DESIGNING A REAL TIME SYSTEM FOR CAR NUMBER DETECTION USING DISCRETE HOPFIELD NETWORK

A.BANERJEE[1], K.BASU[2] and A.KONAR[3]
COMPUTER VISION AND ROBOTICS LAB
ELECTRONICS AND TELECOMMUNICATION ENGG
JADAVPUR UNIVERSITY
[1]ayanbanerjee2006@yahoo.co.in, [2]kanadbasu2005@yahoo.co.in & [3]konaramit@yahoo.co.in

Abstract

The paper addresses a novel scheme for detection of car numbers from its rear end number plates. The work has extensive applications in automatic identification of cars, responsible for Cox pollution or road accidents. It involves a series of image processing steps followed by character recognition using discrete Hopfield Neural Network. Experiments with real number plates of cars have been undertaken to validate the proposed scheme to be presented shortly. Experiment results are encouraging in the sense that failure rates in identification of car numbers is negligible of the order of 10 in every 1000. Naturally, the robustness of the algorithm is beyond question. One more interesting feature of the technique employed in the present context is its insignificantly small computation time of the order of 2.5 seconds, justifying its amenability for real time implementation.

1. Introduction

The recognition of the number plate from the rear side of the car comprises basically of two steps. The first step is to scan the picture of the rear side of the car and to locate the position of the number plate hence segregating the number plate from the rest of the picture. The second step is to process the number plate as segregated in the first step and recognize the alphanumeric characters present in the number plate. So this work basically explores two basic areas that of image processing and that of character recognition. Similar works have already been conducted by various researchers in [1,2,3,4]. In this work, we have used basic image processing algorithms used to preprocess the picture before extraction of the number plate. For the final extraction of the number plate the paper proposes a scanning algorithm. The last and most important aspect in identifying car numbers is to recognize the individual characters and numbers on the number plate. Any classical algorithm of character recognition could have been used in the present context. However, in this work a non-classical neural network based algorithm for character recognition is selected. Ter Bruggee, et al [1] had used cellular neural networks for the said purpose. In this paper we have utilized discrete Hopfield Neural Network, which is used to identify the characters in the number plate. In this paper, we first state the proposed algorithm for extraction of the number plate from the picture of the rear side of the car and its subsequent processing for recognition of the characters. Then we present the simulation results for a particular car. The simulations are done on Matlab texture.

2. Algorithm

The basic steps used in the algorithm for car number detection are outlined here. First, colored photographs of the rear ends of the cars are taken by a movie camera placed at an emulated corner of a street in a working model. The acquired image is then eroded and dilated by using a square filter of 40 dimensions. This is well known as morphological opening [5], and it helps to remove the background from the image. Secondly, the extracted image is passed on through a two-line dilation filter of 5 dimension, enabling the number plate to segregate from the background. The subsequent operations on the image are then undertaken at gray level, and so we need to convert the colored image into gray image. After such conversion, the number plate looks brighter while the numerals and the characters look dark. A horizontal scan over the entire image is then performed to segment the number plate only from the image. This process is easier to implement contrary to the edge detection algorithm stated by Parkar, et al[2]. In the next step, the positions the individual characters from the number plate are extracted. To do this, the gray image of the number plate is first thinned using a Laplacian of Gaussian filter. The gray image after thinning has a black background with the characters written in white, they being approximately one pixel thick. Now, vertical scanning is performed on the binary image, searching for a stretch of white pixels. When this stretch exceeds a preset value (compatible with the dimension of the number plate), the top position of the characters of the number plate is obtained. We perform the

same operations from the bottom of the number plate and obtain the location of the lower part of the characters. Then the gap between consecutive characters is utilized to extract the individual characters. The individual characters are thus extracted from the gray image of the number plate.

The last and most important aspect in identifying car numbers is to recognize the individual characters and numbers on the number plate. Any classical algorithm of character recognition could have been used in the present context. We however have selected a non-classical Neural Network based algorithm for character recognition. Prior to employing the algorithm, we need to thin the characters using a LOG (Laplacian of Gaussian) filter. The thinning is performed to reduce the width of a character to approximately one pixel. The thinned image is then converted to a binary image by thresholding and a discrete Hopfield Network is used to recognize the individual characters and numbers on the number plate.

3. Simulation Results

We have performed simulations on about 1500 cars with a good success rate. The failure rate is about 1 in 1000. Henceforth we discuss about a particular example, which we had simulated in practice. Illustrative pictures of only one of the cars are shown here.

Step–1: - Acquisition of Colored Photographs of the rear side of the cars. This particular example shows the rear side of a Maruti Omni car.

Fig1: - Rear side of a Maruti Omni

Step–2: -**Morphological opening of images using square filter**. The image is morphologically opened and the result is shown at the top of the next column.

Fig2: - Image after Morphological opening

Step–3: Dilation of the image obtained as output from Step--2: - 2 line filters, one horizontal and one vertical, of dimension 5 are employed for this purpose.

Step – 4: Conversion to Binary and Gray Image:- Image is converted to binary in order to locate the number plate in the picture and into gray image for extracting purpose. The image after filtering and conversion to binary image is shown below.

Fig3: - Binary image

Step 5: Scanning of the Binary Image to locate number plate: - We start by horizontal scanning of the binary image – locating white spots. Once a white spot is reached, we perform vertical scanning from that point to obtain the vertical stretch of white pixels. If the stretch exceeds a preset value, we continue a new horizontal scanning, now with the whole vertical stretch – and thereby notice the rectangular extent of the white pixels. If this stretch exceeds some pre-estimated value which is compatible with the dimensions of the number plate – the stretch is identified to be the position of the plate.

Step–6: **Extraction of the number plate**: - Once the position of the number plate is located, it is extracted from both the intensity and binary image. The binary image is

used for locating the position of individual characters in the number plate. The intensity image is used to extract them.

Fig4: - Image of the extracted number plate

Step–7: Location & Extraction of Characters of Number Plate: - Vertical scanning is performed on the binary image, searching for a stretch of white pixels. When this stretch exceeds a preset value (compatible with the dimension of the number plate), the top position of the characters of the number plate is obtained. We perform the same operations from the bottom of the number plate and obtain the location of the lower part of the characters.

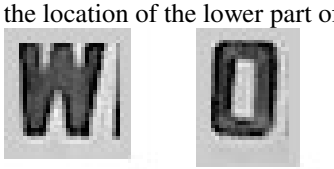

Fig5: - Image of extracted characters from number plate

STEP – 8: **Recognition of characters**:-Once the individual characters of the number plate are extracted thinning is performed on the characters using a 'log' filter (Laplacian of Gaussian). On performing thinning operation the characters approximately become one pixel thick. This thinned image is then converted to binary image. Now a discrete Hopfield network is used for the recognition of these characters.

4. Discrete Hopfield Network

The maximum size of a character is first found out. In this particular example the maximum size is less than 42 pixel X 32 pixel. These characters are then mapped into a 42 pixel X 32 pixel matrix. So, a total of 9 two dimensional matrices is obtained for each car (there are a maximum of 9 characters in the number plate of a car). These two dimensional matrices are then converted to row vectors. This is done by appending the second row of the matrices to the first row and then appending the third row to the second row and so on. So, 9 row matrices are obtained for each car each of which is 1 pixel X 1344 pixel in size.

Now a neural network is designed with 1344 neurons. This neural network is trained with each of the characters of the number plate. Whenever the nth character of the number plate of a car is fed as an input to the network the row vectors corresponding to the input is first subtracted from each of the row vectors corresponding to the nth character of the cars, which are used for training the network. If there is a perfect match between input pattern and training pattern the resultant row matrix will have all zeroes. In case of no match there will be nonzero elements in the resultant matrix. Now a search is made for finding those training patterns for which the number of nonzero terms is minimum. Then the input pattern is most associated with those patterns. A two dimensional matrix is formed for the nth character of the car by taking the row vectors corresponding to those training patterns as the rows of the two dimensional matrix. Let one such two dimensional matrix be denoted by S. For better understanding of the character recognition problem let us take an example. Figure 6 shows the first character of a Maruti Omni car, figure 7 shows the first character of an Ambassador, and figure8 shows the first character of a Tata Indica car. These characters are used to train the network. Now when the first character of another Ambassador fig 1, different from the first one, is fed as input to the network the minimum in the number of nonzero numbers is obtained for fig 1 and fig 3. Now taking the row vectors corresponding to fig 1 and fig 3 forms the S matrix. The weight matrix is formed according to the following formula.

$W = S^T * S - I$ where I is a unit matrix of dimension 1344 by 1344.

fig 6 fig 7 fig 8

Stable point

fig 9

With this weight matrix a stable point of figure 1 is reached for figure 9 as input.

A portion of the weight matrix for the recognition of the first character is shown below.

w =

2	2	2	0	-2	0	2	0	2	2	2	2	2	2
0	-2	2	2	0	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	-2	-2	0	2	0	2	2	2	2	2	2
0	-2	0	2	0	2	2	2	2	2	2	2	2	2

Another example for the 5th character of the number plate of the cars is shown in the next page.

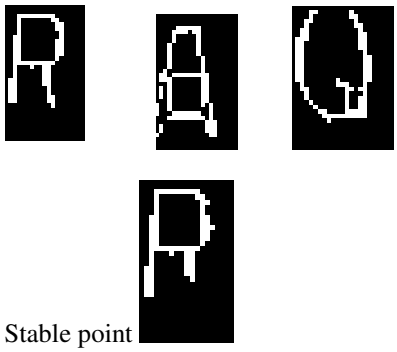

Fig.10. Analysis of stable points.

Conclusion

Our method has proved to be effective in the sense that error rates are very low. Use of Discrete Hopfield Network has guaranteed a better security against noise over the cellular neural network used by [1]. This is because discrete hopfield nets are inherently stable to noise and perturbations. The use of scanning algorithm in order to locate the number plate also proved to be much more efficient than the edge detection algorithm used by [2]. We have been able to separate adjacent characters whose spacing is as less as one-thirtieth of the character widths. However when two characters are joined together then the extracted image of the characters also contain those two characters joined together. Our method has been a font-independent method, since the training set comprised of different fonts of characters. However, this method only works with static images of cars. It will be necessary to work with car dynamic images, so that the system can be readily implemented for public services. We hope to work on this field in the future. We are also trying to find out the response of the character recognition algorithm when the characters are heavily deformed and to obtain an approximate allowable distortion limit for this algorithm to work well. We are also trying to implement this algorithm in cases where the images are blurred by dust particles being emitted by car tires when they accelerate.

References

[1] "License plate recognition using DTCNNs"- ter Brugge M.H., et al at the proceedings of Fifth IEEE International Workshop on Cellular Neural Networks and their Applications.14th-17th April,1998 .Pgs-212- 217.

[2] "An approach to license plate Recognition"- J R Parkar, et al.

[3] "Automatic License plate Recognition" Shyang-:Lih Chang, et al, IEEE Transactions on Intelligent Transport System.

[4] "Detection and Recognition of License plate characters with different appearances"- Shen-Zheng-Wang, et al. 2003 IEEE Transactions on Intelligent Transport System.

[5] "Digital Image Processing", Rafael C. Gonzalez, Richard E. Woods, 2nd Edition, Pearson Education, pg-550.