Volume 01, No. 01

International Journal of Scientific Engineering and Technology Research

Jul-Dec 2012, P.P. 36-44

www.semargroups.org

Automatic Vehicle Identification by Number Plate Recognition Using Hough Transformation

V.SUNITHA[1], K.DURGA PRASAD[2]

[1]*M.Tech Student of JNCE, Mahabobnagar, AP-India,e-mail:vulapala_sunitha@yahoo.com*

[2]*Assoct Prof,ECE Dept, JNCE, Mahabobnagar, AP-India*

Abstract: Automatic license plate recognition (LPR) plays an important role in numerous applications and a number of techniques have been proposed. However, most of them worked under restricted conditions, such as fixed illumination, limited vehicle speed, designated routes, and stationary backgrounds. In this study, as few constraints as possible on the working environment are considered. The proposed LPR technique consists of two main modules: a license plate locating module and a license number identification module. The former characterized by fuzzy disciplines attempts to extract license plates from an input image, while the latter conceptualized in terms of neural subjects aims to identify the number present in a license plate. In This proposed algorithm consists of three major parts: Extraction of plate region, segmentation of characters and recognition of plate characters. For extracting the plate region, edge detection algorithms and smearing algorithms are used. In segmentation part, smearing algorithms, filtering and some morphological algorithms are used. And finally statistical based template matching is used for recognition of plate characters. The performance of the proposed algorithm has been tested on real images. Based on the experimental results, we noted that our algorithm shows superior performance in car license plate recognition.

Keywords: *Color edge detector, license plate locating, license plate recognition (LPR), self-organizing (SO) character recognition, spring model, topological sorting, two-stage fuzzy aggregation.*

I. INTRODUCTION

Automatic license plate recognition (LPR) plays an importantrole in numerous applications such as unattended parking lots [3], [5], security control of restricted areas [8], traffic law enforcement [7], [33], congestion pricing [5], and automatic toll collection [20]. Due to different working environments, LPR techniques vary from application to application. Most previous works have in some way restricted their working conditions [9], such as limiting them to indoor scenes, stationary backgrounds [30], fixed illumination [7], prescribed driveways [22], [26], limited vehicle speeds [1], or designated ranges of the distance between camera and vehicle [23]. The aim of this study is to lessen many of these restrictions.

Of the various working conditions, outdoor scenes and non-stationary backgrounds may be the two factors that most influence the quality of scene images acquired and in turn the complexity of the techniques needed. In an outdoor environment, illumination not only changes slowly as daytime progresses, but may change rapidly due to changing weather conditions and passing objects (e.g., cars, airplanes, clouds, and overpasses). In addition, pointable cameras create dynamic scenes when they move, pan or zoom. A dynamic scene image may contain multiple license plates or no license plate at all. Moreover, when they do appear in an image, license plates may have arbitrary sizes, orientations and positions. And, if complex backgrounds are involved, detecting license plates can become quite a challenge.

Typically, an LPR process consists of two main stages: 1) locating license plates and 2) identifying license numbers. In the first stage, license plate candidates are determined based on the features of license plates. Features commonly employed have been derived from the license plate format and the alphanumeric characters constituting license numbers. The features regarding license plate format include shape, symmetry [15], height-towidth ratio [23], [25], color [17], [25], texture of grayness [2], [25], spatial frequency [26], and variance of intensity values [8],[10]. Character features include line [34], blob [13], the sign transition of gradient magnitudes, the aspect ratio of

characters [12], the distribution of intervals between characters [28], and the alignment of characters [32].

In reality, a small set of robust, reliable, and easy-to-detect object features would be adequate. The license plate candidates determined in the locating stage are examined in the license number identification stage. There are two major tasks involved in the identification stage, character separation and character recognition. Character separation has in the past been accomplished by such techniques as projection [11], [30], morphology [2], [10], [28] relaxation labeling, connected components [25], and blob coloring. Every technique has its own advantages and disadvantages. Since the projection method assumes the orientation of a license plate is known and the morphology method requires knowing the sizes of characters, these two approaches are not appropriate for our application because of their required assumptions. Relaxation labeling is by nature iterative and often time consuming. In this study, a hybrid of connected components and blob coloring techniques is considered for character separation.

II. Automatic license plate recognition

In this section, the styles of license plate that are considered in this study are discussed, followed by a brief description of the proposed LPR process. The classes include private automobile, taxi, tour bus, truck, and government vehicles. Other categories of vehicles, such as diplomatic cars and military vehicles, are not addressed since they are rarely seen. Styles of license plates can easily be distinguished based on two attributes:

1) the combination of colors used and

2) the compositional semantics of license numbers.

Fig. 1 shows the proposed LPR process. We assume that the process is incorporated in an event detection system, e.g., a vehicle detector or a traffic law enforcement system. Once the system detects an event, the camera along with the system is activated. The image acquired by the camera is then sent to the LPR process, in which potential license plates are extracted from the image. If no license plate is found, the process returns to await another input image. However, oftentimes multiple license plate candidates are detected. They are closely examined at the license number identification stage. There are two essential tasks involved in this stage, character segmentation and recognition. These two tasks are alternatively invoked in order to achieve optimal results for both segmentation and recognition. The characters recovered from a license plate candidate at this stage are next verified at the confirmation stage.

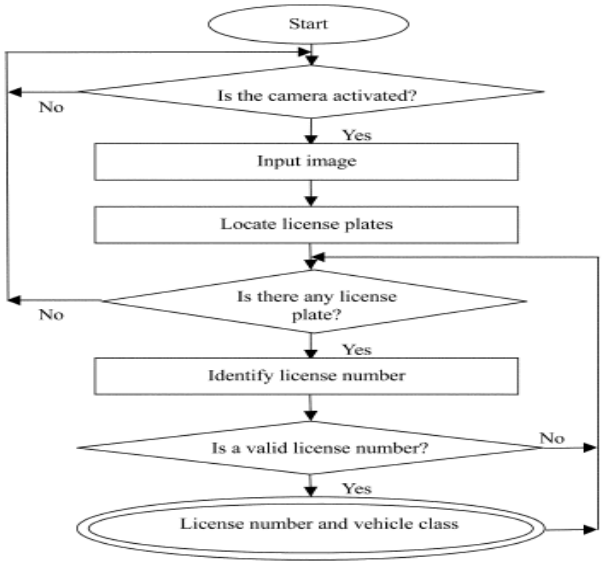

Fig. 1. Diagram of the proposed LPR process.

The group of characters will be deemed to form a valid license number if it agrees with the compositional semantics of license numbers mentioned earlier. Both the valid license number and the associated vehicle class will be returned by the LPR process. The identification and confirmation stages repeat for all of the license plate candidates. Afterwards, the process returns to await another input image.

III. LICENSE NUMBER IDENTIFICATION MODULE

A. Fundamental Idea

Fig. 2 gives the flowchart for the identification module. There are two major components constituting the module preprocessing and recognition. The preprocessing component consists of three tasks, binarization, connected component labelling, and noise removal, which are arranged in sequence. The recognition component is composed of two main procedures, character segmentation and recognition. To obtain optimal results for both the procedures, they are alternatively invoked.

Since the camera may be rolled and/or pitched with respect to license plates, it is desirable that their images be transformed to a predefined size and orientation before performing license number identification. However, without information about relationships between the camera and working environments, the transformations can only be conducted blindly or by trial-anderror. In the proposed method since the transformation step is omitted, it is

inevitable that difficulties in the subsequent steps will increase.

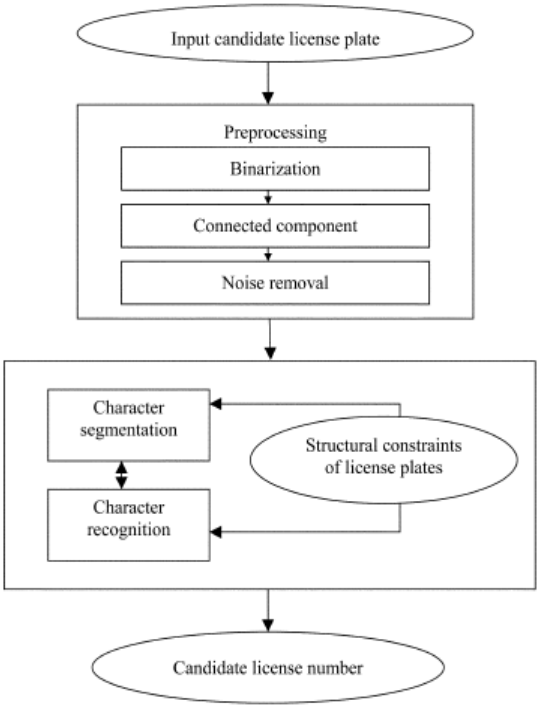

Fig. 2. Flowchart for the license number identification module.

Considering a license plate candidate, it is first binarize. Since some information will somehow be lost during binarization, a variable thresholding technique previously proposed by Nakagawa and Rosenfeld [24] is employed. The technique determines a local optimal threshold value for each image pixel so as to avoid the problem originating from nonuniform illumination. Although variable thresholding cannot completely compensate for the information loss mentioned above, it at least preserves information that may be lost when using a constant binarization method. There are two purposes for the binarization step: highlighting characters and suppressing background. However, both desired (e.g., characters) and undesired (e.g., noise and borders of vehicle plates) image areas often appear during binarization.

In order to eliminate undesired image areas, a connected component algorithm is first applied to the binarized plate candidate. The aspect ratios of connected components are then calculated. The components whose aspect ratios are outside a prescribed range are deleted. Then an alignment of the remaining components is derived by applying the Hough transform to the centers of gravity of components. The components disagreeing with the alignment are removed. If the number of remaining components is still larger than a prescribed number (eight in practice), connected components are deleted one at a time starting with the smallest. Here, we choose eight as the prescribed number because a license number consists of five or six characters and characters may be broken. The removal process continues until either of two conditions is satisfied. Either the number of remaining components equals the prescribed number, or a dramatic change in size from the previously removed component to the current one under consideration is encountered. We assume that noise components are much smaller than characters.

The above procedure does not guarantee that each of the surviving components will correspond to an individual character. A component may be due to noise, an incomplete character, a distorted character, or characters that appear to touch. To distinguish them, we utilize attributes of license plates, including the aspect ratios of individual characters, the regular intervals between characters, and the number of characters constituting license numbers. We refer to these attributes collectively as the structural constraints of license plates.We also introduce the operators of delete, merge, split and recover into the character segmentation procedure. Note that characters may be missed during license plate location and binarization. The recover operator is introduced to retrieve missing characters.

During processing the segmentation procedure applies the first three operators (delete, merge, and split) to the set of surviving components in an attempt to determine if a component satisfies the structural constraints of license plates. If such a component can be determined, the character recognition procedure is invoked to identify a character from the component. The above process repeats until no character can be extracted from the set of surviving components. Thereafter, if the number of extracted characters is less than the number of characters in license numbers, the recover operator starts at the outermost characters of those detected and searches for characters along the alignment of the known characters. The search continues until no character can be retrieved within an extent determined by the average width of characters as well as intervals between characters.

Next, the collection of identified characters is verified in the confirmation stage, where the compositional semantics of license numbers plays an

important role. The set of characters will be deemed to form a valid license number if it agrees with the compositional semantics.

B. Optical Character Recognition

In this subsection we discuss the character recognition procedure. Since, as already mentioned, license plates may be bent and/or tilted with respect to the camera, characters extracted from such license plates may be deformed. Furthermore, input characters may be noisy, broken or incomplete. Character recognition techniques should be able to tolerate these defects. In this Fig.3. Nodal types: (a) end-point, (b) three-way node, and (c) four-way node. study, we develop our own character recognition approach to suit our particular application. The proposed approach consists of three steps: character categorization, topological sorting, and self-organizing (SO) recognition. In the first step, the input character is distinguished as numerical or alphabetical. This is easily accomplished by referring to the compositional semantics of license numbers. In the next step, the topological features of the input character are computed and are compared with those of prestored character templates. Compatible templates will form a test set, in which the character template that best matches the input character is determined. The template test is performed by a SO character recognition procedure.

1) Topological Sorting: The topological features of characters utilized in this study include the number of holes, endpoints, three-way nodes, and four-way nodes (see Fig. 4 for their definitions). These features are invariant to spatial transformations (including rotation, translation and scale change). Moreover, these features, which are qualitative in nature, can be easily and reliably detected compared to quantitative features. However, input characters are usually imperfect; extra or missing features may occur. The following rule is employed for topological sorting. A character template is compatible with a given character whenever 1) their difference in the numbers of holes is within the range and 2) their difference between the numbers of nodes of any type is within the range, a smaller range is given to the hole feature because it is enerally more reliable in detection than nodes. In our experiments no more than three out of ten numerical character templates and six out of 26 alphabetical character templates have passed topological sorting for any given character. This has greatly reduced the number of templates in the test set and hence the time complexity for character recognition.

2) Template Test: The templates in the test set are matched against the input character and the best match is determined. The template test is primarily accomplished using a SO character recognition approach, which is based on Kohonen's SO neural network [19]. The idea behind the proposed technique is as follows. Given an unknown character and a character template, the input character is encoded in terms of synaptic weights in the between-layer links of the neural network. The character template here serves as a stimulus, which repeatedly innervates the neural network, causing the synaptic weights of the neural network to gradually change. This process continues until the weights stabilize. We sum up the changes of synaptic weights during processing. The total change in weight in a sense reflects the level of dissimilarity between the unknown character and the character template.

Let $C = \{c_1, \ldots, c_L\}$ be the set of character templates surviving from the topological sorting of an unknown input character. Let $d_1, \ldots, d_L$ denote the computed dissimilarities between the unknown character and the character templates.

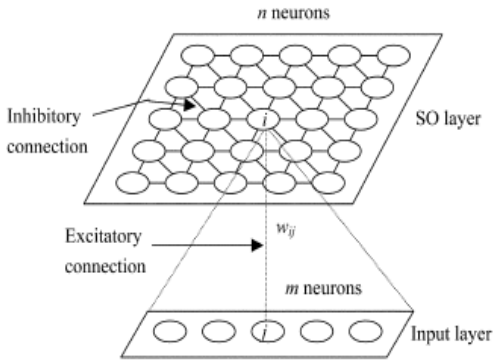

Fig. 3. Kohonen SO neural model.

the unknown character is taken to be the class to which the unknown character belongs.

a) SO Neural Model: In this subsection, we brief the key components of the Kohonen SO neural model, which will be used in the later practical implementation. Referring to Fig3, the underlying configuration of the SO neural network consists of two layers, an input layer and an SO layer.

Let w_{ij} be the weight of the link between SO neuron n_i and input neuron . The weight vector for n_i is $W_i = (w_{i1}, w_{i2}, \ldots, w_{im})$, where m is the number of input

neurons. Let $\mathbf{v}_i = (v_1, v_2, \ldots, v_m)$ denote an external stimulus. The input to due to the stimulus is

$$I_i^s = \mathbf{W}_i \cdot \mathbf{V} = \sum_{k=1}^{m} w_{ik} v_k. \quad (1)$$

The lateral interaction among SO neurons is characterized by a "Mexican-hat" function [21], denoted $h(\mathbf{r})$, where represents a position vector. Let u_{ik} be the weight of the connection between SO neurons n_i and n_k located at $\mathbf{r}_i$ and $\mathbf{r}_k$, respectively. The input to n_i due to lateral interaction is

$$I_i^l = \sum_{n_k \epsilon N, k \neq i} a_k \mathbf{u}_{ik} h(\mathbf{r}_k - \mathbf{r}_i) \quad (2)$$

where N is the set of SO neurons and $a_k = \psi(net_k)$ is the activation of n_k, in which net_k to be defined is the net input to n_k and ψ is the output function defined by a sigmoid function.

A leakage term $e(a)$, which dissipates activations of SO neurons once a stimulus has been removed, is introduced for every SO neuron. The net input to n_i then sums the inputs from the stimulus, lateral interaction, and leakage

$$net_i = I_i^s + I_i^l + e(a_i). \quad (3)$$

During competition among SO neurons, the winner n_c is determined by $net_c = \max_{1<i<n}\{net_i\}$. Next, the winner together with its neighbors, say set N_c, engage in a group learning process. During this process a neuron close to the winner will gain a high rate of learning while a neuron located far from the winner will have a low rate of learning. The rate of learning is governed by the Gaussian function g. The learning rule for the neurons in is then defined as

$$\Delta \mathbf{w}_k = (\mathbf{v} - \mathbf{w}_k) g(\mathbf{r}_k - \mathbf{r}_c), \quad k \epsilon N_c. \quad (4)$$

Finally, the updating equation for SO neuron is

$$\mathbf{w}_k^{t+1} = \mathbf{w}_k^t + \rho(t) \Delta \mathbf{w}_k^t, \quad k \epsilon N_c \quad (5)$$

Where $\rho(t)$ is the step size, which decreases monotonically with increasing t.

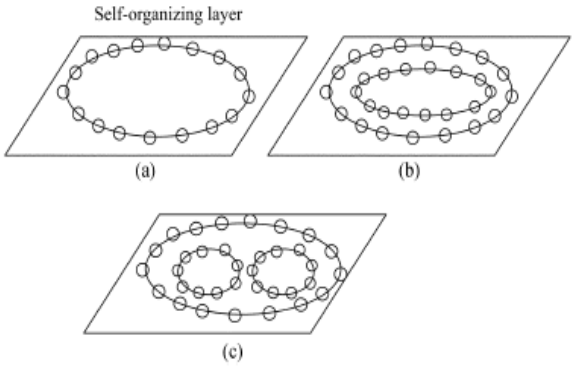

Fig4. SO layers: (a) 0-hole, (b) 1-hole, and (c) 2-hole SO layers.

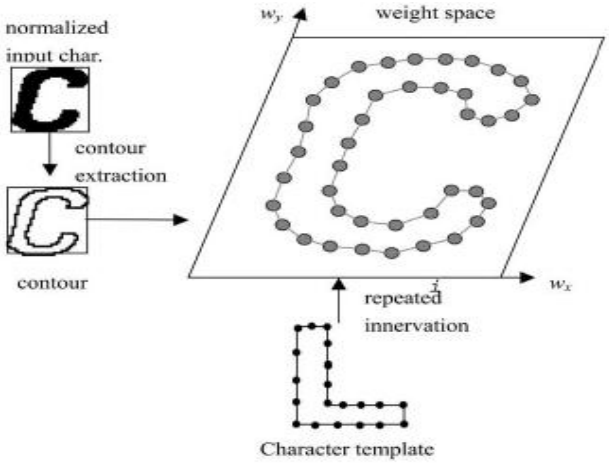

Fig5. Example of SO character recognition.

b) Practical Implementation: To begin, we group characters into three categories, referred to as 0-hole, 1-hole, and 2-hole, according to the number of holes contained in the characters. Each category has its own associated SO neural network, which contains 40 SO neurons and two input neurons. The difference among the three neural networks is primarily in their configurations of SO layer (see Fig. 4).

Referring to the example shown in Fig.5, suppose that we are given an unknown character ("C" in this example). The character is normalized in size (16 by 16 pixels) in order to be consistent with the character templates. The number of holes in the character is computed. Here, we always choose the neural network according to the computed number regardless of whether the computed number is the true number of holes for that character. Since the input character ("C")

has no hole, the neural network with the 0-hole SO layer is chosen. Next, the contour and its length of the unknown character are found. The length is divided into 40 approximately equally spaced intervals. Starting at any arbitrary point along the contour, the two dimensional (2-D) position vectors, i.e., the (row, column) coordinates, of the 40 interval boundaries are extracted. See the right side of Fig.5. We choose 40 points because they are about half the average number of contour points in the character templates. The position vectors of the 40 contour points are then assigned, one by one in the order of extraction, to the weight vectors of the 40 SO neurons of the chosen neural network. Note that since all the configurations of SO layer of the three neural networks are symmetrical which SO neuron should be assigned first is not important.

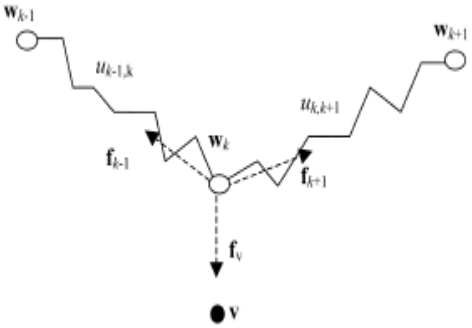

Fig6. Spring model.

Consider a 2-D space with axes corresponding to the two components of weight vectors of SO neurons. The weight vector of each SO neuron can be represented as a point in the space. This space is referred to as the weight space of the neural network. Since the weight vectors of SO neurons are set to the position vectors of contour points of the input character, the contour will be recreated in the weight space when we represent the weight vectors of SO neurons as points in the weight space. See the picture on the right hand side of Fig.5

Suppose that a template ("L" in the example) chosen from the test set for the input character is to be matched against the character. Rather than the entire template, just its contour serves as the stimulating pattern, which repeatedly innervates the neural network until its synaptic weights stabilize. In our implementation the contour points are fed into the input layer of neural network one at a time. Consider an input contour stimulus point v.

The SO neurons of the network compete for the stimulus. The winner n_c is determined by $n_c = \arg(\max_{1\le i\le 40}\{\mathbf{w}_i \cdot \mathbf{v}\})$, where $\mathbf{w}_i$'s are the weight vectors of the 40 SO neurons. The winner and its first- and second-order neighbors, call them set N_c, join in the following learning process

$$\Delta \mathbf{w}_k = \mathbf{d}_k g(\mathbf{r}_k - \mathbf{r}_c), \qquad k \in N_c, \tag{6}$$

Note that $(\mathbf{v} - \mathbf{w}_k)$ in (12) has been replaced by in (14). The $\mathbf{d}_k$ is computed as follows. We use a model, called the spring model, taken from [4], in which SO neurons are assumed being connected with springs. The elastic spring coefficients are simulated with synaptic weights between neurons. Referring to Fig.5 we denote the SO neurons with their weight vectors. Weight vectors $\mathbf{w}_{k-1}, \mathbf{w}_k$, and $\mathbf{w}_{k+1}$ represent SO neurons n_{k-1}, n_k and n_{k+1}, respectively, where n_k is any learner in and and are the two first-order neighbors of N_c. The learner is connected to its two neighbors with the springs, whose coefficients are and . The point stimulus is denoted in the figure.

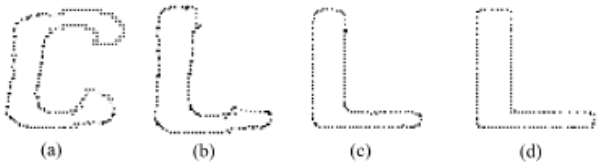

Fig.7. Some intermediate results of shape transformation. (a) 1st iteration. (b) 5th iteration. (c) 10th iteration. (d) 42nd iteration.

The here serves as an attractive source, which during the learning process attempts to pull $\mathbf{w}_k$ toward it with force $\mathbf{f}_v$. However, the springs exerting forces and on try to pull toward its neighbors $\mathbf{w}_{k-1}$ and $\mathbf{w}_{k+1}$. over the interval when correct templates were chosen for testing.

Fig8. Distinguishing parts of ambiguous characters.

c) Remarks: The proposed character recognition approach has difficulty distinguishing character pairs (8, B) and (O, D) especially when they are distorted. To overcome this, we predefine an ambiguity set containing the characters 0, 8, B and D. For each character in the set, the nonambiguous parts of the character are specified. During character recognition, once an

unknown character is classed as one of the characters in the ambiguity set, an additional minor comparison between the unknown character and the classed character is performed. The comparison focuses on only the nonambiguous parts of the character.

Our character recognition method gives different measurements of dissimilarity for the same character with different tilt angles with respect to the camera. Currently, this issue has not troubled us because the characters extracted from the images of license plates are all in a nearly upright position. But, we may improve our algorithm to deal with this by introducing a normalization step to transform license plates into a prescribed orientation prior to license number identification.

IV. EXPERIMENTAL RESULTS

Fig9: Input Image

Fig10: Median filter output image

Fig11: Edge Detection Image.

Fig12: Recognized Characters from Number Plate.

V. CONCLUDING

Compared to most previous work that in some way restricted their working conditions, the techniques presented in this paper are much less restrictive. The proposed LPR algorithm consists of two modules, one for locating license plates and one for identifying license numbers. Soft computing techniques rooted(for license plate location) and neural (for license number identification) disciplines were introduced to compensate for uncertainties caused by noise, measurement error and imperfect processing. Although

the proposed algorithm is concerned with the license plates of one specific country, many parts in the algorithm are readily extended to use with license plates of other countries. Specifically, since color and edge are two fundamental features of license plates, the color edge detector introduced in the locating module is readily adapted to other color schemes by replacing the color parameters embedded in the detector.

VI. REFERENCES

[1] G. Adorni, F. Bergenti, and S. Cagnoni, "Vehicle license plate recognition by means of cellular automata," in Proc. IEEE Int. Conf. Intelligent Vehicles, 2008, pp. 689–693.

[2] M. H. T. Brugge, J. H. Stevens, J. A. G. Nijhuis, and L. Spaanenburg, "License plate recognition using DTCNNs," in Proc. 5th IEEE Int. Workshop on Cellular Neural Networks and Their Applications, 2008, pp. 212–217.

[3] K. R. Castleman, Digital Image Processing. Englewood Cliffs, NJ: Prentice-Hall, 2006, pp. 550–554.

[4] S. W. Chen, G. C. Stockman, and K. E. Chang, "SO dynamic deformation for building of 3-D models," IEEE Trans. Neural Networks, vol. 7, pp. 374–387, June 2006.

[5] J. R. Cowell, "Syntactic pattern recognizer for vehicle dentification numbers," Image and Vision Comput., vol. 13, no. 1, pp. 13–19, 2005.

[6] Y. Cui and Q. Huang, "Character extraction of license plates from video," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1997, pp. 502–507.

[7] P. Davies, N. Emmott, and N. Ayland, "License plate recognition technology for toll violation enforcement," Inst. Elect. Eng. Colloquium Image Analysis for Transport Applications, pp. 7/1–7/5, 1990.

[8] S. Draghici, "A neural network based artificial vision system for license plate recognition," Int. J. Neural Systems, vol. 8, pp. 113–126, 1997.

[9] D. M. Emiris and D. E. Koulouriotis, "Automated optic recognition of alphanumeric content in car license plates in a semi-structured environment," in Proc. Int. Conf. Image Processing, vol. 3, 2001, pp. 50–53.

[10] D. S. Gao and J. Zhou, "Car license plates detection from complex scene," in Proc. 5th Int. Conf. Signal Processing, vol. 2, 2000, pp. 1409–1414.

[11] H. A. Hegt, R. J. De la Haye, and N. A. Khan, "A high performance license plate recognition system," in Proc. IEEE Int. Conf. System, Man,

and Cybernetics, vol. 5, 1998, pp. 4357–4362.

[12] X. F. Hermida, F. M. Rodriguez, J. L. F. Lijo, F. P. Sande, and M. P. Iglesias, "A system for the automatic and real time recognition of VLP's (Vehicle License Plate)," in Proc. Lecture Notes in Computer Science, 1997, vol. 1311, pp. 552–558.

[13] H. Hontani and T. Koga, "Character extraction method without prior knowledge on size and position information," in Proc. IEEE Int. Conf. Vehicle Electronics, 2001, pp. 67–72.

[14] J. M. Keller and R. Krishnapuram, "Fuzzy decision models in computer vision," in Fuzzy Sets, Neural Networks, and Soft Computing, R. R. Yager and L. A. Zadeh, Eds. New York: Van Nostrand, 1994, pp. 213–232.

[15] D. S. Kim and S. I. Chien, "Automatic car license plate extraction using modified generalized symmetry transform and image warping," in Proc. IEEE Int. Symp. Industrial Electronics, vol. 3, 2001, pp. 2022–2027.

[16] K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, "Learning-based approach for license plate recognition," in Proc. IEEE Signal Processing Society Workshop, vol. 2, 2000, pp. 614–623.

[17] S. K. Kim, D. W. Kim, and H. J. Kim, "A recognition of vehicle license plate using a genetic algorithm based segmentation," in Proc. Int. Conf. Image Processing, vol. 2, 1996, pp. 661–664.

[18] G. J. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic, Theory and Applications. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[19] T. Kohonen, Self-Organization and Aassociative Memory. New York: Springer-Verlag, 1989.

[20] R. A. Lotufo, A. D. Morgan, and A. S. Johnson, "Automatic numberplate recognition," Inst. Elect. Eng. Colloquium on Image Analysis for Transport Applications, pp. 6/1–6/6, 1990.

[21] D. Marr, Vision. New York: Freeman, 1982.

[22] K. Miyamoto, K. Nagano, M. Tamagawa, I. Fujita, and M. Yamamoto, "Vehicle license plate recognition by image analysis," in Proc. Int. Conf. Industrial

Electronics, Control and Instrumentation, 1991, pp. 1734–1738.

[23] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," IEEE Trans. Veh. Technol., vol. 49, pp. 2309–2319, Nov. 2000.

[24] Y. Nakagawa and A. Rosenfeld, "Some experiments on variable thresholding," Pattern Recognition, vol. 11, no. 3, pp. 191–204, 1979.

[25] J. A. G. Nijhuis, M. H. T. Brugge, K. A. Helmholt, J. P. W. Pluim, L. Spaanenburg, R. S. Venema, and M. A. Westenberg, "Car license plate recognition with neural networks and fuzzy logic," in Proc. IEEE Int. Conf. Neural Networks, vol. 5, 1995, pp. 2232–2236.

[26] R. Parisi, E. D. D. Claudio, G. Lucarelli, and G. Orlandi, "Car plate recognition by neural networks and image processing," in Proc. IEEE Int. Symp. Circuits and Systems, vol. 3, 1998, pp. 195–198.

[27] S. W. Perry, H. S. Wong, and L. Guan, Adaptive Image Processing, A Computational Intelligence Perspective. Boca Raton, FL: CRC, 2002.

[28] J. C. H. Poon, M. Ghadiali, G. M. T. Mao, and L. M. Sheung, "A robust vision system for vehicle license plate recognition using grey-scale morphology," in Proc. IEEE Int. Symp. Industrial Electronics, vol. 1, 1995, pp. 394–399.

[29] H. Ritter, T. Martinetz, and K. Schulten, Neural Computation and Selforganizing Maps an Introduction. New York: Addison-Wesley, 1992.

[30] L. Salgado, J. M. Menendez, E. Rendon, and N. Garcia, "Automatic car plate detection and recognition through intelligent vision engineering," in Proc. IEEE Int. Carnahan Conf. Security Technology, 1999, pp. 71–76.

[31] T. Sirithinaphong and K. Chamnongthai, "The recognition of car license plate for automatic parking system," in Proc. 5th Int. Symp. Signal Processing and its Applications, 1998, pp. 455–457.

[32] Y. S. Soh, B. T. Chun, and H. S. Yoon, "Design of real time vehicle identification system," in Proc. IEEE Int. Conf. System, Man, and Cybernetics: Humans, Information, and Technology, vol. 3, 1994, pp. 2147–2152.

[33] K. Yamaguchi, Y. Nagaya, K. Ueda, H. Nemoto, and M. Nakagawa, "A method for iidentifying specific vehicles using template matching," in Proc. IEEE Int. Conf. Intelligent Transportation Systems, 1999, pp.

[34] M. Yu and Y. D. Kim, "An approach to Korean license plate recognition based on vertical edge matching," in Proc. IEEE Int. Conf. Systems, Man, and Cybernetics, 2000, pp. 2975–2980.

[35] N. H. C. Yung, K. H. Au, and A. H. S. Lai, "Recognition of vehicle registration mark on moving vehicles in an outdoor environment," in Proc. IEEE Int. Conf. Intelligent Transportation Systems, 1999, pp.