Artificial Neural Networks in the Automatic License Plate Recognition.

Ascencio López José Ignacio, Ramírez Martínez José María
Facultad de Ciencias
Universidad Autónoma de Baja California
Km. 103 Carretera Tijuana, Ensenada, B.C.
MÉXICO

Abstract: One of the basic goals of computer sciences is the creation of tools capable of interact with the objects surrounding them, taking decisions autonomously, based on the occurrence of certain events, using artificial senses, like vision. In this paper, the authors present the character recognition problematic, digital image processing stages involved in it, and the most appropriate techniques to carry out each of these stages, as well as the results obtained.

Key-Words: automatic recognition, artificial vision, neural networks, image processing, backpropagation.

1. Introduction

One of the basic objectives of the computer sciences is the creation of tools able to obtain a behavior similar to that of the humans, interacting with the objects in their surroundings and taking decisions based on the occurrence of certain events, with the minimal intervention of an intermediary, giving the machine artificial senses, like vision. Artificial vision uses digital image processing techniques to make a computer capable to distinguish different shapes.

The most general stages involved in digital image processing are acquisition, preprocessing, segmentation, description and recognition [1]. In this paper some results of the techniques that were the most appropriated to carry out the digital image-processing task (visual patterns recognition, license plates, in particular) are shown.

The description stage, where a proprietary scheme is used, is specially emphasized.
Finally, the recognition stage, where a set of artificial neural networks trained by backpropagation accomplishes the task, is presented.

2. Problem formulation

Automatic license plate recognition is a problem that when solved successfully it would have a wide range of applications; some of them are vehicular databases, access control, monitoring, integration with intelligent agents, etc. The objective is to recognize automatically the string that represents automobile license plate, and once recognized, it can be used as a key for a database query in the application.

2.1 Related work

In 1993, Eric W. Brown, elaborated an optical character recognition system, applying a backpropagation neural network. The network was trained with three sets of characters, each set was a different font, containing 84 characters. Each character was represented in 8x8 matrices. The reference indicates that there were difficulties in training process (a slow convergence), using standard backpropagation. However, it also points out that once trained, the network performed its task properly, giving fast responses, unlike a system also developed by the author, where he used a traditional technique (pattern matching) for the recognition task.
The network learned necessary information in order to carry out its task during the implementation stage. Although less

accurate than its counterpart system (54% of correctly recognized characters, compared to a 72% achieved by the traditional technique system).

A digit recognition system was designed and implemented at Massachusetts Institute of Technology in 1999. The system recognizes withdrawal amount written in Brazilian checks, using as recognition module an array of four multilayered neural networks trained by backpropagation. The input patterns for two of the networks were obtained using a description scheme, and the inputs for the other two were the information of the 16x16 bitmaps containing the digits.

Objective of the network array is to increase recognition rate, the first two networks corrects the other two, and vice versa. Networks were trained using a set of 3103 digits, and were tested with another 1444 digits set. The percentage of successful recognition was 82.7%.

3. Recognition process

Recognition process starts up from a bank containing one hundred images, captured with a digital camera that produces 24 bit True Color, Tiff formatted imaging, with dimensions 320 pixels wide by 200 pixels high taken at a fixed distance (reducing scale invariant to minimum).

3.1 Segmentation.

In this case in particular, the segmentation stage was carried out in two steps. The goal of this stage is to obtain the characters printed on the license plate of the vehicle on the image. First of all, license plate region is located in order to obtain these characters. Image with the license plate is located using color information through several variable thresholds. Possibility of finding regions same color as the license plate implies that we can obtain more than one region during the process, therefore, a second property was used: the license plate area, so the system can distinguish regions that corresponds to other kinds of objects, that match color with the license plate.

Let R(x,y), G(x,y), B(x,y), represent the RGB components of the pixel located at the x,y position. To consider pixel x,y as part of the license plate, the next condition (that the pixel is of a specific color) must be true

$$\{(r_1 < R(x,y) < r_2) \wedge (g_1 < G(x,y) < g_2) \wedge (b_1 < B(x,y) < b_2):\ r_1, r_2, g_1, g_2, b_1, b_2 \in [0,255]\} \quad (1)$$

In our case, Baja California's *border* plates are yellow, with green-colored characters, therefore the condition to consider a pixel as yellow is

$$(128 < R(x,y) \leq 255) \wedge (128 < G(x,y) \leq 255) \wedge (0 < B(x,y) \leq 128) \quad (2)$$

and with the condition

$$(0 < R(x,y) \leq 128) \wedge (128 < G(x,y) \leq 255) \wedge (0 < B(x,y) \leq 128) \quad (3)$$

a pixel will be considered as green.

Our first segmentation condition will then be

$$\{(128 < R(x,y) \leq 255) \wedge (128 < G(x,y) \leq 255) \wedge (0 < B(x,y) \leq 128)\} \vee \{(0 < R(x,y) \leq 128) \wedge (128 < G(x,y) \leq 255) \wedge (0 < B(x,y) \leq 128)\} \quad (4)$$

In fig. 1 a binarized image obtained using several variable threshold technique is shown.

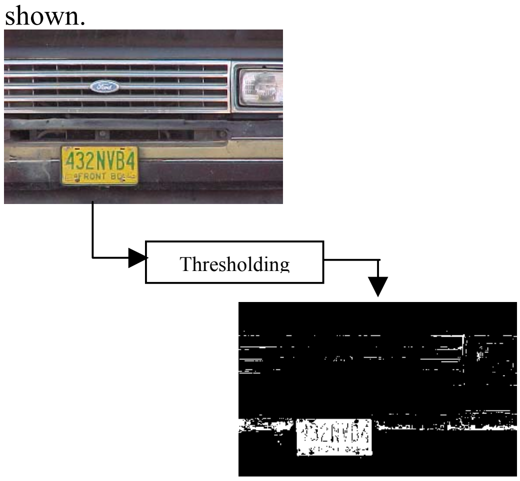

Fig. 1. Result of thresholding

Through the morphological opening, particles with radius less than 2 pixels are eliminated, and by closing operation, holes with radius less than 2 pixels are filled. In fig. 2 a result of these two operations are shown.

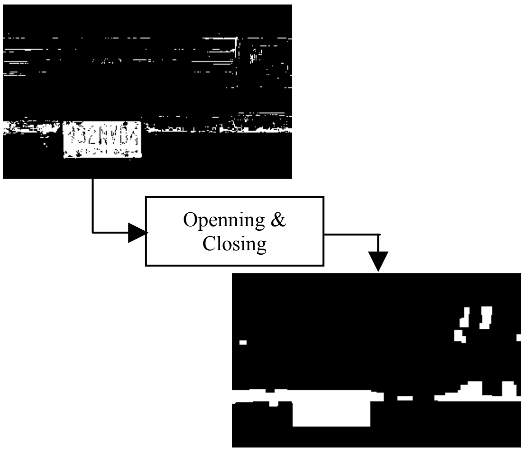

Fig. 2. Openning and Closing Results

Once morphological operations are carried out, it is possible to obtain more than one region, so it is necessary to eliminate those regions are not part of license plate. As mentioned, all of the captured images were taken at a fixed distance, so we know beforehand the area that the region we are looking for might cover. In order to determine if an area is going to be eliminated, its area must be within a certain range values. This way the search space is reduced, eliminating very small or very large regions. In fig. 3, the region elimination is illustrated.

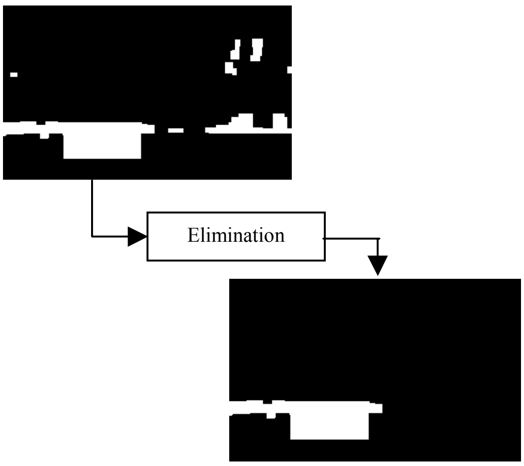

Fig. 3. Region elimination

Once we have reduced the number of regions, it is necessary to determine which one (if there are several) is actually the license plate. It is known that it has a rectangular shape, so the next step is to locate a rectangular zone. This was accomplished by pattern matching.

In fig. 4 we show the result of the first step of segmentation.

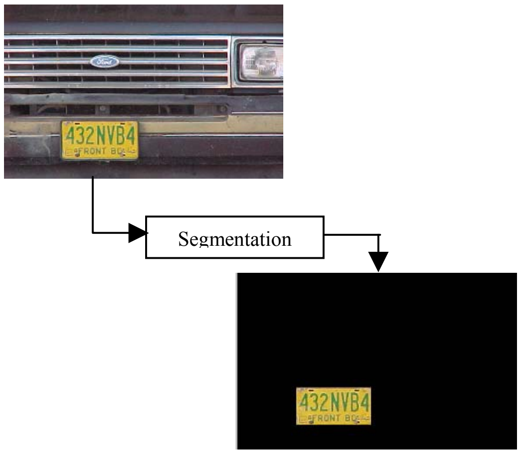

Fig. 4. Plate segmentation

Character segmentation was carried out in a similar way, by several variable thresholds. In this case the color license plate image was binarized considering only green pixels. In fig. 5 a binarized license plate is shown

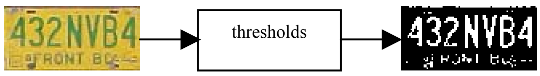

Fig. 5. Binarizing license plate

We also used important data crucial for the segmentation task (the data was a result of observations made to some characteristics of the vehicles): Baja California's Vehicle Department assigns a kind of license plate to compact vehicles, and other different kind to cargo vehicles. License plates corresponding to compact vehicles begin with a sequence of three digits, follow by set of three letters (where always the first of these three letters is the N), and the last character is a digit. The seven characters of a cargo vehicle license plate begin with a group of three letters (where the first is Z) and four digits.

In fig. 6 we show the format of each of these types of license plates.

DDD**N**CCD   **Z**CCDDDD

Fig. 6 Distribution of characters in compact and cargo vehicle license plates, respectively. D is digit, C is a letter, **N** and **Z** special cases of letters.

This information was very helpful, it made the character location process easier, once N or Z was located, we calculated the positions of all of the other characters.

In order to increase the accuracy of the location process, we used pattern matching, using a generic character, i.e. a character that is formed by the most characteristic elements (lines) of each character (digits and letters)

The matching was carried out locally, 5 pixels to the left of the location obtained, and 5 pixels to de right (displacing the pattern horizontally). In fig. 7 generic character is shown.

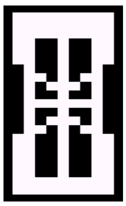

Fig 7. Generic character

The positions are used in the color license plate image to obtain the characters.

Fig. 8 Character segmentation

3.2 Description

Once we have segmented the characters, it is possible to send data directly to the recognition module. However, the amount of data is usually large, thus affecting the recognizer performance. In order to avoid this situation, it is recommended to use some features of the objects, instead of the pixels representing the objects, this way the information is compacted [3]. The description stage consists in finding and obtaining the most important features (also known as descriptors) of the objects to be recognized, they must meet the following requirements [3]:

- Discrimination
- Reliability.
- Independence.
- Small numbers.

There are different description schemes, in our case, two schemes were used; the first is a mesh-based scheme, in the second one, we use hole information (number of holes in a character, vertical position and area).

In the main description scheme, 3 cells wide by 7 cells high mesh is overlaid on the (previously binarized) character. Characters are stored in a 9 by 21 pixel matrix, thus each cell of the mesh covers a 3x3 pixel neighborhood. Once the mesh is placed on the character, number of white pixels inside each cell is counted; the sum is divided by 9 (total of pixels), obtaining an average of the white pixels. The descriptor vector, containing 21 entries (one for each cell), stores these values. In fig. 9 an example of a binarized character with the mesh overlaid is shown.

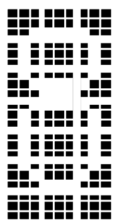

Fig 9 Character with mesh

The secondary scheme utilizes information about the holes a character may have (2 at the most). Holes are located first, and then their vertical positions (up, in the middle, down) are determined, and area the hole covers.

3.3 Recognition.

As in each of the other stages, there exist different alternatives to carry recognition task out. We used artificial neural networks.
A set of four multilayered artificial neural networks trained by backpropagation was implemented (three for letter recognition, and one for digit recognition). An additional set of four single-layer multiperceptrons for characters classification using the hole information was used.

It was decided to use two sets of neural networks in order to increase the recognition success rate, if the first set is not able to recognize one character, probably the other set might.
In case of letter recognition using the mesh based representation scheme, 3 networks where used.
One recognizes only the letter Z, the second one recognizes letters that are the most probably to appear after the first letter, and the third one classifies the rest of the letters. A fourth network was used for the recognition of digits.
All of these networks have 21 input units, corresponding to each value of the descriptor vector. Z recognizer has only input and output layers, with one output unit. The second letter recognizer has one hidden layer with 11 units, and 5 output units. The network for the recognition of the rest of the letters has also a hidden layer with 20 units, and 15 output units (the license plates don't contain all of the letters in the alphabet)
It was necessary to try with several networks, each one with a different architecture (varying the number of hidden layers, and number of units in each hidden layer, and changing learning rate during the training phase, and analyzing their performance), in order to select these networks. Each network was tested with a set of approximately 100 automatically segmented characters
The digit recognizer has one hidden layer with 18 units, and ten output units. The selection process was similar to that of the letter recognizers. In this case a set of 115 automatically segmented digits was used.

3.4 Operation mode of the neural networks during the recognition stage.

In order to achieve the license plate recognition, information about the digits and letters distribution was used. First, it is verified if the license plate corresponds to a cargo vehicle, i.e., we check if the first character is a Z, thus we send its pattern to the Z recognizer. If the character is actually a Z, we already know that the next character is also a letter, so we send it to the second letter recognizer. If the second letter is not one of the most likely to appear after the first one, we send it to the network dedicated to the recognition of the rest of the letters, the third character is send to the third network. The next four characters are sent to the digit recognizer.
If the Z recognizer shows that the pattern does not correspond to Z, it means that the license plate belongs to a compact vehicle, thus we send the first three characters to the digit recognizer, and the set of letters are recognized in the same way as in the first case.
If any of the networks of the first set is not able of recognize one character, which character is processed to obtain the hole information, is sent to the second set of neural networks.

3.5 Obtained Results

During the test stage of the system, a 93.3% of successful letter recognition was achieved and an 87% of correctly recognized digits. However, the percentage of successfully recognized license plates was 65%, due to bad character segmentation (of any of the 7 character string).

4. Conclusions

In the machine printed character recognition, artificial neural networks are a nice option, due to its ability to derive from complicated and/or imprecise data, besides their capability to detect and extract patterns that are difficult for other kinds of techniques.

The more information is used in the digital image processing, the more decision resources might be considered, in our case, license plate color and area information was crucial during segmentation.
It is important to use the information that is obtained from the problem domain. In our case, the information about the two types of license plates and that in each type there always appear Z and N was very helpful during the segmentation, and the knowledge about the distribution of letters and digits in each type of license plate was very helpful during recognition.
It is quite important to use a set of several neural networks in the recognition module, to increase success rate, due to the fact that each network will be devoted to classify different elements of information.

References:

[1] Gonzalez, Rafael C. and Richard E. Woods. 1992. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc. Estados Unidos.

[2] Castleman, Kenneth R. 1979. *Digital Image Processing*.
Prentice Hall, Englewood Cliffs, New Jersey.

[3] Brown, Eric W. 1998. *Optical Caracter Recognition*. Internet site.
URL:
http://www.ccs.neu.edu/home/feneric/charrecnn.html

[4] Sinha, Anshu. *An Improved Recognition Module for the Identification of Handwritten Digits*. MIT. Internet site. URL:http://scanner-group.mit.edu/htdocs/thesis/anshuthesis.htm. 1999.