used in order to recognize different digits. This approach is based on pattern recognition principles. The system of OCR engine is based on a template-matching algorithm. The shapes identified as possible characters in the previous steps of the process are binarized and scaled to match the size of the templates in a database, i.e. 42x24 pixels. Each shape is compared with all the characters in the database by template matching and the best match among them is selected. Template matching algorithm gives better results if the source is cooperative.

Template shape

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

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

Each template is of size 42x24 pixels. Each of these templates is stored in the database and they are retrieved for comparison of templates during character recognition process.

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

Template Matching using Euler Number

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

$$E = C - H \qquad \ldots\ldots\ldots\ldots\ldots\ldots \text{(4)}$$

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

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

i) Characters without holes:

Euler number = 1

ii) Characters with one hole:

Euler number = 0

iii) Characters with two holes:

Euler number = -1

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

The following algorithm has been used to implement image to text conversion using template matching.

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

Step2: The scanned image is first loaded and after converting the input image to binary, each character from the image is detected by the technique of segmentation. First, the image is cropped to fit the text. After that, line by line the image is cropped

Step3: Thereafter in each line word by word the image is cropped, followed by cropping of each character in a word to fit the text.

Step4: Each character which is detected is resized to the size of template window (42x24).

Step5: After resizing the character, correlation coefficient for each template with the character is found and the correlation coefficient values are stored in a matrix.

The main operation used for the classification was the two-dimensional correlation. This operation gives a value of the similarity between two matrices (images). Corr2 function develops this operation according to the equation below

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

Step6: The index of the best match is stored as the recognized character. After recognizing the first character the next character is taken and thus after recognizing the first line, the next line is taken, and procedure from step3 is repeated until the last line detected is empty.

VI. Quantitative Performance Analysis

In this section, a more technically accurate performance study is conducted, making use of some segmentation quality metrics for quantifying the correctness of the foreground detection provided by algorithm.

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

1) true positives (TP): correctly classified foreground pixels;
2) true negatives (TN): correctly classified background pixels;
3) false positives (FP): incorrectly classified foreground pixels;
4) false negatives (FN): incorrectly classified background pixels;

Table 1
Typical confusion matrix in a binary decision problem

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

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

TP rate : $\mathrm{TPR} = \frac{\mathrm{TP}}{\text{total of actual positives}} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$

FP rate : $\mathrm{TPR} = \frac{\mathrm{FP}}{\text{total of actual negatives}} = \frac{\mathrm{FP}}{\mathrm{TN+FP}}$

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

Recall : RE = TPR.

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

Jaccard coefficient: $\mathrm{S_J} = \frac{\mathrm{TP}}{\mathrm{TP+FP+FN}}$, $0 \le \mathrm{SJ} \le 1$

Yule coefficient: $\mathrm{S_Y} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} + \frac{\mathrm{TN}}{\mathrm{TN+FN}} - 1$, $(-1 \le \mathrm{SY} \le 1)$

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

$$\mathrm{E_y} = \sqrt{(\gamma\mathrm{FPR})^2 + (1-\gamma)(1-\mathrm{TPR})^2}$$

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

$$\mathrm{S} = \{ \mathrm{S_F}, \mathrm{S_J}, 0.5(1+\mathrm{S_Y}) \} \quad , \quad \mathrm{E} = \{ \mathrm{E_{0.25}}, \mathrm{E_{0.50}}, \mathrm{E_{0.75}} \}$$

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

Table 2
Quantitative performance analysis

Frame no	$\mathbf{S_F}$	$\mathbf{S_J}$	0.5 (1+$\mathbf{S_Y}$)},	$\mathbf{E_{0.25}}$	$\mathbf{E_{0.50}}$	$\mathbf{E_{0.75}}$
22	0.64	0.53	0.72	0.32	0.29	0.24
75	0.57	0.61	0.63	0.34	0.30	0.20
94	0.53	0.47	0.64	0.40	0.34	0.27
116	0.42	0.45	0.59	0.53	0.41	0.35

VII. Conclusion

Here we have proposed a system for automatic vehicle detection and tracking to overcome the problem of background motion and varying illumination condition. The algorithm works fine with all kind of video data set taken. The system uses no colour information and works on the grayscale video imagery. The proposed algorithm is based on reducing spatial and intensity resolution. So this system requires less data storage for storing the video files in the memory. Further, we may have a very straightforward implementation of the proposed system in VLSI chip as it uses less space, less power and fewer components for VLSI design, resulting in low-cost system. Thus our system is a very cost effective as well as a robust and efficient object tracking system.

REFERENCES

[1] Hu et al. ”A Survey on Visual Surveillance of Object Motion and Behaviors” .IEEE Trans. On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 34, No. 3, August 2004

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

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

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

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

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

[7] S. Kim and S. Chien, "Automatic car license plate extraction using modified generalized symmetry transform and image warping," IEEE International Symposium on industrial electronics. vol. 3. pp. 2022-2027.2001

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

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

[10] Chris Stauffer, W. Eric, and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22:747–757, 2000.