Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(5):1734-1742

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

A robust license plate detection algorithm based on multi-features

Shangzhi Xu

Electronic and Information Engineering Department in Tongji University, Shanghai, China

---

ABSTRACT

This paper presents a robust vehicle license plate detection algorithm based on multi-features, including mathematical morphology, rectangle features, edge statistics and characters features. Here we utilize the word "robust" to describe the proposed algorithm because it is not only adaptive to variance occasions, such as variance of the illumination, vehicle position, the color, both background and foreground, the acclivitous angle and the size while working in different complex environment, but also can be used in several regional license plates while some other algorithms just work well for one region and badly for another. We have used the algorithm for detecting both American and Chinese plates and gotten a good performance in both regions. Those are the two primary contributions of the algorithm. Another contribution is that we use a character feature verification algorithm to determine the final detection result in the candidate rectangles. This is proved to be more effective than other features.

Key words: license plate detection, morphology, edge statistics, rectangle features, characters features

---

INTRODUCTION

In the last decade, intelligent transportation systems have had a wide impact in people's life and lots of emphasis had been given to the license plate recognition (LPR) system, since LPR has many applications in intelligent infrastructure, including arterial management systems for traffic surveillance, electronic payment systems (toll payment, parking fee payment) without the human interruption, and so on.

Usually, a LPR system consists of three parts, which are the license plate detection, the character segmentation, and the character recognition. Among these three, correctly detecting the position of the vehicle license plate is the most important and basic part, which directly affects the system's overall accuracy. To make the extraction process successful and robust, the algorithm must be adaptive to lots of conditions such as the poor image quality from the uneven lighting condition and various observation angles from the vehicles and cameras, as long as they don't come to extremeness.

Attention has previously been focused on this scope for a long time, and a lot of work has been done for license plate recognition. Many techniques mentioned above have been used, such as Morphological operations [1][2][3], edge extraction [4][5]. And Techniques based upon combinations of mathematical morphology and edge statistics [6]–[9] feature very good results. Gradient magnitude and their local variance in an image are computed in the methods. And block-based processing is also supported. Then, regions with a high edge magnitude and high edge variance are identified. The advantage is that it can be applied to an image with unclear license plate boundary and can be implemented simply and fast since this method does not depend on the edge of license plate boundary. However, edge-based methods alone can hardly be applied to complex images, since they are too sensitive to unwanted edges, which may also show high edge magnitude or variance. In spite of this, when combined with other features, the license plate extraction rate is relatively high and fast compared.

In reference [10], a fast algorithm for detecting license plate in various conditions is proposed. Both statistical features and Haar-like features are used in the algorithm. The classifiers based on statistical features are trained through simple learning procedures. Then AdaBoost learning procedure is used to select important Haar-like features and construct classifiers. The final cascade classifier is obtained by combining the above two kinds of classifiers. At the same time, the authors propose a new vertical edge map which only keeps the edge points with vertical edge angles and so the algorithm eliminates more background regions from further detecting, which makes the algorithm very fast.

A lot of other methods are also analyzed in [11] and the primary contributes it presented focus on the segmentation step of LPR. However, it still valuable due to the detailed analysis and compare with different methods, not only in the license plate detection, but also in the segmentation and recognition.

In our approach multi-features are proposed including the above mathematical morphology and other features such as the rectangle and characters features. Organic usage of the combined multi-features can efficiently solve some problems that caused by only one feature. Coincident characters verification applied to the candidate rectangles can make a better performance than other features. The experimental results are presented from the application of our approach to both the Chinese and American plates to demonstrate the viability, effectiveness, and robustness of our algorithm.

The paper is organized as follows. The next section utilizes the mathematical morphology to form the primary rectangles. In Section III, rectangles and edge features are introduced into to make a precise location. Then coincident characters verification is proposed to obtain the final result in section IV. Experimental results are presented in Section V, and finally, conclusions and future extensions are presented in Section VI.

II. MORPHOLOGY LOCATION

The flow chart of mathematical morphology algorithm can be described as Fig. 1 below:

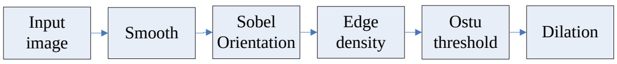


Figure 1. Mathematical morphology location

There are many methods of performing edge detection. But for the image that contains license plate may also include the dynamo and fore-baffle of vehicle, etc, which have very strong horizontal edges. These edges have huge impact on the license plate localization. So we have to suppress the horizontal edges and extract the vertical ones. Before vertical edge detection, a linear filter is used to smooth the image and apply the illuminance normalization to reduce the influence of light.

We can obtain the vertical gradient magnitude map by the Sobel operator Sx, calculated by (1).

$$\begin{aligned} Sx(i,j) =& |[f(i-1,j-1)+2f(i,j-1)+f(i+1,j-1)] \\ & -[f(i-1,j+1)+2f(i,j+1)+f(i+1,j+1)]| \end{aligned} \tag{1}$$

But that isn't enough. In our algorithm, whether a gradient is vertical or not is judged by not only its magnitude but also its edge orientation angle. Suppose Gx is the gradient obtained from Sx, Gy is the gradient obtained from Sy respectively, calculated by (2).

$$\begin{aligned} Sy(i,j) =& |[f(i-1,j-1)+2f(i-1,j)+f(i-1,j+1)] \\ & -[f(i+1,j-1)+2f(i+1,j)+f(i+1,j+1)]| \end{aligned} \tag{2}$$

And then the edge angle is defined as (3):

$$\theta = \arctan\frac{Gx}{Gy} \tag{3}$$

Only those gray values of pixels whose edge angle are more than 45 degree and less than 135 degree are kept, the others are set to zero no matter how big its magnitude is. Fig.2 shows the result of the step.

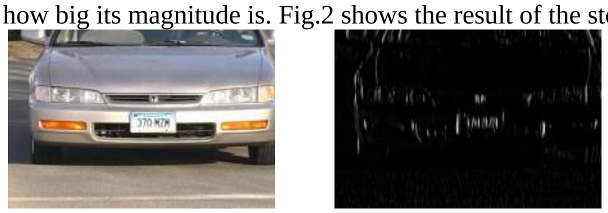


Figure 2. (a) Original image **(b) Vertical edge map**

LP tend to have a high density of edges. So we can measure the edge density by summing all edge pixels using the following equation in $col \times row = N$ block, which is center at g(i, j),shown as (4).

$$d(i,j) = \frac{1}{N}\sum_{col}\sum_{row} g(i,j) \tag{4}$$

Here, g(i, j) is the magnitude of vertical gradient and in our algorithm we choose a 3×15 block, depending on the real size of the plate region. The result of edge density is shown in Fig.3. The OTSU method is used to threshold the density map and the result is shown as Fig.4.

Figure 3. Edge density map

Figure 4. OTSU threshold

Then we dilate the image use a horizontal mask. The size of the mask is 1×11. This step can join the small closed blocks to a larger one, which will be helpful to the next step. Thus, the dispersive object pixels are combined to large shape of blobs. The mask is larger than other algorithm [9], because here we want to ensure the plate will be connected as one. Actually when the algorithm is applied to the Chinese plate, larger mask (1×13) is utilized since the Chinese plate is longer than the American one. We can effectively remove the redundant blobs caused by larger dilation using other features below. After that, classical blob analysis algorithm is utilized to obtain the shape of the blobs, and discard some small blobs being considered as noise. Fig.5 shows the result after dilation and blob analysis.

Figure 5. (a) Dilation (b) Blob analysis

III. PRECISE LOCATION

The connected component analysis algorithm is applied to the processed images and then we can obtain some candidate rectangles to identify the plate. However, lots of these candidates are illegal. In this section we will propose rectangles and edge features in order to delete some non-plate rectangles and modify the possible plate to a precise size and shape.

The rectangles and edge features cooperate with each other and make a good performance in the precise location here. They can be described as follows:

Rectangles' R/A/D: With the important information from the CCA, some features of region, such as the aspect ratio (R), the area (A) and the density (D) of region are applied. Let R denote the region of rectangles with width W and height H, and let N denote of the number of the object pixels in the rectangles. The R/A/D can be calculated as (5):

$$\begin{cases} R = W/H \\ A = W \times H \\ D = N/(W \times H) \end{cases} \tag{5}$$

Combine Rectangles: After getting the rectangles, they are combined because of the unevenness. The two rectangles that are close enough or intersected with each other, and have near location are combined to one, since some characters in the American plates may be a little farther than others. In this way, we won't lose these characters. Of course, features R/A/D will be calculate again to be as the weight of the candidates.

Vertical edge line: Since the characters in the license plate have enough vertical edge information, we can utilize them to make a more precise detection in the size and shape and even eliminate some illegal rectangles with little vertical edge information. The vertical edge line map can be obtained as follows:

After obtaining the Sobel map, we use the OTSU algorithm to Threshold the map. In this binary vertical edge map, we combine the points to vertical lines. The edge points are named as feature points. These points are scanned in the vertical direction. If the distance of neighbor feature point is less than MAXLENGTH, then the two points form a line. If the distance from the following vertical point to nearest point of above line is also satisfied the above condition, and then it is combined to the line. By analogy, a group of lines are gotten. The density of points is calculated according to (6).

$$pd = \frac{NumPoint}{LineLength} \tag{6}$$

Here, $LineLength$ represents the length of the line and $NumPoint$ represents the number of points in the line. If the $LineLength$ are longer than the MINLINELENGTH and point density in the line is between the MINPOINTDEN and MAXPOINTDEN, then the line is reserved, otherwise is deleted. Fig 6(a) shows the vertical lines in the rectangle.

Then we can analyze the vertical edge lines in a rectangle in order to make a precise detection and eliminate some candidate. We can calculate number of the vertical edge lines in the rectangle to decide whether it is a candidate or not. Some plates without enough vertical lines are eliminated. To make a precise detection, we calculate the line density to filter some illegal lines which doesn't belong to the character. For each line, we calculate its middle to form a horizontal line, and the number of other lines in the rectangle intersect with the horizontal line in the rectangle are considered as the density of the vertical line. Only lines whose density is larger than a certain threshold are preserved and those lines without enough adjacent vertical lines are considered as non-character area. We can utilize these carefully selected lines in the rectangle to form a new, small and precise rectangle. We can easily find that, the new number of rectangles become less, since the rectangles without vertical lines in them are discarded, and the rectangle are become smaller as utilizing the vertical lines as the borders.

Then, the R/A/D (equation (5)) features are considered. Only those rectangles' R/A/D values lie in the reasonable range are preserved as the candidates. Usually, only 1-5 candidates are selected to be verified. Fig.6 (b) shows the result, and compare to Fig.6 (a), most illegal rectangles are deleted and the detection becomes very precise.

Figure 6. (a)Vertical lines in rectangle (b)Precise detection

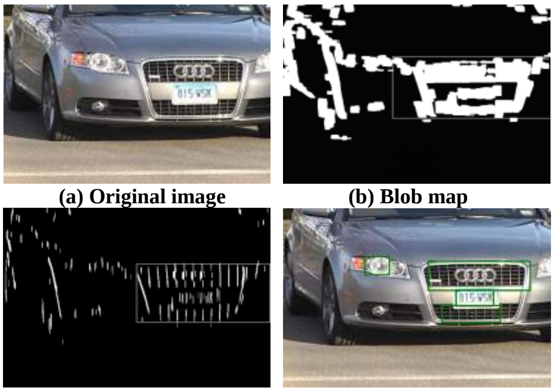
The performance seems very good in this step. But there are still two main problems, though they don't appear in the above series images. One is there may be some very large blob contains plate and also other vertical edge information, such as barrier in the vehicle. We are not able to eliminate those non-character lines because they also have enough vertical lines. Then when utilizing the R/A/D features, the large rectangle formed by the blob will be eliminate since it is too large. The other the horizontal character edge loss in the left and right border since some we just utilize the vertical edge lines as the precise detection criterion. Some characters, such as '4', 'E', 'F' will lose the horizontal edge. We have to propose more approaches to solve them.

Divide large rectangles: We solve the first problem by divide these large rectangles into small ones due to the vertical lines in the large rectangle. We find the number of vertical lines in the large rectangle by horizontal scan. If the number is more than a certain threshold, we mark it as the possible candidate plate area and make a continuous scan with the next row of the image. If the height of the vertical lines is larger than height of the smallest plate to be considered, we will form a small rectangle embodying all of these lines. The new rectangle's height is the continuous rows of the vertical lines and its width is calculated by (7):

$$width = height * MAXRATIO \tag{7}$$

Here, $MAXRATIO$ is the maximum ratio R as defined in equation (5).

The following figures are shown to present the effect of the solution.Fig.7 (a) is the original image and Fig.7 (b) shows the blob and rectangle of it.



(a) Original image **(b) Blob map**

(c) Vertical lines **(d)The later algorithm**

Figure 7. The solution of large rectangle

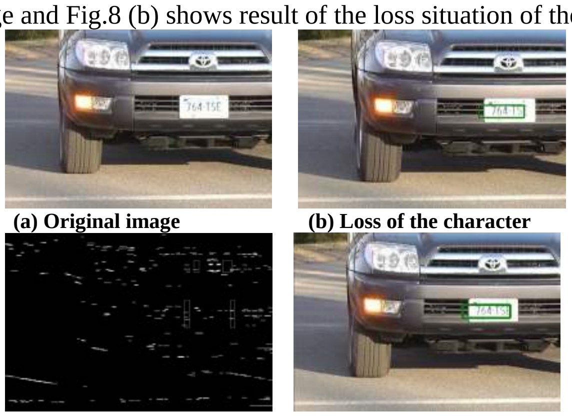
From Fig.7 (b), we can see that, the blob of the plate is connected with the barrier of the car. The algorithm before discards this large blob, so there is no right candidate found in the image. Fig.7 (c) shows the vertical lines in the rectangle. We can find that too many vertical lines are here than we can’t delete some lines to find the plate area. After dividing the large blob due to the vertical lines in the map, the result is shown in the Fig.7 (d). We can find that, the barrier and the plate are divided successfully, and the barrier will be discarded in the next step, leaving only the plate for segmentation.

Finding the loss: We precisely located the border of the plate by vertical lines of the maps especially for the side border. It does a good job if the plate is located at the border of the plate, other than the border of the characters. But sometimes when the plate border is far away from the characters, failing to form a connected blob, location may shifts to the border of the character. In order to solve the problem, we presented an algorithm utilizing the gray density before. We obtain the Sobel vertical gray map at first, then in the area formed by the Pre-Location rectangle’s left and the new rectangle’s left, there gray density are calculated as (8):

$$D_a = \frac{\sum g(i,j)}{H \times W} \tag{8}$$

If D_a is larger than the threshold, the left side of the rectangle will be expand certain pixels to make up the lost character of the left in the plate.

This method is effective when the luminance is steady. But the problem is the threshold and pixels to be expanded can not be precisely selected, especially when the luminance is variable. So we present another method to solve the problems----Finding the horizontal lines (obtained by the Sobel horizontal edge and OTSU threshold) in a small area of the border, in both the two sides. This area is formed as follows: the top and bottom are as the same with the location. One side is the border, and the other is far away from it by 15 pixels toward to the outer of the plate (for example, to the left side, we must form the rectangle with a distance small than the border and to the right, it should be larger than that). Since the character is just partly lost, 15 pixels are enough to include the missing part. We will expand the rectangle if we successfully find the horizontal lines in the area. This method is much more robust and precise than the former.
Fig.8 (a) shows the original image and Fig.8 (b) shows result of the loss situation of the algorithm.



(a) Original image **(b) Loss of the character**

(c) Horizontal lines **(d) Final result**

Figure 8. Find the loss part

We can see the border character “E” loses the right side. Fig.8 (c) is the horizontal edge lines in corresponding small area and shown in Fig.7 (d), we successfully find the loss part due to the horizontal lines.

IV. COINCIDENT CHARACTERS VERIFICATION

We obtain 1-5 license plate candidates above. But usually there is only one plate even sometimes no plate in the image. To distinguish a plate ROI from other ROI, an elementary character detection method for coincident characters verification is utilized. A plate ROI is identified only when it has enough character-like blobs (we consider it more than four) detected.

We use Adaptive-Threshold to obtain a binary image. The approach to find the threshold takes the assumption that smaller image regions are more likely to have approximately uniform illumination, thus being more suitable for threshold. The main idea of the Adaptive-Threshold method is shown below:

We divide an image into an array of overlapping sub-images (windowSize × windowSize) and then find the optimum threshold for each sub-image. The algorithm will consider each pixel at a time, calculate the mean value of the local neighborhood 'window size', shown as (9) and thresholds the current pixel to white if the difference between the calculated mean and the current pixel value is lower than the mean-offset.

$$\begin{array}{l}(x\text{-}windowSize/2,\ y\text{-}windowSize/2,\\ x\text{+}windowSize/2,\ y\text{+}windowSize/2)\end{array} \tag{9}$$

In our algorithm, windowSize is set to 15 pixels and mean-offset is set to 5. Fig.9 (a) is the original image and Fig.9 (b) shows the binary image after Adaptive-Threshold.

(a) Original image **(b) Binary image**

Figure 9. Adaptive-Threshold result

After Adaptive-Threshold, a blob analysis method is then used to detect the character-like regions. The character-like regions are restricted by several conditions, such as the proportion of length and width (range in 1.5~4.5), the scale and the coordinates of their centers. Considering the scale, all the blobs are classified into several categories according to the differences of scale value of each blob. The category gap is 30% (obtained by experimentation) of each category’s mean scale. We only choose the blobs in the same category and center near the middle line. Under these restrictions, only those character-like regions are selected, the number of which can determine whether a region is a plate. We call the characters as coincident characters, since they have nearly the same proportion of length and width, the scale and the center coordinates. Thus the algorithm is called coincident characters verification. It is effective to eliminate those confused candidates with abundant vertical edge information, such as the front lamp of the vehicle and the barrier. This method is much better than other features utilized for selecting the final plate area presented in some other papers.

Fig.6 (b) shows the result of the precise location of the plate. There are three candidates with enough vertical information and adequate R/A/D feature values. The characters analysis and the result of coincident characters verification are shown as Fig.

(a) Candidate with characters

(b) Candidate without characters

(c) Candidate with characters

Figure 10. Coincident characters verification

As shown in Fig.10, there is no character-like region detected in an area (b) and (c), while in area (a) there are seven character-like regions detected, with which we recognize it as a plate area.

(a) Precise location **(b) Verification**

Figure 11. Detection result

The final result is shown as Fig.11, from the precise location, shown as Fig.11 (a) to verification result shown as Fig.11 (b).

(a) Candidates with lamps **(b) Final result**

Figure 12. Distinguish lamp and plate

Fig.12 shows the effect on eliminating front lamp and barrier of this method. Fig.12 (b) is the final result after eliminate the illegal lamp candidates in Fig.12 (a).

To make this method more robust, the long-line-blobs which conjoin the characters must be erased before character-like analysis. Fig.13 (a) shows an example. The long-line-blobs connect with all the characters, which actually makes just one big blob.

(a) Characters and noisy line-blobs **(b) Result of removal**

Figure 13. Long-line-blobs removal

To get rid of these line-blobs, firstly, all the uninterrupted horizontal lines are recorded, and then all the lines which can compose a long line in one direction (decline or horizon) are erased. The result is shown in Fig.13 (b).

Till now, the whole algorithm is detailed presented. To give an overall outline and review of the algorithm, we propose the flow chart of the license plate detection algorithm, shown as Fig.14.

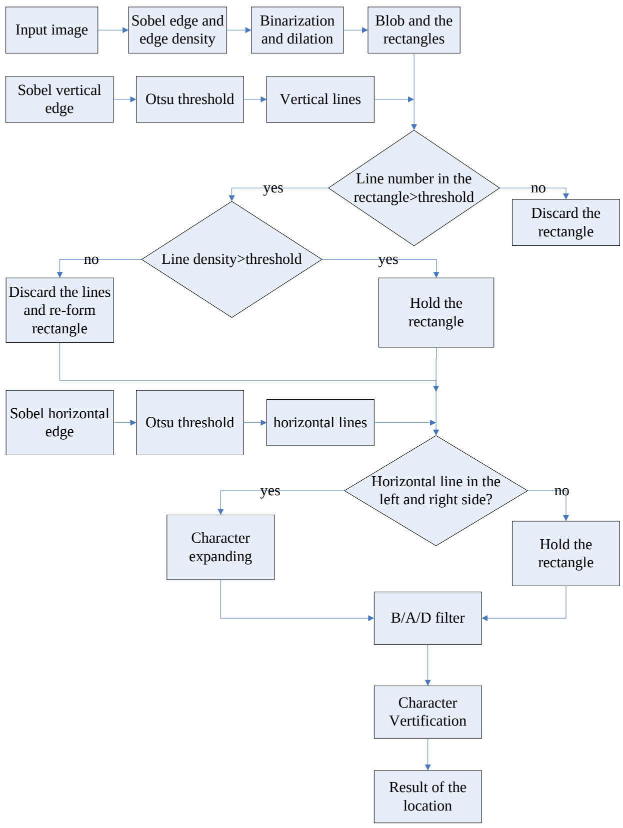


Figure 14. flow chart of the algorithm

V. Experimental Results

The following experiments show that our algorithm is robust to license plates from different regions and nations. We take 2000 Chinese plates and another 2000 of American ones as samples. During which, 1993 Chinese plates and 1986 American ones are successfully detected. The reason of low success ratio of American plates (99.3%) is that they are not as normal and standard as Chinese ones (99.65%), since lots of them are very unique. Further more, results also show that out algorithm are adaptive to variance occasions, such as the color, the acclivitous angle and the size.

Figure 14. American license plate detection results

Fig.14 shows the American and Fig.15 shows the Chinese license plates detection results of our samples. During which, there are images with different illuminance and weather. Meal while, some images with noises of vertical edge information also selected to show the robustness of our algorithm, such as vertical barrier, lamp and plate-like characters existing in the vehicle.

Figure 15. Chinese license plate detection results

CONCLUSION

A multi-features based license plate detection algorithm is presented in this paper. The proposed approach can be divided into three sections, which are, the morphology detection, rectangles and edge features detection, coincident characters verification license plate location. The algorithm gives good results on our database, and it is relatively robust to variations of the illuminance conditions and different kinds of vehicle in different complex environment. From the result of experiment, the algorithm is satisfying and robust. However, there are still lots of work to do in order to solve some problems of unique and personal plate detection. We leave it for the future.

REFERENCES

[1] Hongliang Bai, Junmin Zhu, Changping Liu, "A Fast License Plate Extraction Method on Complex Background", the **2003** IEEE International Conference on Intelligent Transportation System", pp. 985-987, 2003.
[2] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen, "Morphology-based License Plate Detection from Complex Scenes", 16th International Conference On Pattern Recognition, pp.176-179, **2002**.
[3] Fernando Martín, Maite García, José Luis Alba, "New Methods For Automatic Reading of VLP's (Vehicle License Plates)", Signal Processing Patten Recognition and application, **2002**.
[4] Mei Yu, Young Deak Kim, *IEEE Int. Conf. SMC*, vol4, pp.2975-2980, **2000**.
[5] Su-Hyun Lee, Young-Soo Seok, Eung-Joo Lee, "Multi-National Integrated Car-License Plate Recognition System Using Geometrical Feature and Hybrid Pattern Vector", the **2002** International Technical Conference On Circuits/Systems, Computers and Communication.
[6] Last day of access (**2005**, Aug. 10). [Online].
Available: http:// itsdeployment2.ed.ornl.gov/technology_overview/
[7] C. Anagnostopoulos, E. Kayafas, and V. Loumos. (**2000**). *J. Elect. Eng*. [Online]. 1(2), pp. 2–7.
Available: http://www.medialab.ntua.gr/people/canag/journals.php
[8] F. Martín, M. García, and L. Alba, "New methods for automatic reading of VLP's (Vehicle License Plates)," in Proc. IASTED Int. Conf.SPPRA,Jun.**2002**.[Online].Available: http://www.gpi.tsc.uvigo.es/pub/papers/ sppra02.pdf
[9] B. Hongliang and L. Changping, "A hybrid license plate extraction method based on edge statistics and morphology," in Proc. ICPR, **2004**, pp. 831–834.
[10] Huaifeng Zhang, Member, IEEE, Wenjing Jia, Student Member, IEEE, Xiangjian He, Senior Member, IEEE, and Qiang Wu, Member, IEEE. "A Fast Algorithm for License Plate Detection in Various Conditions" Systems, Man and Cybernetics, **2006**. SMC '06. IEEE International Conference. Volume:(3), pp: 2420-2425.
[11] Christos Nikolaos E. Anagnostopoulos, Member, IEEE, Ioannis E. Anagnostopoulos, Member, IEEE, Vassili Loumos, Member, IEEE, and Eleftherios Kayafas, Member, IEEE."A License Plate-Recognition Algorithm for Intelligent Transportation System Applications". Volume(7), pp:377- 392.