IJE TRANSACTIONS B: Applications Vol. 26, No. 11, (November 2013) 1347-1356

International Journal of Engineering

Journal Homepage: www.ije.ir

Real Time Implementation of a License Plate Location Recognition System Based on Adaptive Morphology

M. Ashourian*[a], N. Daneshmandpour[a], O. Sharifi Tehrani[a] , P. Moallem[b]

[a] *Department of Electrical Engineering, Majlesi Branch, Islamic Azad University, Majlesi, Isfahan, Iran*
[b] *Department of Electrical Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran*

PAPER INFO

Paper history:
Received 20 October 2012
Received in revised form 31 January 2013
Accepted 16 May 2013

Keywords:
License Plate Recognition
Car Plate Location Recognition
Real Time Implementation
Adapted Morphology

ABSTRACT

License plate recognition (LPR) using morphology has the advantage of higher resistance to changes of brightness, high speed processing, and low complexity. However, these approaches are sensitive to the distance of the plate from the camera and imaging angle. Various assumptions reported in other works might be unrealistic, and cause major problems in practical experiences. In this paper we considered morphological approaches and improved them using adaptive techniques in order to provide more compatibility with practical applications. We examined the developed system on several car plate image databases with different conditions such as different camera distance, and different car views. The average achieved rate of success was 89.95% for all car plate location recognition, which is more than 6.0% improvement in comparison to previous morphological methods. We further developed and implemented an FPGA realization of the pre-processing stage of the system which is the main computation load of our LPR system.

doi: *10.5829/idosi.ije.2013.26.11b.10*

1. INTRODUCTION[1]

Car license plate recognition has been widely used in intelligent transportation management systems. For some applications such as highway traffic monitoring and tracking specific cars in the highway, real time video processing is necessary, and for applications such as parking lots, automatic toll collection, still image processing is satisfactory and applying it provides more accurate results [1]. Car plate recognition consists of two steps: license plate location detection (LPLD), and optical character recognition (OCR). The first step, LPLD, is more challenging as it depends on various parameters such as the plate angle, light, camera distance from the plate, and the background.

Some previous LPLD methods are based on image analysis in HSI color model [2, 3] which are very sensitive to light conditions, and brightness changes. In [4], the proposed LPLD is based on processing a histogram with different numeric parameters to find the plate location. This method is simple, and performs at a high speed. However, use of numerical parameters affects the accuracy and success rate. There are few methods using Hough transform to detect side lines of the plate [5]. Hough transform detects the lines and curves very efficiently, but it performs relatively slow for plate detection.

Some LPLD methods such as [6, 7] use special characteristics of the license plate to find the plate location. These methods might show high performance for the plates in a specific country, but should be calibrated and modified to be used in other countries with different alphabets [1]. Use of intelligent classification techniques such as SVM and RBF is suggested in [8] with competitive results, but their implementation requires very high speed processors. The method proposed in [9] is carried out on an ordinary processing system, but needs high quality cameras for image capturing.

Morphological techniques have been successfully used in different image recognition systems [10, 11]. In LPR systems, a combination of edge statistics and mathematical morphology methods [12-15] achieve

*Corresponding Author Email: mohsena@ieee.org (Mohsen Ashourian)

very good results. In these methods, the gradient magnitude and the local variance of an image are computed based on the principle that the brightness changes in the LP region are more probable and more frequent than elsewhere. One disadvantage of this approach is that edge based methods alone can hardly be applied to complex images, since they are too sensitive to unwanted edges, and may also show a high edge magnitude or variance (e.g., the radiator region in the front view of the vehicle). Therefore, when combined with morphological steps that eliminate unwanted edges in the processed images, the LPR extraction rate is relatively high and performs fast in comparison to other methods [12, 13].

Most of morphological methods begin with edge detection, followed by morphological operators. Kasaei et al. [14] use edge detection followed by morphology operators, so the plate is extracted as a white rectangular with certain dimensions' ratio. The work proposed in [12] is similar to [15], but it removes some lines which are below a fixed threshold in the histogram of the edge detected image. In a similar approach, Faradji et al. [16] first remove some of the rows below a fixed threshold on the histogram and then estimate the plate region by numerical coordinate selection. Later using morphology operators, the exact location of the plate is found.

There are some other adaptive morphological methods. The authors in [17] develop a new adaptive method for binarization. The proposed method in [18] presents an adaptive approach to character segmentation. In this paper, we introduce adaptive thresholding which has not been discussed before.

Moreover, most of the previously reported LPR methods have not considered hardware implementing challenges which has recently attracted more attention [19-22]. In [19], a system based on multiple processors is used for LPLD implementation. The authors in [20] develop a prototype tool to automate the construction of hardware accelerators, and use it in developing a SoC (System on Chip) for an LPLD system. In [21], an FPGA implementation of LPR is proposed. The input image is segmented into 16 areas, each processed in parallel by a multiple calculation unit. This approach improves the speed and is suitable for video processing with high frame rates. However the system has high complexity due to the large number of filters and morphological operators it uses. In [22], the authors concentrate more on the FPGA implementation of the OCR part of license plate recognition assuming that the LPLD part already works with high accuracy. But, this assumption might not always hold for images with complex background [1].

In this article, a real-time method that is accurate and simple is designed. The proposed method is much less sensitive to light conditions, and car distance from camera, and speed of the cars. We have introduced an FPGA implementation for the LPR pre-processing stage. The pre-processing stage includes I/O, RS-232 image buffering, Sobel filtering, and Gaussian filtering. The parallel processing nature of the FPGA implementation and the idea of using Hardware/Software Co-Design provide the potential of high speed and real time implementation. The remaining LPR stages which are implemented in MATLAB have low data processing complexity, and therefore is not a bottleneck for real-time implementation.

The article is structured as follows. First, the proposed LPR algorithm is described in section 2. Section 3 presents the hardware realization and implementation. Section 4 presents the experimental results of this approach. Finally, section 5 concludes the paper.

2. THE PROPOSED METHOD

Figure 1 shows the flowchart of the developed algorithm consisting of 8 steps which are explained in the following sections.

First, in the pre-processing stage the noise is reduced and smoothing is done to avoid unusable details of the image. In order to extract the vertical histogram of the image, vertical edge detection is applied followed by the binarization step. These steps are similar to those described in [17]. In this paper, the accuracy of previous morphological algorithms has been improved using several innovations. In former approaches, the threshold of the histogram was fixed, whereas in this paper we first normalize the histogram, and then we set a threshold level based on the histogram features. Using the adaptive threshold technique, the chance of finding the exact plate location increases. In this case, we directly select such a region as the plate location. Otherwise, if the algorithm finds more than one region, we further use a new concept called probabilistic erosion after the vertical coordinate selection. Also, labeling connected component previously introduced in [7, 17] has been used to perform plate extraction.

2. 1. Pre-processing

Figure 2 is an image of a car as a sample of the database. The input RGB image is first converted to grayscale, and then a Gaussian filter is used to reduce noise and smooth out unusable image details.

$$h_g(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$
$$h(x, y) = \frac{h_g(x, y)}{\sum_x \sum_y h_g} \quad (1)$$

Equation (1) shows the Gaussian filter relation. In this paper we use a 3×3 Gaussian filter with the variance of 0.8 as shown in Figure 3.

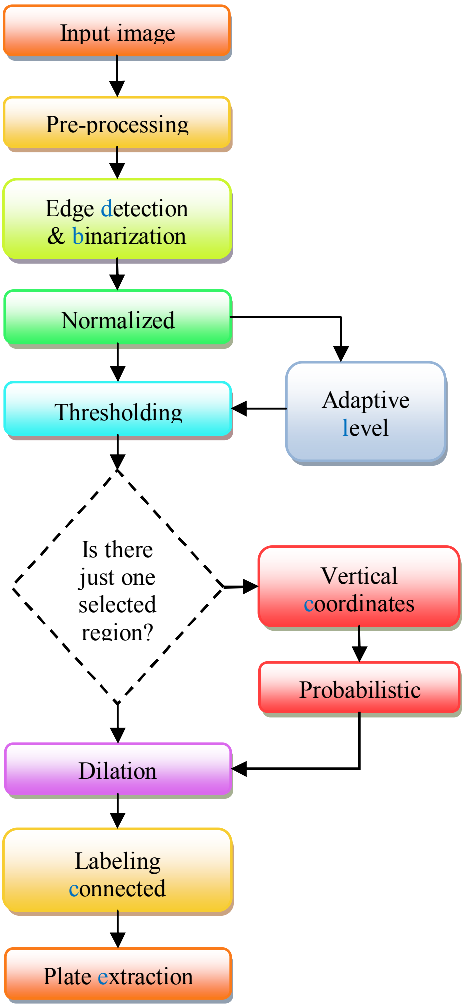


Figure 1. Block diagram for the adaptive morphology method

Figure 2. An image of a car as a sample of the database

Figure 3. The result of pre-processing (grayscale conversion and noise removing)

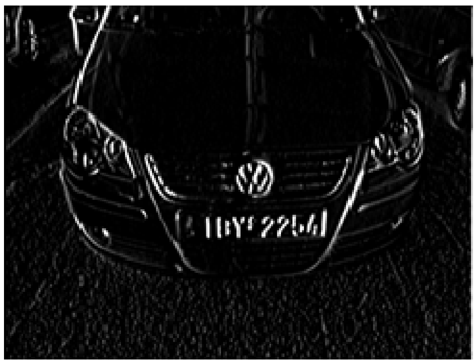


Figure 4. The result of vertical Sobel on Figure 3

Figure 5. Binary color space conversion of Figure 4

The Sobel and Prewitt edge detection is often used in LPR [12, 15-18]. Since the Sobel filter detects more details, it works better for histogram analysis. The plate region has a lot of vertical edges, and a vertical Sobel mask is used for extracting the plate region. The value of pixels in the grayscale images is usually in an interval of [0, 255], which is often normalized to [0, 1]. In transforming the color space from grayscale to binary, normally pixels upper than 0.5 are set to one, and others are set to zero [23]. There are some better methods to select adaptive levels for binarization [24, 25]. Figure 5 shows the binary color space conversion with Otsu's adaptive level [25].

2. 2. Histogram of White Pixels Here, we define the histogram as a graph which shows the number of white pixels in each row of an image. Figure 6 shows the defined histogram for the picture in Figure 5. In this graph X-axis is the number of rows of the image and Y-axis is the sum of white pixels in each row.

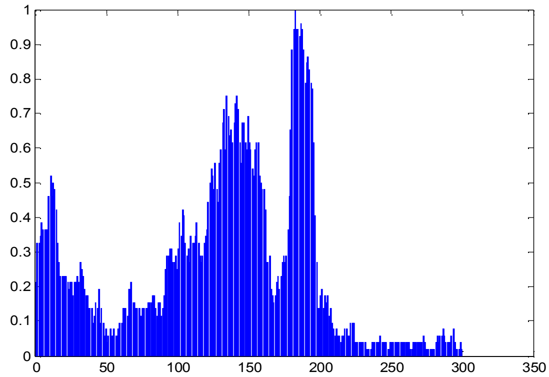


Figure 6. Vertical histogram

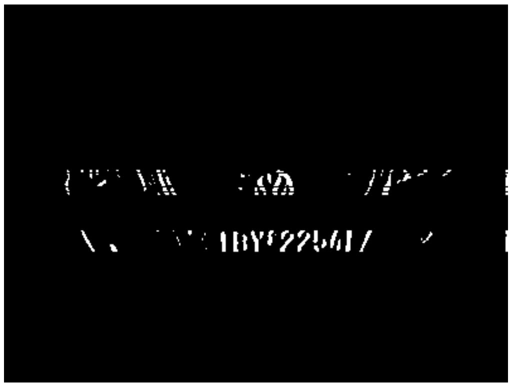
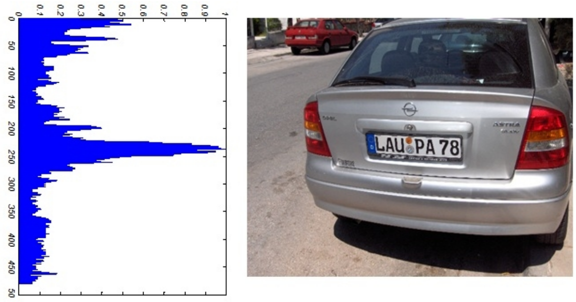
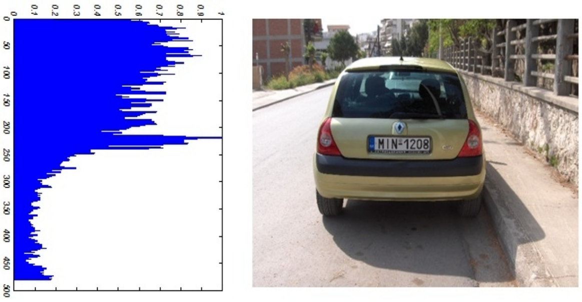


Figure 9. Vertical coordinates selection (Section 2.5)



(a)



(b)

Figure 7. A comparison of two images, one with a low mean of histogram (a) and other with a higher one (b)

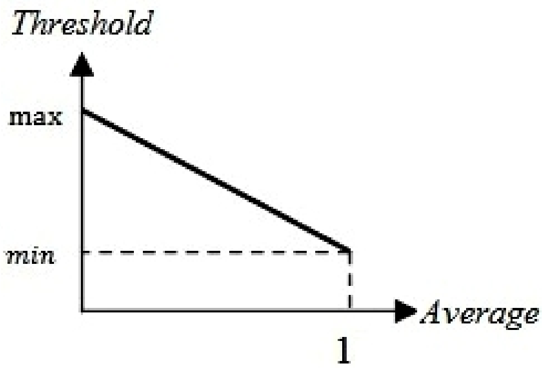


Figure 8. Linear estimation for reverse relation (Section 2.4)

2. 3. Adaptive Thresholding As it is clear from Figure 6, the plate region appears as a peak in the histogram. To separate this area, a threshold level should be used. Thresholding means to keep the rows which are above this threshold level in the histogram, and removing the rest. Using a fixed threshold for all images is not favorable and it reduces the reliability of some previous approaches. So, we use an adaptive threshold based on the features of each image.

Before thresholding, the histogram should be normalized which means the distance of the minimum to the maximum is mapped to the interval of zero and one. For this purpose we divide all values of the histogram by their maximum value.

Our investigation on various image databases and their corresponding histograms show that there is a reverse relationship between the histogram threshold and the total average value of the image pixels' brightness. When the car image is taken directly and there are little excess background details on it, there is just one peak in the histogram, and other parts of the histogram have low levels. Thus, the average value is low and we can choose a threshold close to peak value.

On the other hand, when the car image is taken with an angle or when the car images have higher background details, the histogram has several extreme points. Figure 7 gives a comparison of two images with different types of histograms. In Figure 7(b) the average value is larger so it is better to choose a lower threshold. Here the case is that we do not know which peak belongs to the plate region, and applying an upper threshold may incorrectly remove the plate itself while some other parts of the image are kept. We find a solution in the next steps to remove these extra parts. For simplicity, we assume that the relation is following the line with a negative slope as depicted in Figure 8.

2. 4. Vertical Coordinates Selection The output of the first round of processing might result in some areas that are identified incorrectly as the plate area. In this step we try to select the exact vertical coordinate of the plate. Two numeric parameters, p and q are selected based on the plate size. If the widths of these parts are less than p or more than q, they will be removed. Figure 9 shows the result of removing the excess rows.

2. 5. Probabilistic Erosion Here the morphological operator, erosion, is applied which is defined by Equation (2):

$$A\Theta B=\{z\,|\,(B_z)\subseteq A\} \tag{2}$$

The result of erosion A with structural element B is all the components of B that belong to A [26]. Figure 10 shows two typical structural elements. In addition, the probability of finding the plate in lower positions of the image is more than on higher positions. In an LPR system, sometimes the program selects a region at the top of the image as the plate location, while plates are normally placed on lower positions of the car and consequently, it can be found on lower positions of the image. Therefore, we define a new type of erosion with a linear structural element. But its length decreases from the first row to the last. This means less erosion performs in the lower positions of the image. Figure 11 shows the result of applying erosion to the image.

2. 6. Dilation Dilation is another morphological operator defined by Equation (3). In this paper we use rectangular dilates meaning dilation with a rectangular structural element [26]:

$$A\oplus B=\{z\,|\,(\hat{B}_z)\cap A\neq\phi\} \tag{3}$$

2. 7. Plate Region Detection and Extraction As shown in Figure 12, after applying the morphological operator (dilation), the output image contains some white boxes detected as the plate region, and some other white places in a dark background. In this step we should select one of these areas as the plate region. We propose an efficient technique based on the labeling connected component approach [20]. At first we find the coordinates of the white boxes in Figure 12, and then we refer back to Figure 11 and we label the connected components within the boxes' coordinates. The plate region is extracted by searching for the box coordinates that have the largest number of labeled components. Finally, vertical dilate is performed to extract the whole plate more accurately. Figure 13 shows the result of the final step with vertical dilates.

3. HARDWARE REALIZATION AND IMPLEMENTATION

Our hardware developed system is based on Field Programmable Gate Array (FPGA). The FPGA integrate circuits consisting of two dimensional logic gates.

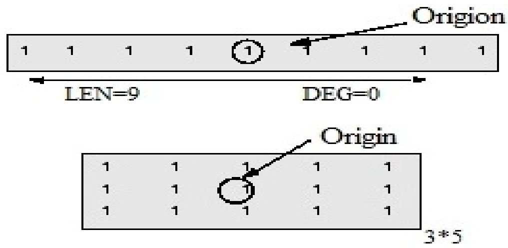


Figure 10. Two examples of structural elements: A horizontal line with length of 9 pixels and a rectangular with dimension of 3×5 pixels

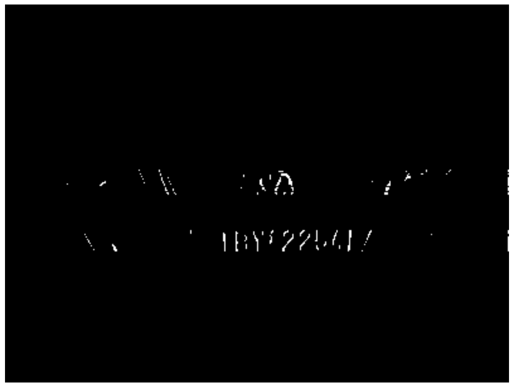


Figure 11. Probabilistic erosion

Figure 12. The result of performing dilation

Figure 13. Plate region detection and extraction

Figure 14. The extracted plate

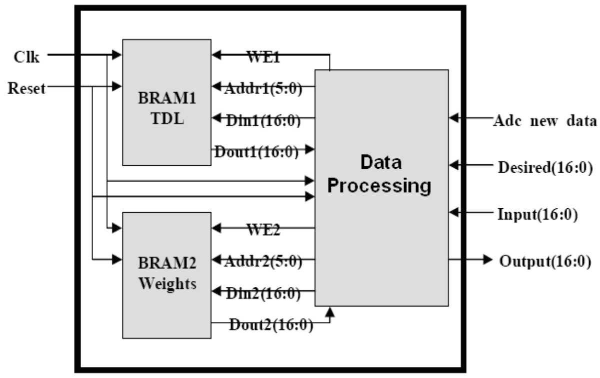


Figure 15. Entity diagram of the pre-processing stage

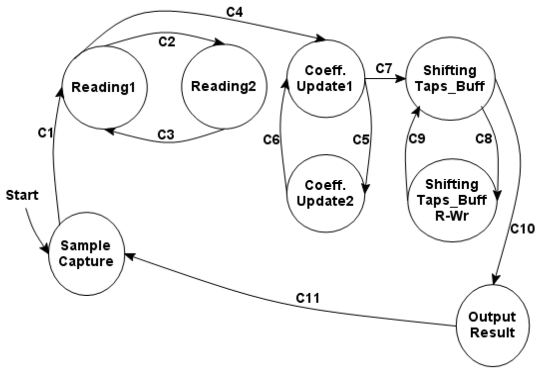


Figure 16. Data Flow Diagram (DFD) in the pre-processing stage

TABLE 1. State change conditions in filter DFD

State	condition description
C1	If sample capture is done
C2	If BRAMs current address fetch, decode and load, completed
C3	Current input and weight read, completed
C4	All Tap Delay Line (TDL) reading, completed
C5	BRAMs current address fetch, decode and load, completed
C6	Current weight updating and writing it to BRAM, completed
C7	All weights updating, completed
C8	BRAMs current address fetch, decode and load, completed
C9	Current input insertion-shifting, completed
C10	One-step shifting of all TDL, completed
C11	Output calculation and loading, completed

TABLE 2. Resource utilization on XC3S500E-4fg320 FPGA chip

Logic utilization	Used	Available	Utilization
Number of slice flip flops	1054	9,312	12%
Number of 4 input LUTs	2,945	9,312	32%
Number of occupied slices	1,994	4,656	43%
Number of bonded IOBs	30	232	13%
Number of BUFGMUXs	4	24	16%

Recently they are equipped with major elements such as Digital Clock Management, Phased Locked loop, Delay Locked Loop, Block RAM, Distributed RAM, and etc. We can use software and hardware cores like DCT, FFT, filters and etc. The most important cores are MicroBlaze in soft processors, and PPC or ARM in hard processors.

In this article we used the MicroBlaze processor as a software-processor core which operates on the Xilkernel real-time operating system. Also we used C and VHDL programming language for implementing desired functions and algorithms. This method is optimum both for cost and hardware resource utilization.

For implementing required pre-processing functions (Gaussian and Sobel filters) we use the Pure Hardware Model proposed in [27, 28]. This model is optimum in the hardware perspective and utilizes minimum internal resources. The Entity Diagram and the Data Flow Diagram (DFD) are depicted in Figure 15 and 16, respectively. Table 1 shows the state change conditions in the pre-processing data flow.

Buffering is performed by the internal Block-RAMs of the FPGA, so the image information are received via the computer serial port (RS-232) with Baud Rate of 115200 bps. Therefore, data are buffered and provided for later processing [29].

After this step, the processed data are sent via the serial port to the computer which runs the MATLAB software with previous Block-RAMs. For implementing the RS-232 send and receive blocks, we used an optimum hardware core of this protocol designed based on state machine in [29]. MicroBlaze 32-bit processor with Xilkernel real-time operating system provides the capability of multi-tasking and multi-thread programming [29]. The “hardware processing” and “operating system” parts are software implemented, and the parts of “buffering”, “filtering”, and “I/O” are implemented with pure hardware. These parts are integrated and perform a plan of HW/SW CO-Design. In this article we used the XC3S500E model of Starter-Kit that is produced by Xilinx company. Table 2 shows the results of the hardware implementation (resource utilization). The accuracy of the implemented hardware system is evaluated as 88.5%.

TABLE 3. The rate of success for different types of image

Databases	Number of total images	Number of correctly detected images	Rate of success
Close front view [5]	100	85	85%
Close back view [5]	109	106	97.24%
Wide front view [5]	67	53	79.1%
Wide back view [5]	52	48	92.3%
Front truck view [5]	37	26	70.27%
Arabic character plate	135	121	89.62%
Total	**488**	**439**	**89.95%**

TABLE 4. Comprehension of the performance of three methods

Method	Rate of success
Proposed method	**89.95%**
Basic fixed threshold [16]	83.5%
Method proposed in [15]	76.75%

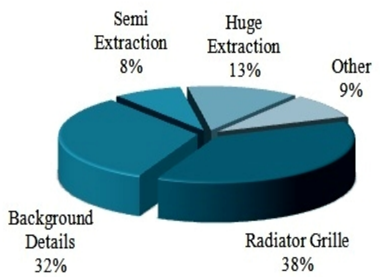


Figure 18. Comprehension of false detection categories

The accuracy of the grid sensitivity is examined for the arrangements of five different non-uniform grid systems with 5829, 7121, 8796, 10466, and 13658 elements, respectively. The results are shown in Table 1. From these comparisons, it is suggested that 10466 non-uniform elements are sufficient to produce accurate results.

4. EXPERIMENTAL RESULT

Here we present the average rate of success in recovering the plate area for the systems proposed in [12, 15], and our developed system. Both systems use morphological operations as their core processing unit. In the processing step explained in section 2.4, we chose 0.7 and 0.3. For the processing part in Section 2.5 we adjusted p equal to 8 and q equal to 30

The proposed method has been carried out on the car image databases introduced in [1] and some other pictures we took ourselves with Arabic characters. Thus, the program has been applied to a total of 488 images including 353 plates with Latin alphabet from [1] and 135 local plate images with Arabic alphabets. The database introduced in [1] consists of five distinct portions. Table 3 shows the success rate for each part, and Table 4 gives a comprehension of the performance of three other methods. False detection is defined as the shortcoming of the proposed algorithm. The total amount of the false detection is about 10.05%. We have classified the weakness of the algorithm on each of the five categories. Figure 18 presents the error categories. we present the average rate of success in recovering the plate area for the systems proposed in [12, 15], and our developed system.

5. CONCLUSION

We propose a novel and robust method based on morphology for license plate location. Various images are included in the database with different complexity and backgrounds, camera angle, and camera distance. The proposed method detects the plate location by 89.95% success which shows 6.45% increase compared to the previous methods [16] with fixed thresholds. Most of the previous approaches are tested in specific conditions [1-22]. But our reported result covers almost all situations proving the accuracy and robustness of the proposed system in practice. We further propose an implementation of the LPR pre-processing stage on FPGA. Future research includes implementation of the whole system on an FPGA platform.

6. REFERENCES

1. Anagnostopoulos, C.-N., Anagnostopoulos, I. E., Psoroulas, I. D., Loumos, V. and Kayafas, E., "License plate recognition from still images and video sequences: A survey", ***Intelligent***

Transportation Systems, IEEE Transactions on, Vol. 9, No. 3, (2008), 377-391.

2. Zhangfan, C. and Rongbao, C., "License plate location method based on modified hsi model of color image", in Electronic Measurement & Instruments, ICEMI'09. 9th International Conference on, IEEE, (2009), 4-197-4-201.
3. Chang, S.-L., Chen, L.-S., Chung, Y.-C. and Chen, S.-W., "Automatic license plate recognition", ***Intelligent Transportation Systems, IEEE Transactions on***, Vol. 5, No. 1, (2004), 42-53.
4. Wang, J., Gao, G. and Yang, H., "Research and implementation of license plate location based on histogram division method", in Electronic Measurement & Instruments, ICEMI'09. 9th International Conference on, IEEE, (2009), 230-233.
5. Kamat, V. and Ganesan, S., "An efficient implementation of the hough transform for detecting vehicle license plates using dsp's", in Real-Time Technology and Applications Symposium, Proceedings, IEEE, (1995), 58-59.
6. Mai, V. D., Miao, D. Q. and Wang, R. Z., "An improved method for vietnam license plate location based on mathematic morphology and measuring properties of image regions", ***Applied Mechanics and Materials***, Vol. 105, (2012), 1995-1999.
7. Li, B., Tian, B., Yao, Q. and Wang, K., "A vehicle license plate recognition system based on analysis of maximally stable extremal regions", in Networking, Sensing and Control (ICNSC), 9th IEEE International Conference on, IEEE., (2012), 399-404.
8. Kurniawan, F. and Khalil, M. S., "Performance comparison between svm-based and rbf-based for detection of saudi license plate", in Information Science and Digital Content Technology (ICIDT), 8th International Conference on, IEEE. Vol. 3, (2012), 537-541.
9. Jeffrey, Z. and Ramalingam, S., "High definition licence plate detection algorithm", in Southeastcon, Proceedings of IEEE, (2012), 1-6.
10. Memariyan, F. and Ekhtiyari, E., "Study on wicking measurement in thin layer textiles by processing digital images", ***International Journal of Engineering-Transactions A: Basics***, Vol. 23, No. 1, (2009), 101-110.
11. Hassanpour, H. and Razzazi, M. R., "Design and implementation of a software system for detecting orthographical or morphological errors in persian words", ***International Journal of Engineering-Transactions B: Applications***, Vol. 20, No. 2, (2007), 127-134.
12. Zheng, D., Zhao, Y. and Wang, J., "An efficient method of license plate location", ***Pattern Recognition Letters***, Vol. 26, No. 15, (2005), 2431-2438.
13. Yang, F. and Ma, Z., "Vehicle license plate location based on histogramming and mathematical morphology", in Automatic Identification Advanced Technologies, Fourth IEEE Workshop (2005), 89-94.
14. Kasaei, S. M., Kasaei, S. M. and Monadjemi, S., "A novel morphological method for detection and recognition of vehicle license plates", ***American Journal of Applied Sciences***, Vol. 6, No. 12, (2009), 2066.
15. Martin, F., Garcia, M. and Alba, J. L., "New methods for automatic reading of vlp's (vehicle license plates)", in Proc. IASTED Int. Conf. SPPRA, (2002), 126-131.
16. Faradji, F., Rezaie, A. H. and Ziaratban, M., "A morphological-based license plate location", in Image Processing, ICIP, IEEE International Conference on, IEEE. Vol. 1, (2007), 57-60.
17. Al-Ghaili, A. M., Mashohor, S., Ramli, A. R. and Ismail, A., "Car license plate detection method for malaysian plates-styles by using a web camera", ***Journal of Science & Technology***, Vol. 18, No.2, (2010), 303–319.
18. Nomura, S., Yamanaka, K., Katai, O., Kawakami, H. and Shiose, T., "A novel adaptive morphological approach for degraded character image segmentation", ***Pattern Recognition***, Vol. 38, No. 11, (2005), 1961-1975.
19. Liu, Z., Fu, H. and Xie, M., "Multiple processors license plate recognition system for intelligent transportation management", in Intelligent Information Technology Application, IITA'08. Second International Symposium on, IEEE. Vol. 1, (2008), 333-336.
20. Bellas, N., Chai, S. M., Dwyer, M. and Linzmeier, D., "Fpga implementation of a license plate recognition soc using automatically generated streaming accelerators", in Parallel and Distributed Processing Symposium, IPDPS 20th International, IEEE., (2006), 8 pp.
21. Kanamori, T., Amano, H., Arai, M. and Ajioka, Y., "A high speed license plate recognition system on an fpga", in Field Programmable Logic and Applications, FPL, International Conference on, IEEE, (2007), 554-557.
22. Zhai, X. and Bensaali, F., "Improved number plate character segmentation algorithm and its efficient fpga implementation", ***Journal of Real-Time Image Processing***, (2012), 1-13.
23. Haralock, R. M. and Shapiro, L. G., "Computer and robot vision", Addison-Wesley Longman Publishing Co., Inc., (1991).
24. Costa, L. d. F. D. and Cesar Jr, R. M., "Shape analysis and classification: Theory and practice", CRC Press, Inc., (2000).
25. Otsu, N., "A threshold selection method from gray-level histograms", ***Automatica***, Vol. 11, No. 285-296, (1975), 23-27.
26. Thompson, C. M. and Shure, L., "Image processing toolbox: User's guide", The MartWorks, (1995).
27. Tehrani, O. S., Ashourian, M. and Moallem, P., "An fpga-based implementation of fixed-point standard-lms algorithm with low resource utilization and fast convergence", ***International Review on Computers and Software***, Vol. 5, No. 4, (2010), 436-444.
28. Tehrani, O. S., Ashourian, M. and Moallem, P., "Fpga implementation of a channel noise canceller for image transmission", in Machine Vision and Image Processing (MVIP), 2010 6th Iranian, IEEE, (2010), 1-6.
29. Lee, S. S., Lee, H. W. and Yoon, S. H., "Batch arrival queue with N-policy and single vacation", ***Computers & Operations Research***, Vol. 22, No. 2, (1995), 173-189. .

Figure.** Different database modes comprehension

Figure.** False detection samples of each categories

Real Time Implementation of a License Plate Location Recognition System Based on Adaptive Morphology

M. Ashourian[a], N. Daneshmandpour[a], O. Sharifi Tehrani[a], P. Moallem[b]

[a] *Department of Electrical Engineering, Majlesi Branch, Islamic Azad University, Majlesi, Isfahan, Iran*
[b] *Department of Electrical Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran*

P A P E R I N F O

Paper history:
Received 20 October 2012
Received in revised form 31 January 2013
Accepted 16 May 2013

Keywords:
License Plate Recognition
Car Plate Location Recognition
Real Time Implementation
Adapted Morphology

چکیده

یک سیستمهای تشخیص پلاک مبتنی بر مورفولوژی دارای امتیازاتی همچون حساسیت کمتر به تغییرات روشنایی ؛ سرعت بالاتر و پیچیدگی پایین می باشند. با این حال این روشها به فاصله از دوربین و زاویه عکس برداری حساسند. فرضیات ساده کننده ای که در بسیاری از مقالات مبتنی بر روش مورفولوژی مطرح شده اند در یک پیاده سازی عملی واقعی نبوده و باعث کاهش کارایی می گردند. در این مقاله روش مورفولوژی انتخاب گردیده است و روشهای وفقی جهت بهبود کارایی آن در عمل اعمال گردیده اند. روش طراحی شده بر روی چند پایگاه تصویری شامل تصاویری متنوع از نظر فاصله و زاویه عکس برداری مورد آزمایش قرار گرفت. نتایج به طور متوسط ۸۹/۹۵ درصد تشخیص درست منطقه پلاک را نشان می دهد که حدود ٦ درصد بیشتر ازدیگر روشهای تشخیص پلاک مبتنی بر مورفولوژی است . علاوه بر این بخش پیش پردازش سیستم که قسمت عمده بار محاسباتی را در یک سیستم تشخیص محل پلاک به عهده دارد برروی سخت افزاری پیاده سازی گردید نیز ارزیابی گردید.

doi: *10.5829/idosi.ije.2013.26.11b.10*