Intelligent License Plate Detection Using Genetic Algorithm

Jawad Muhammad[1], Halis Altun[2]

[1]Electrical and Computer Engineering, Graduate School of Mevlana University, Konya, Turkey
jmuhammad@mevlana.edu.tr
[2]Electrical and Electronics Engineering, Karatay University, Konya, Turkey
halis.altun@karatay.edu.tr

Abstract

In this paper, genetic algorithm is proposed that performs plate detection by randomly scanning an input image using a fixed detection window repeatedly, until a region with the highest evaluation score is obtained. The performance of the genetic algorithm is evaluated based on the area coverage of pixels in an input image. It was found that the GA can cover up to 90% of the input image in just less than an average of 50 iterations using 30x130 detection window size, with 20 population members per iteration. Furthermore, the algorithm was tested on a database that contains 1537 car images. Out of these images, more than 98% of the plates were successfully detected.

1. Introduction

Plate detection is automatic and efficient localisation of a license plate region in a given image. It is the first step in performing automatic license plate recognition systems. Researches in plate detection can be categorised into two main approaches; the region-based approaches and the pixel-to-pixel approach. In the region based approach, the input image is divided into regions based on some specified properties of the plate. All the regions are then evaluated and a region is selected from the generated regions as a candidate plate region. In a typical region based approach, morphological operations are the commonly used methods. Mathematical morphology coupled with edge statistics methods have proven to be effective in performing plate detection [1] [2] [3] [4]. In the pixel-to-pixel approach, every pixel in a given image is evaluated by selecting its neighbouring pixels to form a rectangular region with the pixel being the upper most left pixel in the rectangle. The pixel to pixel operation involves scanning the whole input image using a detection window. Learning based methods for plate detection perfectly fit into this category [5-8]. In the learning based approach, plate detection process is performed by scanning the input image from pixel to pixel, by placing a detection window at every pixel. The region within the detection window is evaluated using a trained classifier and the region with the best classification score is chosen as the candidate plate region. In this approach, the whole image space is scanned sequentially to cover all possible candidate regions. Louka Dlagnekov (2004) finds the vertical and horizontal derivatives of the plate image as features and uses a classifier trained with Adaboost to classify parts of an image at a given pixel within a detection window as either license plate or non-license plate [5]. Ho, W. T. et al. (2009) proposed a two stage learning based license plate detection method by using Adaboost classifier in the first stage, followed by SVM classifier with SIFT features in the second stage [6]. Assis da Silva, et al. (2013) proposed SIFT features with template matching for detecting the plate region in a particular image [7]. Ashtari, A. H. et. al. (2014) proposed template matching, based on colour features of an Iranian license plate to detect the plate in the image [8]. Wang, R. el. al. (2014) utilises a classifier trained by Gentle AdaBoost based on the HOG features to detect a plate region [9]. Anagnostopoulos et al. (2006) proposed a pixel-to-pixel scanning of input image with 2 sliding concentric windows in segmenting the plate region of an image [10].

Another variation of the pixel-to-pixel approach is the use of optimization based algorithms which does not sequentially scan an input image. Instead, randomly selected pixel locations are considered as potential solutions domain and an optimal solution is then found by optimizing a cost function in the form of a pixel location which in turn serves as the detected plate region. One of the optimization approaches is to use Genetic Algorithm (GA) to repeatedly select pixel locations randomly, and then evaluate rectangular regions at the locations [11-12]. The pixel location, which minimizes the defined cost function and its associated region, is selected as the plate region. Avcı, G.,et. al. (2008) proposes the use of GA to detect plate region by randomly sampling pixel location [11]. They first extracts the statistical features at the selected location of input image and then identifies the region with statistical features that is closely similar to the plate as the detected plate region. Similarly, Peker, M. et. Al (2008) used the same approach in order to evaluate the effect of GA parameters on the quality of the solution [12].

A comprehensive survey on the various methods and techniques for plate detection proposed by researchers is presented in [13-14].

In this work, GA method had been employed that perform plate detection by repeatedly selecting points within the image randomly, evaluating regions at these points and selecting regions that gives the best similarity score. Unlike in [11], that employs feature extraction based on edge statics, histogram of oriented gradient based features are used in this work.

In the next section, the detail of the algorithm is explained. Implementation and Results follows in subsequent sections.

2. Proposed Algorithm

The proposed algorithm comprises of two parts:

- Generation of candidates regions
- Evaluation and selection of the best candidate as detected plate

2.1. Candidate regions generation

Candidate regions are generated by random selection of regions in a given input image. The selection process is guided by a candidate evaluation algorithm which is based on the similarity score of the regions, explained in the next section. Genetic Algorithm (GA) is employed for this task. With GA, the input image is scanned repeatedly. Points within the input image are randomly selected. A detection window is then placed at each of the selected points and the pixels covered by the window are considered to be within one region. The regions generated from all the points make up the list of the candidate regions. The candidate regions are then evaluated and similarity score is assigned by the evaluation function discussed in the next section. The pseudo code of Genetic Algorithm is shown in Table 1. From the pseudo-code, candidate region comprises of two parts (the row and column number of its top left corner pixel). Fig. 1 describes the composition of a candidate region and some sample candidate regions.

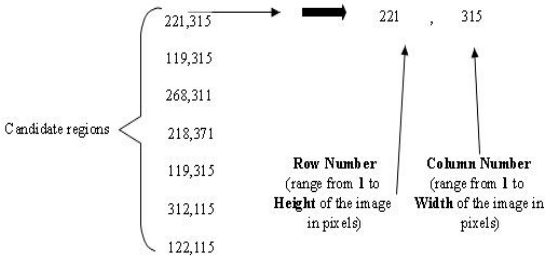
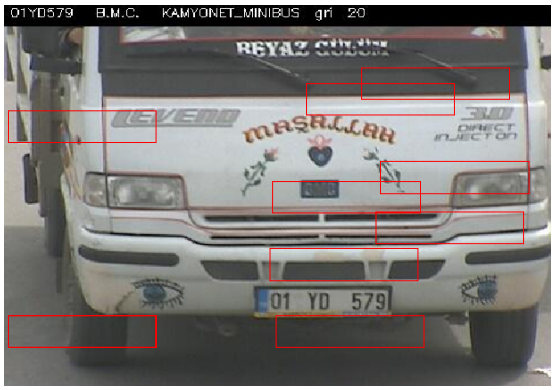


Fig. 1. Candidate region member composition

Fig. 2 depicts the GA in action at a particular iteration with rectangles representing generated candidate regions. In each iteration, the maximum number of candidate regions generated are limited to a certain fixed value termed in Table 1 as *p*. Also, the size of the detection window is fixed at all iterations.



.

Fig. 2. GA in action showing generated candidate regions

2.2. Evaluation and selection of the candidates

The evaluation function assigns similarity scores to each of the generated candidate members. As it is highly difficult to devise an analytical function, which measures the degree of similarity of the candidate region as a plate, a neural network is used, which generates a similarity score. The evaluation involves extracting the features of each candidate member. The features are based on Histogram of Gradient (HOG) obtained by exploiting the structural information of the plate region. A plate is assumed to be structured into sections in form of groups separated by a space. HOG is computed for each section and the HOG vectors for all the sections are concatenated to form the feature vector. Each feature vector extracted, is assigned similarity score using a trained neural network.

Table 1: Pseudo-Code Of The Ga Used For The Prposed Algorithm

GA(*Fitness, Fitness_threshold, p, r, m****)***

Fitness: Evaluation function that assigns an evaluation score, given a candidate region.
Fitness_threshold: A threshold specifying the termination criterion.
p*: Number of generated candidate regions at each iteration.*
r*: The fraction of the population to be replaced by Crossover at each step.*
m*: The mutation rate.*

- ***Initialize population:*** *P*← *Generate p* candidate regions *at random*
- ***Evaluate:*** *For each* h *in P, compute* $Fitness(ci)'$
- ***While*** $[\max_c Fitness(c)] <$ *Fitness_threshold* ***do***

Create a new generation, Ps:

1. ***Select:*** *Probabilistically select* $(1-r)p$ *members of P to add to* P_s*. The probability* $\Pr(hi)$ *of selecting* chromosome ci *from P is given by*

$$\Pr(ci) = \frac{Fitness(ci)}{\sum_{j=1}^{p} Fitness(cj)}$$

2. Crossover: *Probabilistically select* $\frac{r.p}{2}$ *pairs of* candidate regions *from P, according to* $P(ci)$ *given above. For each pair,* $(c_1 - c_2)$*, produce two offspring by applying the Crossover operator.*
Add all offspring to P_s*.*
3. ***Mutate:*** *Choose* m *percent of the members of* P_s *with uniform probability. For each, invert one randomly selected bit in its representation.*
4. ***Update:*** *P*←P_s*.*
5. Evaluate: *for each* h *in P, compute* $Fitness(ci)$

Return the candidate region *from P that has the highest fitness*

.

The neural network is trained by a manually cropped license plate described in [15]. At the end of iteration during the GA execution, the evaluation function is called and all the GA candidate members are evaluated and similarity scores assigned. The value is between 0 and 1 corresponding to probability or likelihood of a region being a plate region or not. Candidate with values close to 1 are more likely to be plate regions. And those with values close to 0 are more like to be non-plate regions. The member with the highest similarity score is selected and stored as potential detected plate region. If the similarity score of this selected member is greater than or equal to 0.999 or the maximum iteration is reached, this selected member is returned as the final detected plate. Fig. 3 shows the result at the end of the GA operation where the member with the highest similarity score is marked in green with a similarity score of over 0.999. During evaluation, any candidate member not fully contained within the image will not be evaluated. Hence, all candidate members at the borders of the image are automatically assigned a negative similarity score making them the least favourite to be selected as a plate.

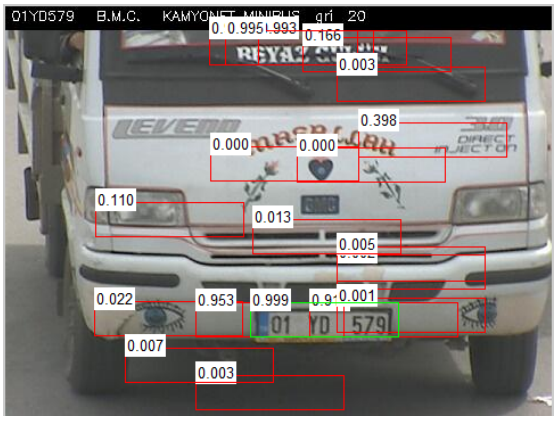


Fig. 3. Final result after the GA execution

3. Implementation Results

The proposed algorithm is implemented on a computer with 64 bit Windows 7 enterprise operating system; 4 GB of RAM; Intel® Core™ i5 -2400 CPU at 3.10GHz 3.10 GHz Processor. The database used for testing is a collection of car images containing mainly Turkish license plates described in [15]. The database contains a total of 1537 images with only one license plate in each of them.

Values of parameters used for the implementation of the proposed algorithm are as shown in Table 3

Table 3. Values of parameters used

Fitness_threshold	*Maximum_Iteration*	*p*	*r*	*m*
0.999	*70*	20	*0.6*	*0.6*

The proposed algorithm is evaluated based on the average coverage of image pixels over 20 runs. The result is depicted in Fig. 4. It can be observed that the algorithm covered almost more than 80 % of the image pixels in less than 20 iterations. Also the detection of plate is successfully completed on an average of 18 iterations. This is more efficient that performing a sequential scan of the image whereby all the pixels have to be covered.

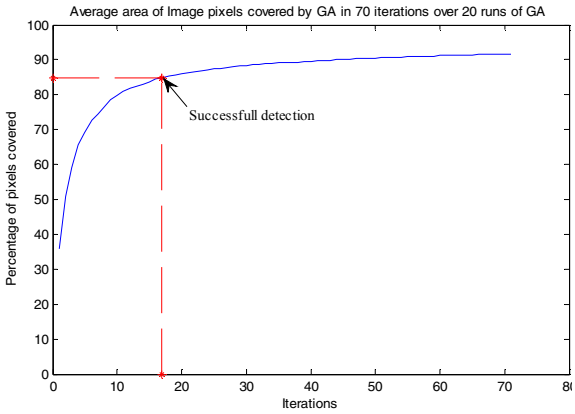


Fig. 4. Average area of image pixels on one image covered by the proposed GA over 20 runs at a maximum iterations of 70, 20 candidate members and 30 x 130 detection window size

The proposed algorithm is tested with the database that contains 1537 images. The results are obtained by running the proposed algorithm 5 times and taking the average of the 5 results, which is shown on the Table 2. It can be observed that overall success rate of the algorithm is over 98%.

Table 2. Detection Rates of running our proposed algorithm 5 times

No. of Images	**Detection Rate (%) for 5 runs**					**Average**
	1.run	**2.run**	**3.run**	**4.run**	**5.run**	
1537	98.50	98.76	98.83	98.83	98.83	**98.75**

4. Conclusion

In this paper, we present an approach of using GA to effectively detect license plate with over 98% accuracy. We evaluate the GA performance in scanning the input image and it covers over 90% of the image in less than 50 iterations using 20 candidate members and 30 x 130 detection window sizes

7. References

[1] M. Fernando, M. Garcia & J. L. Alba "New methods for automatic reading of VLP's (Vehicle License Plates)", in Signal Processing, Pattern Recognition, and Applications, Crete, SPPRA, 2002, pp. 126-131.

[2] H. Bai, and L. Changping. "A hybrid license plate extraction method based on edge statistics and morphology." in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 2, pp. 831-834.

[3] Z, Danian, Y. Zhao, and J. Wang. "An efficient method of license plate location, Pattern Recognition Letters." *ISSN 0167-8655, 10.1016/j. patrec*14 (2005): 2431-2438.

[4] W. SherrZheng, and H. Lee. "Detection and recognition of license plate characters with different appearances." In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 2, pp. 979-984. IEEE, 2003.

[5] D. Louka. "License plate detection using adaboost." Computer Science and Engineering Department, San Diego (2004)

[6] H. W. Teng, H. W. Lim, and Y. H. Tay "Two-stage license plate detection using gentle Adaboost and SIFT-SVM." In *Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on*, pp. 109-114. IEEE, 2009.

[7] D. Silva, F. Assis, A. O. Artero, M. S. V. D. Paiva, and Ricardo Luis Barbosa. "ALPRs-A new approach for license plate recognition using the SIFT algorithm." *arXiv preprint arXiv:1303.1667* (2013).

[8] A. A. Hossein, M. J. Nordin, and M. Fathy "An Iranian License Plate Recognition System Based on Color Features." *Intelligent Transportation Systems, IEEE Transactions on* 15, no. 4 (2014): 1690-1705.

[9] W. Runmin, N. Sang, R. Wang, and L. Jiang. "Detection and tracking strategy for license plate detection in video." *Optik-International Journal for Light and Electron Optics* 125, no. 10 (2014): 2283-2288.

[10] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos and E. Kayafas "A license plate-recognition algorithm for intelligent transportation system applications." *Intelligent Transportation Systems, IEEE Transactions on* 7, no. 3 (2006): 377-392.

[11] G. Avci, M. K. Mehmet, H. Altun, Fuat K., and A. Mehmet "Implementation of an Hybrid Approach on FPGA for License Plate Detection Using Genetic Algorithm and Neural Networks."*INISTA 2009* (2009): 392

[12] M. Peker; H. Altun, F. Karakaya, The effect of genetic algorithm parameters on the solution of plate location detection, Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th, 1 - 4, DOI: 10.1109/SIU.2008.4632604

[13] D. Gilly and K. Raimond. "A survey on license plate recognition systems" International Journal of Computer Applications 61, no. 6 (2013): 34-40.

[14] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas "License plate recognition from still images and video sequences: A survey." *Intelligent Transportation Systems, IEEE Transactions on* 9, no. 3 (2008): 377-391.

[15] J. Muhammad, "Implementing An Improved Intelligent Licence Plate Detection System Using Image Processing And Pattern Recognition Algorithms", M.S. thesis, Comp. Eng., Mevlana Univ., Konya, Turkey, 2014.