

## An Evaluation of License Plate Recognition Algorithms

Hadi Sharifi Kolour

Department of Information Technology  
University of Guilan  
Rasht, Iran  
[Sharifiha@gmail.com](mailto:Sharifiha@gmail.com)

Asadollah Shahbahrami

Department of Computer Engineering  
University of Guilan  
Rasht, Iran  
[shahbahrami@guilan.ac.ir](mailto:shahbahrami@guilan.ac.ir)

**Abstract—***In the last decades vehicle license plate recognition systems are as central part in many traffic management and security systems such as automatic speed control, tracking stolen cars, automatic toll management, and access control to limited areas. There are many techniques for license plate detection. The goal of this paper is study and evaluate some most important LPD algorithms and compared them in terms of accuracy, performance, complexity, and their usefulness in different environmental condition. This evaluation gives views to the developers or end-users to choose the most appropriate technique for their applications. Our study and investigation show that the dynamic programming algorithm is the fastest and the Gabor transform is the most accuracy algorithm compared to other algorithms.*

**Keywords—***License Plate Recognition, Digital Image Processing, Computer Vision.*

### 1. INTRODUCTION

License Plate Recognition (LPR) is an important function in intelligent traffic control systems. These systems have many applications such as parking management systems, access control, border control and monitoring, and tracking vehicles.

There are many algorithms for LPR. Some of them have a good accuracy with more complexity than others. Some of those algorithms are computationally intensive. However, selecting one of them based on some criteria such as execution time, memory usage, complexity of the algorithm, and its accuracy in different situation is a challenging problem. Based on that the purpose of this paper is to study, to investigate and to evaluate some important LPD algorithms and compared them in terms of complexity, execution time, and their accuracy in different situations. In addition, some practical criteria are also considered. Some practical aspects of algorithm are as follows [8,16,17].

- ❖ The ability to use in embedded systems or SoC due to the minimum hardware requirements for running.
- ❖ Applicable in real-time systems due to high processing speed and accuracy.
- ❖ The ability to use in both indoor and outdoor environments due to independency of lighting and weather conditions.

The LPR is used in real-time systems; it should provide both accuracy and acceptable response time [1,15]. Most of the LPR systems are based on image processing techniques and character recognition systems [7, 9, 20, 23, 27]. Each LPR system consists of three basic sections namely, image acquisition, License Plate Detection (LPD), and Optical Character Reader (OCR). The image acquisition section receives a signal from a motion sensor and captures an image using a camera. In order to reduce motion blur it should use a high speed shutter.

The LPD section of the system analyzes the captured image to find plate location or alphanumeric characters. Some algorithms are based on finding the license plate by using image features such as the shape, color, or height-to-width ratio. The performance of these algorithms is very sensitive to changes in environmental conditions such as light or weather conditions that affect the quality of image features. Third part segments the characters and uses an OCR module to read the segmented characters that appear in the plate [2]. The object recognition systems include two functions, detecting the object in a scene and recognizing that object [3]. Most of image processing techniques for the LPD are based on the neural networks, Gabor transform or Hough transform, and Ada-Boost models. But there are some common

features affecting on performance of these algorithms including [13, 22]:

- Lighting conditions such as cloudy weather, night working hours, reflecting sunlight or car's taillight or front light.
- Complex background which effects on speed of detecting the real region of plate.
- Damaged or dirty license plates that causes to gain fault identification numbers.
- Varying view angle and efficient distance between camera and moving car.

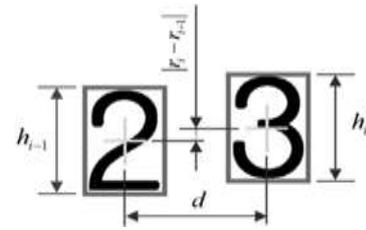
Therefore, in describing the algorithms we point to practical criteria in addition to technical measures such as run time and performance.

This paper is an extended version of our conference published paper in [14] with more evaluation and comparative study than it. This paper is organized as follows. We discuss some LPD algorithms such as dynamic programming-based, Hough transform, Gabor transform, morphology-based algorithms, AdaBoost algorithm, in Section 2, 3, 4, 5, and 6, respectively. Section 7 describes a combination of Edge-based and color aided model. We compare and discuss different algorithms in Section 8. Finally, conclusions are drawn in Section 9.

## 2 Dynamic Programming-Based Method

In the Dynamic Programming-based (DP) algorithm [4], developers never need to find license plate location in the image. This is because it segments the alphanumeric characters directly on the license plate. It does not also require any image features of the license plate such as edges, colors, or lines, which are always affected by intensity variations.

To implement this algorithm, a wide range of threshold values are considered to detect blobs, which are containing license plate number. As indicated in [4], the threshold value starts from 10 and increases at intervals of 10 and ends to 240 in gray-level values. Each blob has some key specifications like the height, the width, coordinates of center position, and the threshold value used for extracting it. There is an energy minimizing framework to extract the correct blobs. The vertical and horizontal distance between the center positions of two neighboring characters must be minimized. If two neighboring blobs with the center distance  $d$ , as shown in Fig. 1, are within the permitted range, then they can be considered as correct candidates for part of the numeric characters. In practice, the permitted distance is considered in the range from 10 to 30 pixels.



**Fig. 1.** Geometric specification between two neighboring numerical character blobs. Horizontal distance between two blobs is shown with ( $d$ ) [4].

The blob extraction module uses most of the computational time, because the threshold values are changed repetitively and image labeling algorithm must be executed once for each threshold value. The DP algorithm has a low computational time. Based on that, it is known as a fast algorithm that makes it suitable for real-time systems. Experiments results show that DP consumes 3 msec for blob extraction that is depicted in Table 1 [4]. Its accuracy is up to 97.14%.

# of blobs	Found threshold value	Blob extraction computing time(sec)
316	110,120	0.296
333	120	0.281
303	80,90,100	0.11
418	100	0.218
406	90,100	0.172
256	120,130	0.141

**Table 1.** Performance of dynamic programming-based algorithm to extract license plate numbers [4].



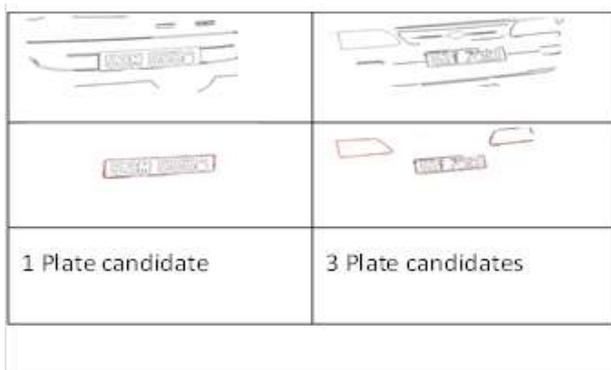
**Fig. 2.** Images in the top of the figure are taken in daytime and bottom ones are taken in nighttime. Both images have been used in evaluation of the DP-base system [4].

Some test images taken at daytime and nighttime are illustrated in Fig. 2. At the bottom of each image, the extracted numbers are shown in the blobs. The DP algorithm can be used in both indoor and outdoor situations because of its independency of lighting and weather conditions.

### 3. Hough Transform

Second algorithm is based on combination of Hough transform and contour algorithm [1, 25]. It is one of the most efficient algorithms to detect lines from binary images. It looks for regions containing two parallel lines which considered as plate candidates. Execution time is a disadvantage of the Hough transform. It requires too much computation when being applied to a binary image with high resolution. In other words, the computational time for high resolution images are so much. Although, image thinning pre-processing can improve the algorithm's speed, Hough transform computational time is still high and it is difficult to use it for real-time traffic management systems.

In order to improve the performance, Hough transform is combined with contour algorithm. From the extracted edging image, it uses the contour algorithm to detect closed boundaries of objects. These contour lines are transformed to Hough coordinate to find two interacted parallel lines (one of 2-parallel lines holds back the other 2-parallel lines and establishes a parallelogram-form object) that are considered as a plate-candidate. Since there are quite few (black) pixels in the contour lines, the transformation of these points to Hough coordinate requires much less computation. Hence, the speed of the algorithm is improved significantly without the loss of accuracy as shown in Fig. 3.



**Fig. 3.** Combination of contour algorithm and Hough transform for the LPD system [1].

However, this technique may detect the headlights or windscreen falsely as license plate candidates. This is because they have parallelogram shape. Candidates for license plate could be evaluated by a module to reject incorrect ones, and the true one remains. From the two horizontal lines of a candidate, it can calculate exactly how inclined the line was from horizontal coordinate. Then it applies a rotate transformation to adjust it to straight angle. After processed, these straight binary plate-candidate regions were passed to a number of heuristics and algorithms for evaluation. The evaluation algorithm of the license plate candidates is based on two main steps which are as follows.

- The ratios of width to height are checked. These ratios should be in the permitted range. If we consider the width as W, the height as H, and the permitted range as (minWHRatio, maxWHRatio) then:

$$\min WHRatio < \frac{W}{H} < \max WHRatio \quad (1)$$

- Evaluate the candidates by counting objects cut using horizontal crosscuts. There is a predefined range of objects for each desired license plate type which should be cut in a horizontal line on the plate. Checking this property is useful to evaluate the candidates.

### 4. Gabor Transform

Gabor transform is also used for LPR [2]. It is a computer vision system that detects license plates and segments license plate into characters in an image by using the Gabor transform in detection and vector quantization in segmentation. The Gabor filter is a texture analysis tool. The texture of whole plate and its components including alphanumeric characters is used for detecting desired location. The benefit of this technique is texture analyzing in all directions and scale. The filter responses that result from the convolution with Gabor filters are directly used as license plate detector. Three different scales including 9, 11, and 15 pixels, and four directions including 0°, 45°, 90°, and 135° are used, resulting in a 12 Gabor filters. Fig. 4 shows an intensity image and its Gabor filter response. High values in the image indicate the probable plate regions.

In order to segment these regions, first threshold algorithm is applied and the binary image is produced. Then, the morphological dilation operator used to the binary image in order to merge neighboring regions. Finally, the license plate regions are simply extracted. The complexity of this algorithm for an image of size

$N \times N$  with a filter size of  $W \times W$  with consideration of a fixed angle is  $N^2W^2$ .



**Fig. 4.** Left images are original images and the right images are the outputs of the Gabor filter responses [2].

The implemented results of the Gabor filter and its accuracy is depicted in Table 2. In this test different images from lighting conditions including day and night have been used.

Operation	Rate	Result
LPD Performance	294/300	98%
LPS Performance	2029/2154	94.2%

**Table 2.** Performance summary for Gabor filter.[2]

Gabor-based method using multi-scale and multi-directional capabilities is more suitable for outdoor environments. It can identify license plate numbers with no restrictions on view angle or distance.

## 5. Morphology-Based Algorithm

Morphology is an image processing tool, which is based on shapes [5, 6, 18]. Each shape has a Structural

Element (SE) and morphological operators use this SE to analyze digital images. License plate has the rectangle shape, so morphology is suitable for LPR because of the rectangle shape of plates. By using this method, many candidates may be detected. To evaluate candidates and reject false ones and find the license plate location some features such as shape, aspect ratio, and width to height ratio are checked.

The LPR procedure based on morphology operators is described as follows.

- Preprocessing stage containing a convert of RGB image to gray scale image. If original image is taken in gray scale then this preprocessing step is not needed.
- Applying the Sobel edge detection algorithm on the gray scale image.
- Edge dilating operator (morphology)
- Close operator (morphology)
- Noise cleaning

As mentioned, the edge detection algorithms find candidate edges based on optical features and brightness characteristics of the image. Efficiency of this method may drops in complex images. Because, this method is very sensitive to non-dependent and additional edges and offers them as license plate candidates. But according to use of morphology operators and eliminating the extra edges, it leads to high-speed detection of the license plate region.

This algorithm applies a set of 105 test images taken from a camera and almost equal distance. The average execution time of algorithm is 5 seconds and its accuracy is 80.39%. In compared to discussed algorithms, it is more computational intensive. However, its implementation is simple.

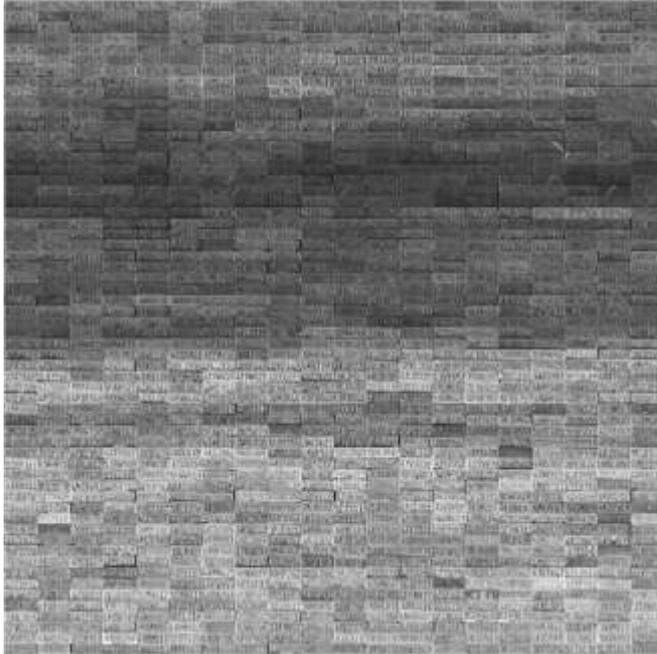
## 6. AdaBoost Algorithm

The AdaBoost stands for Adaptive Boosting, is an algorithm that takes a large number of weak classifiers and selects some of them to construct a set of strong classifiers [3]. AdaBoost in addition to license plate detection has the ability of face detection and recognizing the objects in a scene. The algorithm is based on system training and needs a training dataset containing a large set of license plate images which manually extracted from a set of images including cars.

As shown in Fig. 5, a large set of training images taken in a variation of light and illuminating conditions used to scan the scene. As illustrated in the figure, top regions that are darker than other areas are extracted from images taken at night or cloudy weather conditions.

The system proposed in [3] has a training set containing 158 images. Such a system is able to detect

95.6% of true license plates. Using more classifiers can improve the detection rate and increase performance of the algorithm. Nonetheless, the AdaBoost is slower than edge-based methods and is very sensitive to view angle and distance.



**Fig. 5.** A large set of training images [3].

According to its long running time, using the AdaBoost in real-time applications is not justified. The rate of identification of correct license plates can be increased by boosting the classifier. So, it can be applied in non-real-time applications to identify license plates correctly. The system needs a large set of training images that occupies lots of memory to run. Consequently, it also cannot be used in embedded systems and is suitable for workstation application software.

## 7. Edge-based and Color-aided Model

The proposed method in [10, 11, 12] involved two basic steps as follows: first step is preprocessing, that is a technique to improve the quality of input image using stretching the image contrast, and second is a fast algorithm to eliminate invalid license plate candidates. In the first step that can be called as preprocessing step, the system achieves to the goal of quality enhancement using intensity variance and edge density. Image enhancement using local intensity variance is based on the principle that the constituting pixels of the license plate have been limited to a fixed range of intensity variance and they have no significant changes. A function is used to increase the image contrast in plate regions with a given value of variance intensity. In the

second step it increases the image contrast at license plate candidate regions using Sobel operator and 2D Gaussian filter.

After preprocessing stage, it is the time for identifying the location of the license plate. License plate detection is performed via four sub-steps involving: vertical edge density estimation, designing a matched filter, region extension procedure, and license plate extraction using morphological processing. Finally the supplementary stage begins which includes color analysis of the license plate. The color pattern of license plate differs based on regional designing regularity [19, 24]. The vertical edge density estimation uses a low threshold for edge detection to avoid missing plate edges. The matched filter enforces the constancy of intensity values within plate region along horizontal direction. Due to irregular shape of the candidates, all these regions are extended to facilitate detecting the true one. Knowing the geometric characteristics of the license plate, such as shape, aspect ratio of sides, and area of plate region, and using morphological operators the genuine license plate can be extracted from a set of multiple candidates. In order to enhance system accuracy, especially in images with complex background, the image's color data analysis is used. For this purpose, the standard formats used in the license plates, which are separated in country or region level, are used. For example, license plates belonging to the European Union member countries have a unique color format. In a country like Iraq, the provincial license plates have been separated from each other. Considering plates color format in each region the system is implemented, the accuracy of search filter can be improved.

## 8. Discussion

The DP algorithm does not need any edge detection algorithm. The environmental conditions do not almost affect the image features. In other words, its performance is assured in day-night and every weather conditions. The DP does not consume processor time-cycle for converting a gray scale image to binary and neither for edge detecting. Its two strength points are first, the capability of using in real-time applications, and second, the ability of applying both in indoor and outdoor environments.

The Hough transform algorithm is a time consuming method. In order to improve its performance thinning algorithm must be executed in the preprocessing stage.

Gabor transform has high performance in both LPD and LPS sections due to the ability of multi-scale and multi-direction execution and it is desirable for outdoor environments. Its only drawback is high execution time

regarding to its computational complexity and that is an obstacle in real-time applications.

Morphology based algorithm has the lowest accuracy and executing speed, but it has a simple implementation.

The AdaBoost has a simple implementation and training step is very fast, but it is slower than edge-based methods. Furthermore, it has another drawback that is high sensitivity to distance and view angle. Given these assumptions, Adaboost is not suitable for real-time applications; on the other hand, due to the large volume of required hardware, it is also not used in embedded systems. However, due to its high accuracy, it is a good option for non-real-time applications.

The Edge-based and Color-aided method has a simple implementation and less complexity of the algorithm that make it suitable for real-time applications. Regarding to sensitivity of the introduced method in distance and view angle, always a restriction should be considered in its implementation. Its other disadvantage is the use of threshold in the image improving step. So that, it affects directly on computing time and has opposite impact on system performance. The weakness of the method is long execution time due to complexity of color analysis step. This drawback makes it suitable for real-time applications. Table 3 compares discussed *algorithms* based on four basic factors.

## 9. Conclusion

Intelligent control traffic systems use License Plate Recognition function. There are many algorithms for LPR. In this paper, six LPR techniques, dynamic programming, Hough transform, Gabor transform, Morphology-based, AdaBoost, and edge-based model have been discussed and compared to each other based on different criteria. Our study shows that the dynamic programming algorithm is the fastest and the Gabor transform is the most accuracy algorithm. In our future work we focus on hardware implementation of these algorithms.

Morphology	Low	High	Yes	High
AdaBoost	Low	High	No	Low
Edge-based & Color-aided	Medium	Medium	Yes	High

**Table 3.** Comparing LPR algorithms based on four basic factors.

## References

- [1] Duan, T. D, Hong, T. L., Hoang, T. V., "Building an Automatic Vehicle License Plate Recognition System", International Conference on Computer Science, Research, Innovation, and Vision for the Future, 2005.
- [2] Kahraman, F., Kurt, B., and Gokmen, M., "Licence Plate Character Segmentation Based on the Gabor Transform and Vector Quantization", International Symposium on Computer and Information Sciences, 2003.
- [3] Dlagnekov, L., "License Plate Detection Using AdaBoost", La Jolla: Computer Science Engineering Department, University of California San Diego, March 2004
- [4] Kang, D. J., "Dynamic Programming-based Method for Extraction of License Plate Numbers of Speeding Vehicles on the Highway", International Journal of Automotive Technology, Vol. 10, No. 2, pp. 205–210, 2009.
- [5] Kasaei, S. H., Kasaei, S. M., and Kasaei, S. A., "New Morphology-Based Method for Robust Iranian Car Plate Detection and Recognition", Internatoinal Journal of Computer Theory and Engineering, Vol. 2, No. 2, April 2010.
- [6] Martin, F., Garcia, M., Alba, J. L., "New Methods for Automatic Reading of VLP's (Vehicle License Plates)", International Conference on Signal Processing Pattern Recognition and Applications, 2002.
- [7] Zhang, H., and He., X. "A Fast Algorithm for License Plate Detection in Various Conditions", International Confense on Systems, Man, and Cybernetics, October 2006.
- [8] Arth, A., Limberger, F., and Bischof, H., "Real-Time License Plate Recognition on an Embeded DSP-Platform", International Conference on Computer Vision and Pattern Recognition, 2007.
- [9] Anagnostopoulos, C.N., Anagnostopoulos, I., Loumos, V., and Kayafas, E., "A license Plate Recognition Algorithm for Intelligent Transportation System Applications", IEEE Transactions on Intelligent Transportation Systems, Vol. 7, No 3, pp. 377-392, 2006.
- [10] Broumandnia, A., and Fathy, M., "Application of Pattern Recognition for Farsi License Plate Recognition", International Journal of Graphics Vision and Image Processing, Vol. 5, pp. 25-31, 2005
- [11] Abolghasemi, V., and Ahmadifar, A., "An edge-based color-aided method for license plate detection", Journal of Image and Vision Computing, Vol 27, pp. 1134-1142, 2009
- [12] Lee, E.R., Kim, P.K., and Kim, H.J., "Automatic Recognition of a Car License Plate Using Color Image Processing", IEEE International Conference on Image Processing Austin, pp. 301-305, 1994
- [13] Caner, H., Gecim, H.S., and Alkar, A.Z., "Efficient Embeded Neural-Network-Based License Plate Recognition System", IEEE Transactions on Vehicular Technology, Vol. 57, No. 5, pp. 2675-2683, 2008

Method	Implementation complexity	Sensitive to environmental conditions	Edge detecting	Computational time
Dynamic Programming	High	Low	No	Low
Hough+Contur	Medium	Low	Yes	High
Gabor transform	Low	High	Yes	High

- [14] Sharifi, H., and Shahbahrami, A., "A Comparative Study on Different Algorithms of License Plate Recognition Systems", International Conference on Digital Information and Communication Technology and its Applications (DICTAP), Lecture Notes on Computer Science, CCIS, Vol. 167, pp. 691-696, 2011
- [15] Bailey, D.G., Irecki, D., Lim, B.K., and Yang, L., "Test Bed for Number Plate Recognition Applications", Proceeding of the IEEE International Workshop on Electronic Design, Test and Application, 2009
- [16] Bellas, N., Chai, S.M., and Linzmeier, D., "FPGA Implementation of A License Plate Recognition SoC Using Automatically Generated Streaming Accelerators", 20<sup>th</sup> International Parallel and Distributed Processing Symposium (IPDPS), page 8, 2006
- [17] Kang, J., Kang, M., Park, C., Kim, J., and Choi, Y., "Implementation of Embedded System for Vehicle Tracking and License Plate Recognition Using Spatial Relative Distance", 26<sup>th</sup> International Conference on Information Technology Interfaces, 2004
- [18] Zamani, A., and Movahedi, M., "Car Plate Identification Using Morphology and Constatnt Moments Transform", Conference Ferdowsi University, Mashhad, Iran, 2007
- [19] Chen, Z., Wang, G., Lin, C., "Automatic License Plate Location and Recognition Based on Feature Saliense", International Journal of Computational Cognition, 2007
- [20] Ozbay, S., and Ercelebi, E., "Automatic Vehicle Identification by Plate Recognition", World Accademi of Science Engineering and Technology, 2005
- [21] Deb, K., Chae, H., and Jo, K., "Vehicle License Plate Detection Method Based on Sliding Concentric Windows and Histogram", Journal of Computers, Vol. 4, No. 8, 2009
- [22] Akoum, A., Daya, B., and Chauvet, P., "Two Neural Networks for License Number Plates Recognition", Journal of Theoretical and Applied Information Technology, 2005
- [23] Tahir, A., Habib, H.A., and Khan, M.F., "License Plate Recognition Algorithm for Pakistani License Plates", Canadian Journal on Image Processing and Computer Vision, Vol. 1, No. 2, 2010
- [24] Mahini, H., Kasaei, S., and Dorri, F., "An Efficient Features-based License Plate Localization Method", Proceeding of IEEE International Conference on Pattern Recognition, 2006
- [25] Kamat, V., and Ganesan, S., "An Efficient Implementation of the Hough Transform for Detecting Vehicle License Plates Using DSP's", Real Time Technology and Applications Symposium, Page 58, 1995
- [26] Parasuraman, K., and Kumar, P., "An Efficient Method for Indian Vehicle License Plate Extraction and Character Segmentation", IEEE International Conference on Computational Intelligence and Computing Research, 2010
- [27] Nguyen, C., Ardabilian, M., and Chen, L., "Unifying Approach for Fast License Plate Localization and Super-Resolution", 2010