License Plate Detection based on Genetic Neural Networks, Morphology, and Active Contours.

Joaquín Olivares, Jose M. Palomares, Jose M. Soto, and Juan Carlos Gámez

Department of Computer Architecture, Electronics, and Electronic Technology.
University of Córdoba, Córdoba, Spain
{olivares,jmpalomares,jmsoto,jcgamez}@uco.es

Abstract. This paper describes a new method for License Plate Detection based on Genetic Neural Networks, Morphology, and Active Contours. Given an image is divided into several virtual regions sized 10×10 pixels, applying several performance algorithms within each virtual region, algorithms such as edge detection, histograms, and binary thresholding, etc. These results are used as inputs for a Genetic Neural Network, which provides the initial selection for the probable situation of the license plate. Further refinement is applied using active contours to fit the output tightly to the license plate. With a small and well–chosen subset of images, the system is able to deal with a large variety of images with real–world characteristics obtaining great precision in the detection. The effectiveness for the proposed method is very high (97%). This method will be the first stage of a surveillance system which takes into account not only the actual license plate but also the model of the car to determine if a car should be taken as a threat.

1 Introduction

Pattern Recognition is an Artificial Intelligence technique which allows to find and classify objects or structures contained inside bigger structures. Nowadays, there are several techniques based on Pattern Recognition, for License Plate Recognition (LPR) on digital images. These techniques are used in many systems [1], for example, in car parking access control, to measure average vehicle speed over long distances, to detect border crossings, to detect a stolen car with a list that can be updated in real time, for electronic toll collection on pay-per-use roads or for its use by the police to detect vehicle infringements.

Nowadays, most LPR systems [2] are split in two different parts: the Capture Unit which takes the image of the vehicle, and the Process Unit (PU) which carries out the recognition of the license plate. In the same way, the PU is split in two different sub–problems: License Plate Detection (LPD) and Digit/Character Recognition. In the LPD step, correct results are very difficult to obtain, due to the great variety of characteristics that can be present in each image: poor resolution, blurry images, objects obscuring (part of) the plate and different contrast due to reflection, rays of light, or shadows, among others.

In this work, a new method for LPD is proposed. This method is an hybridization of Genetic Neural Network (GNN), Morphology, and Active Contours. The article is divided in six different sections. In section 2, some methods developed by other authors are revised. The proposed method is described in section 3, showing the three main modules. The following two sections (Sec. 4 and 5) present the parameters of the model and some experimental results, respectively. Finally, in section 6, main conclusions of the article are discussed and some future research is proposed.

2 Background

Duan et al. [3] proposed an efficient boundary–based method for license plate detection combining the Hough transform and Active Contour algorithms. According to authors, this method optimizes speed and accuracy. Furthermore, the method has a preprocessing module based on three algorithms, graying, normalizing, and histogram equalization to improve the rate of LPD and a module responsible for license plate verification. The detection module was evaluated and obtained an error rate of 1.8%.
Kahraman et al. [4] presented an algorithm for LPD using the Gabor transform. In this algorithm, the images are convolved with 12 different Gabor filters. In the convolved images, the license plates show a high value of intensity and can be segmented with a thresholding algorithm, some morphological operators, and a simple utilization of an eight–connected blob. The authors evaluated the performance of this algorithm and achieved a correct detection in 98% of cases.
Cano et al. [5] proposed a method for automatic location of the license plate based on a supervised classifier. An equalization, a Sobel filtering, and a simple thresholding are previously applied to the images. Next, a feature extraction technique is applied. It consists in the processing of a small window centered on each pixel, applying PCA (Principal Component Analysis), which detects whether the pixel is an inner pixel of a license plate region or not. Finally, for license plate location, a classifier based on the *k–nearest neighbours* rule is used.
Clemens et al. [6] presented a module for LPD divided in three parts: a detector that combines the AdaBoost–algorithm with Haar–like features (this detector provides multiple detections for a single license plate), the post–processing stage which merges all those detections and the tracker stage for constraining the search of the detector to certain areas of the image.
A two–stage method based on Adaboost classifier followed by a SIFT descriptor and a SVM filter has been recently described by Ho et al. [7]. They discovered that Adaboost is able to detect almost all license plates, up to 95% of correct detections, although Adaboost provides a large number of false positives (precision, 35%). Their two–stage method still obtains high values of correct detection (92%) whilst improving the precision of the detection (91%).
Anagnostopoulos et al. [8] made a categorization of the methods for LPD into different classes. The first type of method is based on binary image processing and includes all those algorithms that are based on many different techniques like edge statistics, morphology, or connected component analysis. The second

type is based on gray–level processing and includes all those algorithms for LPD that are based on other techniques like region segmentation, image transformations, or statistical measurements. The third type is based on color processing which works with the expected plate appearance. This type contains algorithms regarding color model transformation, fuzzy sets theory, or histogram processing. The last type is based on classifiers and contains algorithms based on computational intelligence such as artificial neural networks, genetic programming, or genetic algorithms.

Feng and Shuoben [9] recently proposed a LPR method based on local shape and color. The method they proposed have some key points similar to the one proposed in this work, such as vertical and horizontal edge extraction and further thresholding of pixels based on their illuminance and color. The method also manages properly slanted and oblique plates, providing a 96.3% acuracy. In the abstract, they state that the method has little influence on light and blur, however, no further explanations are included in their work.

The GNN, used in the first step of the proposed method, is based on that presented by Montana and David [10]. These authors created a Genetic Algorithm for Neural Network weight optimization. This algorithm has five principal components: chromosome encoding, evaluation function, initialization procedure, operators, and parameter settings.

Odeh et al. [11] developed a method to classify different lesions in medical images. Several techniques were used to extract visually relevant characteristics, which were used as inputs for a Genetic Algorithm for pattern classification. Some parts of their method are based in morphology and edge extraction, which are also used in the proposed method in this work.

3 Proposed Method

In this section, a system for LPD based on GNN, Morphology, and Active Contours is described. This system is divided into three main modules: Preprocessing, Processing, and Postprocessing.

3.1 Preprocessing

In this stage, the images are processed using several different algorithms to obtain modified images or computed data, which will be used as the inputs to the GNN of the following phase. These algorithms are binary thresholding, horizontal and vertical Sobel edge filtering, Prewitt edge filtering, and white, gray, and black pixel histograms.

3.2 Processing

In this module, the preprocessed images are divided into 10×10 virtual regions (each region with the same size). The optimum size and amount of the virtual regions were determined applying a large number of experiments, obtaining the best results for that size and number of regions.

For each region, the following information is extracted using the images from the preprocessing stage: number of edge pixels obtained by the Sobel operator, amount of edge pixels obtained by the Prewitt edge filtering (applied with a lower threshold that the one used with Sobel), number of edge pixels provided by a horizontal Sobel edge detector, amount of white pixels obtained from a binarized image, and the amount of white, gray, and black pixels of the original (non binarized) image. Prior to use the GNN to detect license plates, it has to be trained. In the following paragraph, the training of the GNN will be described.

Training The extracted information is introduced in a previously created GNN for training. In each iteration of training, the algorithm obtains the information of one virtual region. The GNN processes the information and gives an output for each virtual region to indicate whether a part of the license is contained in this region or not.

The training phase is made with a set of 6–10 images and their respective virtual regions. These images have to be selected meticulously, in order to train the GNN with highly representative images of different license plates and situations. It means that in the training set of images there should be included at least one image with a sloping license plate, another one with a dark plate, another image with a very bright plate, etc. besides including some images of "standard" license plates. Using this simple hint, the system is able to obtain a quite robust GNN. Furthermore, the training phase can be accomplished fast (ranging from 21 to 35 seconds for the whole training stage in an AMD Turion 64 x2 Dual Core 1.9Ghz running Linux Ubuntu 7.04).

Execution Once that GNN is trained properly, it is able to process a large set of different images. This module also makes use of the previously defined virtual regions of each image. The information extracted of each virtual region is introduced in the previously trained GNN. It determines whether each region contains all or a part of the searched license plate. In that case, the output of the network for that region is activated. Otherwise, the output for that region is deactivated. Finally, all activated regions are joined and combined together to obtain the final region, inside of which the whole plate is located.

3.3 Postprocessing

The final region is usually bigger than the license plate, therefore, it is necessary to apply further processing of the image in order to obtain a more accurate detection of the plate.

The first step in the postprocessing stage is to obtain a binarization of the final processing–stage region. Several morphology operators are applied sequentially on the binarized image: an horizontal opening and closing and two consecutive vertical closings and openings. These operations are designed to remove all unnecessary information from the final region, like loose pixels. The result from this step is a rectangle, which is very similar and quite close to the searched plate.

After that, an active contour (or *snake*) is initialized using the contour of the rectangle as seed. With this final step, the *snake* adapts itself to the border of the license plate, with very little or no error. In Fig. 1, the output of the GNN is represented in a gray rectangle, and the output of the *snake* is represented with a set of black (or white, if the image is too dark to be able to detect black) points.

Sometimes the *snake* cannot be generated due to odd characteristics in the image: license plates too dark or too bright. For this case, the images are processed again to enhance their features. First of all, the histogram of each image is analyzed. If the image is very dark, the system makes a shift of the histogram to make the image lighter. On the opposite, if the image is very bright, the system makes a displacement of the histogram to darken the image. After that, the image is introduced again in the GNN to obtain a new output.

The whole system has an average response time of 3.87 seconds (in the same platform as the training phase) for images without second processing and 7.98 seconds for those images that need histogram displacement, because a whole new processing is needed.

4 Parameters of the Model

There are some parameters that have an influence on the obtained results:

- Activation threshold: it determines whether the output of the network is activated or deactivated. If this parameter is very close to 0, some virtual region (that should not be activated) will be activated. However, if it is very close to 1, some virtual region (that should be activated) will not be activated.
- Number of nodes per layer: nodes of each hidden layer of the network. This parameter has an influence in the accuracy of the results given by the GNN.
- Mutation probability: mutation probability of the genetic algorithm. This parameter has a direct influence with the training process.
- Training images: number of images used in the training phase of the neural network. This parameter also has an influence on the accuracy of the results given by the GNN.

The best values of the activation threshold and the number of nodes per layer have been determined with a genetic algorithm. The optimum value of the amount of training images and the mutation probability have been determined by codifying many configurations and proving each configuration of the GNN to compare which one obtains the best results. These values have the same influence in both training and execution phases.

The experiments have been carried out using a large set of images taken in many different situations: night shots with flash, light underexposition/overexposition, inclined licensed plates, different positions of the license plates, different sizes of the license plates, different models of the license plates, etc.

Training images	Nodes-Layer	Mutation Prob.	% Plates detected
1	20/10	0.1	75%
3	20/10	0.1	82%
10	20/10	0.1	75%
6	20/10	0.1	97%
6	10/5	0.1	92%
6	50/25	0.1	92%
6	100/50	0.1	91%
6	500/100	0.1	92%
6	30/30	0.1	92%
6	20/10	0.1	75%
6	20/10	0.01	94%
6	20/10	0.05	95%
6	20/10	0.15	92%
6	20/10	0.25	86%
6	20/10	0.35	85%

Table 1. Experimental Results

5 Experimental Results

In Tab. 1, the results of the experiments are shown. For each experiment, this table shows the number of images used in the training phase of the GNN, the number of nodes in the two hidden layers, the mutation probability, and the percentage of plates properly detected (with images not used in the training phase). In some cases, the genetic algorithm is not able to find an optimized neural network for the training set of images, because the training images have so different features that they produce unstable changes in the weights of the GNN. In the best case, the system provided 97% of effectiveness (the license plate was correctly detected in 194 out of 200 images). In Tab. 2, the experiment which obtained best results is described. The set of images is divided into six different types of images, according to the features they show. As can be observed, all of the types of images are above 92% of effectiveness in the detection. Moreover, several types of images achieved 100% of detection. The system was unable to provide accurate results for some night images shot with flash. This wrong behavior is due to the great amount of light reflected by the plate, which makes the digits undetectable. However, the method shows great accuracy in the rest of images.

In Fig. 1, some examples of detected plates are depicted. The first case shows an example of the detection of a standard plate. The central image shows how the *snake* has been designed to be able to expand outside the GNN output (the gray rectangle in Fig. 1), because in some cases, the region given by the GNN does not contain the whole plate. The second case presents an example of a plate with a special shape, different from the rest of the plates. In that image, the license plate is composed of two rows of digits and the *snake* adapts itself around

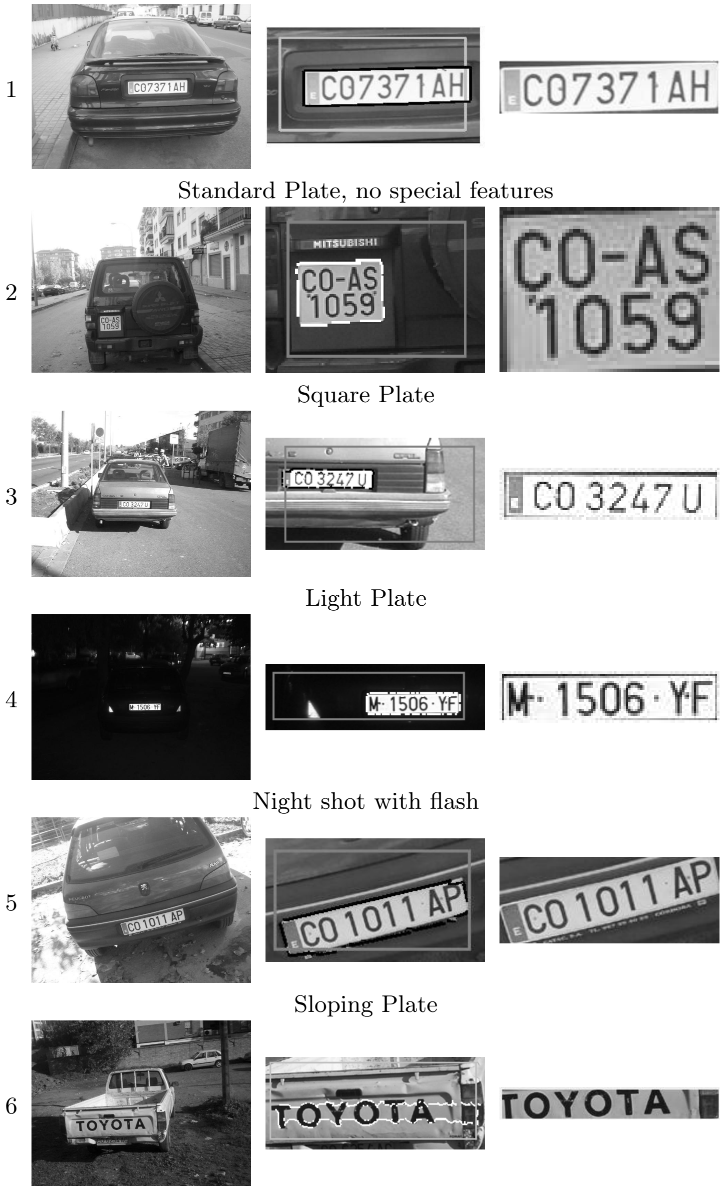
Type of images	**Total Images**	**Plates Detected**	**% Plates Detected**
Standard Images, no special features	94	92	97.9%
Images with sloping plate	21	21	100%
Images with square plate	3	3	100%
Images with dark plate	26	25	96.2%
Images with light plate	31	30	96.8%
Night images	25	23	92%
Total Images	200	194	97%

Table 2. Results of the 4^{th} experiment.

them without any problem. The third type is an example of a light overexposed image. In this type of images, the license plates show greater difference between the license plate background and the digits than in other types of images. This behavior also occurs in night shots with flash, depicted in the following row. In both types of images, the *snake* cannot be generated initially sometimes, due to the great quantity of brightness or darkness in the image. When this happens, the method analyzes the histogram to enhance the image features. The fourth case presents an example of an image taken during night with flash. In this type of images, most of the picture, except for the plate, is very dark and an histogram shifting is applied to make the image brighter. The fifth case shows a sloping plate, for which the *snake* adapts itself accurately. Finally, in some cases, the system is unable to find the proper plate as in the one shown in the sixth row. In it, a large name appears in a similar position than an standard plate and the font of the large name is quite similar to the font used in the license plate, thus, the GNN selects the large name as a plate. Therefore, in order to detect this kind of plates, further processing (i.e. digit recognition) has to be applied to reject plates that do not fit approved license plate models. This situation is not a common case, it has been specifically selected to stress the detection method to the limit. Nevertheless, if the GNN is trained with this type of images, it would be able to detect properly the plate, rejecting the large name.

6 Conclusions

In this article, a new system for LPD has been proposed. This method shows large effectiveness on plate detection, as in the best case, 97% of license plates were properly detected. Furthermore, in most of the cases, the result of the detection presented great precision, with little or even no error, and was very close to the actual license plate. Effectiveness depends on the parameters of the model, whilst precision depends basically on the pictures themselves. It is very important to note that this system provides accurate results even in bad situations, such as, (bright) overexposed images, (dark) underexposed images, and night shots with flash. The method also behaves properly with different positions and models



Standard Plate, no special features

Square Plate

Light Plate

Night shot with flash

Sloping Plate

Wrong Plate

Fig. 1. Examples of License Plate Detection

of the license plate (sloping, not centered, square plates, rectangular plates, etc.). Other authors provide better results in terms of percentage of correct detections, however, the set of images they used is unknown and, thus, it cannot be compared with the proposed method. Most of those methods are not tested against night images, oblique plates, etc. and thus, high detection values are obtained. However, this method has been proved with a real–world dataset with a large variety of cars, license plates and situations.

It is worth noting that with a very small subset of well–chosen images, the system is able to handle properly many different images with different characteristics. This has been proved using just 6 images in the training phase and obtaining a correct detection of the license plate in 194 out of 200 images. The images used in the training phase were not used in the execution phase, and thus, this method shows a quite robust behavior. This system requires little configuration in the training phase, except for fine–tuning of the parameters.

6.1 Future Research

This method is the first stage of a larger intelligent surveillance system for police and other security forces. With this method, every car could be detected and the license plate would be compared with the model of the car for which the license plate was emitted. Thus, any car with a "repeated" license plate (often used by terrorists for bomb–cars) which do not match the real model of the car, would be marked as a suspicious one.

Besides, due to the use of quite simple computation (summation, multiplication, comparisons, look–up tables, and fixed–point mathematical operations), this system is suitable for embedded development. Because of the high precision of the LPD module, this system can be used as the first step of a new module for digit recognition [12].

References

1. License Plate Recognition, http://www.licenseplaterecognition.com/
2. Automatic Number Plate Recognition, http://www.anpr-tutorial.com/
3. Duan, T.D., Hong Du, T.L., Phuoc, T.V., Hoang, N.V.: Building an automatic vehicle license plate recognition system, in Proceedings of International Conference on Computer Science RIVF, pp. 59-63 (2005)
4. Kahraman, F., Kurt, B., Gökmen, M.: License plate character segmentation based on the Gabor transform and vector quantization, in LNCS, vol. 2869, pp. 381-388 (2003)
5. Cano, J., Perez-Cortes, J. C.: Vehicle license plate segmentation in natural images, in LNCS, vol. 2652, pp. 142-149 (2003)
6. Arth, C., Limberger, F., Bischof, H.: Real-Time License Plate Recognition on an Embedded DSP-Platform, in Procs. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'07), pp. 1-8 (2007)
7. Ho, W. T., Lim, H. W., Tay, Y. H.:Two–stage License Plate Detection using Gentle Adaboost and SIFT–SVM, in 2009 First Asian Conf. on Intelligent Information and Database Systems, pp. 109-114 (2009) DOI: 10.1109/ACIIDS.2009.25
8. Anagnostopoulos, C.-N.E., Anagnostopoulos, I.E., Psoroulas, I.D., Loumos, V., Kayafas, E.: License Plate Recognition From Still Images and Video Sequences: A Survey. IEEE Transactions on Intelligent Transportation Systems, vol. 9, Issue 3, pp. 377-391 (2008)

9. Feng, J. and Shuoben, B.: License Plate Location Based on Characteristic of Local Shape and Color, CSIE, vol. 5, WRI World Congress on Computer Science and Information Engineering, pp. 279-282 (2009)
10. Montana, D.J., Davis, L.D.: Training feedforward Networks using Genetic Algorithms, in Procs. of the Intl. Joint Conf. on Artificial Intelligence. Morgan Kauffman, (1989)
11. Odeh, S. M., Ros, E., Rojas, I., Palomares, J. M.: Skin Lesion Diagnosis Using Fluorescence Images, in LNCS, Vol. 4142, pp. 648-659 (2006)
12. Fan, X., Fan, G., Liang, D.: Joint Segmentation and Recognition of License Plate Characters, Image Processing, in Procs. of ICIP 2007, Vol. 4, pp. 353-356 (2007)