Malaysian License Plate Recognition Using Artificial Neural Networks and Evolutionally Computation

Stephen Karungaru, Minoru Fukumi and Norio Akamatsu
Faculty of Engineering, University of Tokushima,
2-1, Minami-Josanjima, Tokushima 770-8506. Japan.
karunga@is.tokushima-u.ac.jp

ABSTRACT

In this paper, a license plate recognition method using neural networks and genetic algorithms is proposed. Our system assumes that license plate localization has already been accomplished. We combine two methods into a hybrid system to recognize the license plate's characters. In the first method, we train neural networks to recognize the characters and then apply a genetic algorithm to guide the neural network in the search. The genetic algorithm is necessary because of size and orientation differences between the neural network training samples and the test characters. The other method is template matching that also uses the genetic algorithm to guide its search for same reasons as in the first method. This work is performed with speed and accuracy in mind, aiming at the online usage of the system. The final system accuracy achieved is 97.3 %.

Keywords: license plate recognition, neural networks, genetic algorithms

1.0 INTRODUCTION

Vehicle Plate Recognition is an image-processing technology used to identify vehicles (usually automatically) by their license plates. Vehicle license plate detection and recognition is an important research field with many applications in traffic control, crime prevention, automatic parking authentication systems, security, etc. In license plate recognition, there are several steps required. One is the license plate localization regardless of the license plate's size and orientation. The second step is the segmentation of the characters in the plate and the normalization of other factors like brightness, contrast, illumination etc. The third step is the recognition of the characters and hence, the license plate.

This paper assumes that the first step, license plate localization is complete, and then concentrates on the last two steps. Once the plate region has been extracted, the characters regions extraction and separation must be performed. Character segmentation is vital because it dictates the accuracy of the system. Each segmented region is searched for only one character. The samples used to train the neural network and to construct the template are all manually extracted from a dataset.

Related works include a wavelet transform based method for extracting license plates from cluttered images achieving a 92.4% accuracy [1] and a morphology-based method for detecting license plates from cluttered images with a detection accuracy of 98% [2]. Hough transform combined with other preprocessing methods is used by [3] and [4]. In [5] an efficient object detection method is proposed.

The rest of this paper is organized as follows. Section 2 describes the license plates that were used in this work while section 3 describes the system design including the neural networks, template matching and the genetic algorithms used. Section 4 describes the results of computer simulations and Section 5 concludes this paper.

2.0 THE LICENSE PLATES

This work concentrates on the license plate images of vehicle in Malaysia. The images were taken at a parking lot with the car initially coming directly towards the camera and then turning towards the right. This image acquisition technique produces a maximum license plate size of about 40x100pixels. There are about 40 images per car and a total of 6444 images.

The images in the database have the following general characteristics.

1. Each of the images is 320x240pixels.
2. In some of the images, there are no cars (only the background) and hence no license plate to recognize.
3. In others, although the vehicle is visible, the license plate is occluded.

Fig. 1 shows some examples of the license plates contained in the dataset.

Fig. 1. Example of license plates in the database.

Notice the difference in the number of characters in the license plates and their lengths.

In this dataset, the license plates under consideration have the following general characteristics,

1. The license plates are all rectangular in shape with only one row of characters.
2. The characters are divided into two groups. The first group on the left consists of alphabets (except i and o) while the second group on the right is made up of numerals.
3. The number of alphabets or numbers vary. This, in effect, means that the lengths of the plates are also different.
4. For a given license plate, the size (height and width) of the characters and the numerals in it is the same.
5. The license plate's main color can either be white or black.

This work aims at the recognition of the license plates described above regardless of their different aspects highlighted.

3.0 CHARACTER SEGMENTATION

Although the license plate recognition methods discussed in this work can search for the characters inside the plate pixel by pixel, the process has a very high calculation cost, making it slow. Therefore, a method, called character segmentation, that can reduce the search space is introduced.

Character segmentation is a process by which the areas that are thought to contain characters are extracted from the license plate. The license plate recognition methods are then applied only inside the segmented regions speeding up the search.

However, character segmentation is not such a trivial problem. Although it is assumed that only one character will be included in each segmented area, this is not usually the case. Some of the regions are joined together including more than one character, Fig. 2.

Fig. 2. Failed character segmentation resulting in some regions having more than one characters enclosed. This usually leads to recognition errors.

For recognition methods (for example template matching) that rely on the number of characters and the size of the extracted region to set their parameters, character segmentation must be accurate otherwise the results of recognition will not be acceptable.

In this work, the character segmentation is performed using image smoothing and histogram threshold methods. After careful observation of the license plates, it was found out that the plates appear in one of two formats (sec. 2). That is, the license plate is either white and the characters in it black or a black plate with white characters. Therefore, since only two main colors are involved, the histogram threshold method, though simple, is capable of producing good segmentation results.

However, image smoothening using median and mean filters must first be applied before the threshold method can be used. This is because the black and white colors on the license plate contain a lot of noise. The median filter reduces the noise and the mean filter smoothens the image. The two filter's kernel is 3x3 pixels in size. After the two filtering processes, the image is ready for the histogram threshold method.

The histogram of the license plate is plotted and the center of the two picks selected to be the threshold to be used for segmentation. This separates the plate into two regions, usually an outside region surrounding a smaller inside one. However, the characters are still enclosed in one of the two regions and must be extracted individually. Since the characters and numerals in a license plate are of the same size, each region can then be segmented into the individual characters by dividing it into equal regions based on the average size of the characters.

4.0 CHARACTER RECOGNITION

Neural networks are chosen as the main classifiers in this work because of their excellent results when dealing with data of many dimensions like the one here. The template matching method is only used to supplement the neural network when its output is below a set threshold. There are 24 alphabets and 10 numerals that must be recognized. There are 24 alphabets in use because O and I are not included, maybe because they resemble the numerals 1 and 0.

4.1 Neural Network

The neural network must be designed to classify 34 different characters (24 alphabets and 10 numerals). While it is possible to design such a neural network, it would take a long time to train and it is would also be slow during testing because of its large size. Small size neural networks are preferred because of their high learning and usage speeds. Therefore, to achieve such a neural network to learn the data, the “divide and conquer” method is applied where the characters are separated into two main groups, alphabets and numerals. These two groups are further subdivided into smaller ones. The resulting small groups can then be classified using several neural networks.

These subdivisions are based on the shape of the characters (our visual observations). For example, making sure that alphabet B and D are in different neural networks because of their similarities.

Finally, this results in the construction and training of five independent neural networks, three for the alphabets and two for the numerals. The five neural networks are:

1. Alpabet_1 to learn the letters A, B, C, E, G, H, U and P
2. Alpabet_2 for the letters D, F, J, M, Q, K, S and V
3. Alpabet_3 to learn the letters L, N, R, T, W, X, Y, and Z
4. Numeral_1 to learn the numerals 1, 9, 3, 4 and 8
5. Numeral_2 to learn the numerals 0, 5, 7, 6 and 2.

The neural networks are selected to be 3 layered trained using the back propagation algorithm. The size of the training sample is 15x45pixels. Therefore, the number of units in the input layer is 675. The output layer has 8 units for the alphabets and 5 for the numerals. 500 samples of each character are used to train the neural network.

The error back propagation method [6] is used to train the neural network. The system is trained to produce an output of 0.95 for the node representing the character or numeral being learnt and 0.05 for all the other output nodes. To further reduce the size of these neural networks, improving the training and test speeds, structural learning with knowledge [7] is used together with the error back propagation method.

4.2 Template Matching

The templates used in this work for each of the characters are constructed from the same data used to train the neural network. Therefore the initial size of the template is 15x45pixels. Each template is the average of 200 images selected at random from the 500 images extracted for neural network training. Note that the height and the width of the template are fixed.

This general template represents, as much as possible, the diversity of the characters in the license plates in the dataset. There are two templates to represent the black and the white characters (the size of the two templates is however the same).

Since template matching using a fixed size template is unlikely to produce good results, genetic algorithms are use to guide the search by automatically adjusting the size and orientation of the test sample to match those of the template.

4.3 Guide Genetic Algorithm Structure

For the neural network and the template matching recognition methods discussed above, the size of the training samples and the template respectively, is fixed (15x45pixels). During testing, these two methods will only produce good results if the size and orientation of the target character is about the same as those of a sample.

However, the target characters are of different size and orientation depending on several factors including the distance of the vehicle from the camera and orientation of the vehicle especially when turning. To solve this problem, a genetic algorithm is introduced to guide the search by selecting the position, size and orientation of the sample before testing. The samples extracted using the genetic algorithm can then be normalized by rescaling and rotating them to the normal size and orientation and then tested on the neural network or using the template matching method.

The design of the genetic algorithm chromosome that must be optimized is then of great importance. The chromosomes represent the parameters of the target region including its dimensions, orientation, and position. The chromosome is made up of the scaling rates m_{lx} and m_{ly} for the length and width of the target area, rotation angle, *ang* and the directional translations (t_x and t_y).

The parameters in the chromosome are coded as follows. The image size used is 320x240 pixels. This size can be taken as largest size of any license plate to be encountered in this system. However, the position of the camera during the image acquisition dictates that the maximum length of the license plate can only be about 25% of the image size, that is 80 pixels. Consequently, the maximum size of the characters is about 60% of the width height (48 pixels).

This is then the largest parameter that must be coded. In binary coding, this will require 6 bits. If a uniform 6 bits is used for all the other parameters, the rotation angle is limited to the range $-32 \le ang \le +32$, if the first bit is used to represent the direction of rotation.

This information is used to determine the length of the genetic algorithm chromosome. The genetic algorithm determines the position, orientation and size of each sample. The scaling rates, the angle and the translation each require 6 bits. Therefore, the genetic algorithm chromosome length is 30. This genetic algorithm is binary coded to allow for bit manipulation during training.

Although the structure of the chromosome in the genetic algorithm is the same for the neural network and the template matching methods, during testing their genetic algorithm fitness's are different.

For the neural network method the genetic algorithm's fitness is the neural network's output for the sample extracted using the present genetic algorithm's parameters.

On the other hand, the template matching involves the comparison of the color in the template and the target image. Each pixel in the template is compared with the corresponding pixel on the target image. The fitness function is the sum of the color difference, Eq. 1,

$$Fitness = \sum_{k=0}^{n} DC_{ij} \qquad (1)$$

Where DC = Pt_{ij} -. Pi_{ij} is the color difference. Pt_{ij} is the color value of the template image and Pi_{ij} is the color value of the target image. i and j represent the position of the pixel on the template image.

The selection method used is the roulette wheel. The initial population is set to 20, the crossover point is point 19 and the probability of mutation is 0.01.

During reproduction, we save the elite solution, and then use the top 40% of the fittest individuals to reproduce 80% of the next population. The remaining 20% of the next population is reproduced by selection of one of the parents from the top 40% group and the other from the remainder of the population. This method improves the search space by ensuring that we not only retain the best individuals for reproduction but also explore the rest of the population for other possible candidates.

The genetic algorithm process is terminated after 100 generation and the elite solution is taken as the final license plate location.

5.0 RESULTS

Computer simulations were carried out using the dataset described in sec. 2. For the purpose of calculating the accuracy of this method, the total number of license plates in the database was counted and the value used as this method's target. This total is 5456 license plates.

The computer simulations in this work were carried out using a Dell Optiplex SX260 Pentium 4 personal computer.

For each character region, the neural network and template matching method were initially individually tested. To use the two methods in a hybrid system, the neural network method was initially used followed by the template matching method for only the samples for which the neural network output was below 0.5.

The results of these computer simulations are shown in table 1.

Note that the results shown in table 1 are not for individual character recognition but for the license plate recognition. Therefore, to classify a license plate correctly then all the characters and numerals in it must be correctly recognized.

Table 1. Results of computer simulations.

Method	Percentage Accuracy	Average time (Milli-seconds)
NN only	**96.2**	**6**
TM only	**95.3**	**5**
Hybrid	**97.3**	**7**

5.0 CONCLUSION

In this paper, a license plate recognition method using neural networks and template matching guided by a genetic algorithm was proposed. Computer simulation produced a detection accuracy of 97.3%. Most of the errors were caused by inaccurate character separation resulting in two or more characters being taken as one.

In the future, a better character segmentation algorithm that can reduce the number of connected character regions must be found. To further improve on the speed of this system, ways to reduce the length of the chromosome, and hence increase the speed of convergence of the genetic algorithm, must be investigated.

REFERENCES

[1]. Ching-Tang Hsieh, Yu-Shan Juan, Kuo-Ming Hung. "Multiple License Plate Detection for Complex Background," aina, vol. 02, no. 2, pp.389-392, 19th 2005.

[2]. Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen. "Morphology-Based License Plate Detection from Complex Scenes," icpr, vol. 03, no. 3, p. 30176, 16 2002.

[3]. Kamat V., Ganesan S. An efficient implementation of the Hough transform for detecting vehicle license plates using DSP'S. Proceedings of Real-Time Technology and Applications Symposium IEEE Comput. Soc. Press. 1995, pp.58-9.

[4]. Yanamura Y., Goto M., Nishiyama D, Soga M, Nakatani H, Saji H. Extraction and tracking of the license plate using Hough transform and voted block matching. IEEE IV2003 Intelligent Vehicles Symposium. Proceedings. IEEE. 2003, pp.243-6.

[5]. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition. CVPR 2001 vol.1, pp.511-518. 2001.

[6] Kah-Kay Sung: Learning and example selection for object and pattern recognition, PhD Thesis, MIT AI Lab, 1996.

[7] M. Ishikawa: Structure learning with forgetting, neural networks journal, Vol. 9, No. 3, pp 509-521, 1993.