Secure License Plate Recognition System Using Associative Memory (Neural Networks)

Arun Kulkarni
Associate Professor
Thadomal Shahani Engineering College,Mumbai
kkkarun@yahoo.com

Ekta Shah
Thadomal Shahani Engineering College,Mumbai
ektabshah1994@gmail.com

Harsh Shah
Thadomal Shahani Engineering College,Mumbai
harshshah949494@gmail.com

KEYWORDS: License Plate Recognition, ANPR, Neural Networks, Associative Memory.

ABSTRACT: Automatic recognition of license plates has become very important in our daily life because of the extravagant increase in the number of vehicles. It becomes very difficult to be managed and monitored by human beings. This issue is solved by using Artificial Neural Networks. It can be used for systems like traffic monitoring, managing parking tolls. Our system tries to extract the license plate characters from live images taken through webcams or high resolution cameras or from offline license plate images and then recognizes and identifies the user and his details that are securely encrypted and stored in the database for discrimination between legal/registered user and illegal user. The approach towards the solution is to first conduct preprocessing of the image captured, to enhance the image to a desired level, the next step is plate localization. The third step of character separation and feature extraction is achieved using morphological erosion, filling holes and using the Bounding Box algorithm. The extracted features of characters are used the train the network using Hebb's Algorithm (Auto-Associative memory). We also verify whether the recognized car color matches with the registered color. The testing results obtained have an efficiency of 80%.

1. INTRODUCTION

The fast secure and efficient recognition of license plates is important for many applications like traffic monitoring, electronic toll collections, as authentication tool for tracking stolen cars. The traditional systems used techniques like template matching [1], [2], nodal point coding [3] which are slow to respond and thus cannot be used for many real-time applications. The LPR systems have become more reliable and wide spread due to the improvements in the software and hardware. Due to the plate variation from place to place, ANPR technology tends to be more region-specific.

The objective of this paper is to successfully locate the standard Indian number plates using Artificial Neural Networks. An automated LPR system is developed here using MATLAB in which images are captured live using camera or pre-captured image which are then sent for computational analysis where pre-processing techniques like median filtering, noise removal resizing is applied. Plate is localized using morphological operations and characters are segmented using Bounding Box algorithm. The extracted characters which are stored as vectors, are then used to train the Auto-Associative memory (Hebb network) and license plate number is successfully extracted which is then used for verification purpose. The color component of the car is also used as a second step in verification to enhance security.

The paper is formulated as follows: Section 2 presents the literature survey of license plate extraction Section 3 presents the proposed methodology for license plate extraction and verification. Section 4 shows the result. Section 5 draws conclusion.

2. REVIEW OF LITERATURE

We have chosen the following technical papers as a reference and foundation for our project.

Manisha Rathore and Saroj Kumari [1] used dilation, noise filtering, horizontal and vertical processing for preprocessing, character segmentation and Template matching algorithm for character recognition.

Ragini Bhat and Bijender Mehandi [2] used Sobel edge detection, Bounding-Box algorithm for character seperartion and template matching for character recognition.

Chih-Hai Fan and Yu-Hang Peng [3] used noise filtering, Sobel edge detection, horizontal followed by vertical projection for character segmentation and nodal points coding for character recognition.

Amr Badr, Mohamed M. Abdelwahab, Ahmed M. Thabet, and Ahmed M. Abdelsadek [4] have described a technique used for number plate recognition of Egyptian number plates. They have used a Feed-forward Artificial Neural network trained using back-propagation theorem (EBPTA) using chain codes as character features.

Prof. Primož Potočnik, Žiga Zadnik [5] have used hand-written character recognition algorithm, which is used to train the neural networks for efficient classification using character database as input. It is a helpful technique in the training phase.

3. SYSTEM DESIGN

3.1 Training Phase

The input character set image consists of characters(A-Z) and numerals (0-9) of different (five)fonts, which is used to train the network. The image is preprocessed by converting the color image to a binary image. Next character location is performed using edge detection and morphological operations like dilation followed by character separation which is done by filling the holes to make the characters suitable for applying bounding box algorithm to separate the characters. It returns the smallest rectangle containing the

region, which is specified by a 1-by-Q*2 vector, where Q signifies the number of image dimensions. Since we use images of the characters which are resized (80*80) i.e. 6400 features of each character is used to train the neural network. (Figure 1,2 and 3)

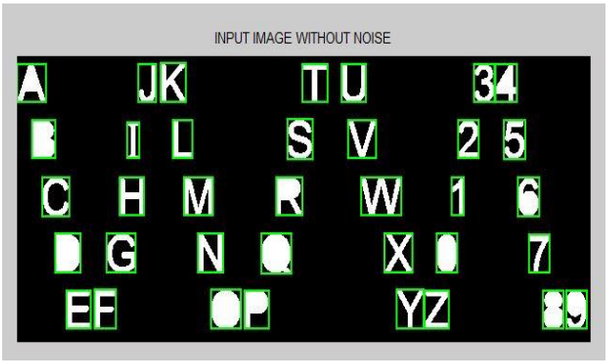


Figure 1 Training dataset

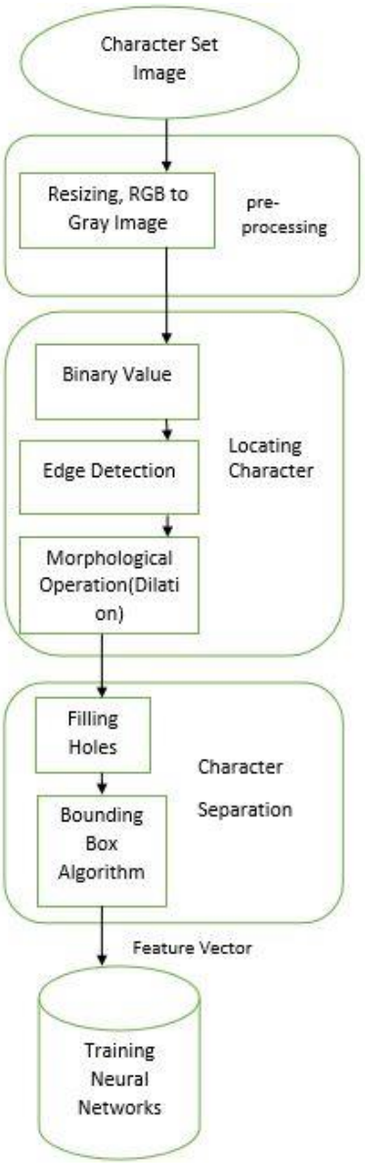


Figure 2 Training Diagram

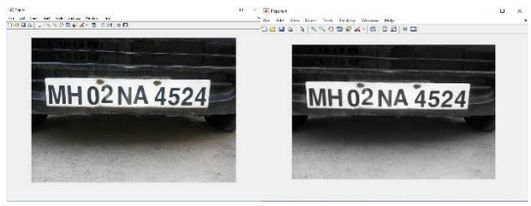
	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	1	1	1	1	1
5	0	0	1	1	1	1
6	0	0	1	1	1	0

…. (6400 features)

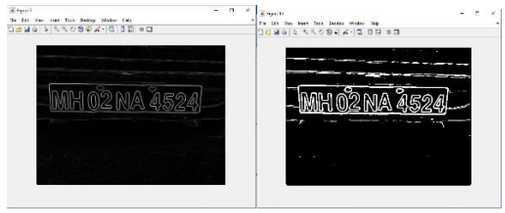
Figure 3 Character Vector

3.2 Testing Phase

The testing algorithm has been described below. The steps are as follows:



a) b)



c) d)

Figure 4.a) Image Acquisition b) Gray level slicing and Noise Filtering c) Morphological Operations d) Plate Location

1. The image is captured using webcams or other high resolution camera in Fig 4.a)
2. The image is resized and converted to a greyscale image. Noise filtering is done to enhance the image to a desired level in Fig 4.b)
3. Plate localization process is carried out with morphological operations like dilation, erosion and subtraction for edge detection in Fig 4.c)
4. Convolution is performed to enhance the edges and the plate image obtained is in Fig 4.d)
5. The image is converted to a binary image for character separation and holes are filled in Fig 6.a)
6. Character separation and removal of edges is done by morphological operations like erosion and subtraction. Filling holes is significant since character separation phase requires each character to be properly filled since Bounding Box algorithm uses connected regions concept to separate each character. So if there are holes within the character it might create two bounding boxes within a single character which leads to incorrect output vector in Fig 6.b)
7. The border objects are removed and Bounding Box algorithm is applied to separate the characters. Once the

characters are extracted, the feature vector is formed, like training phase, which is used for character recognition in Fig 6.c). Figure 5 shows the testing diagram.

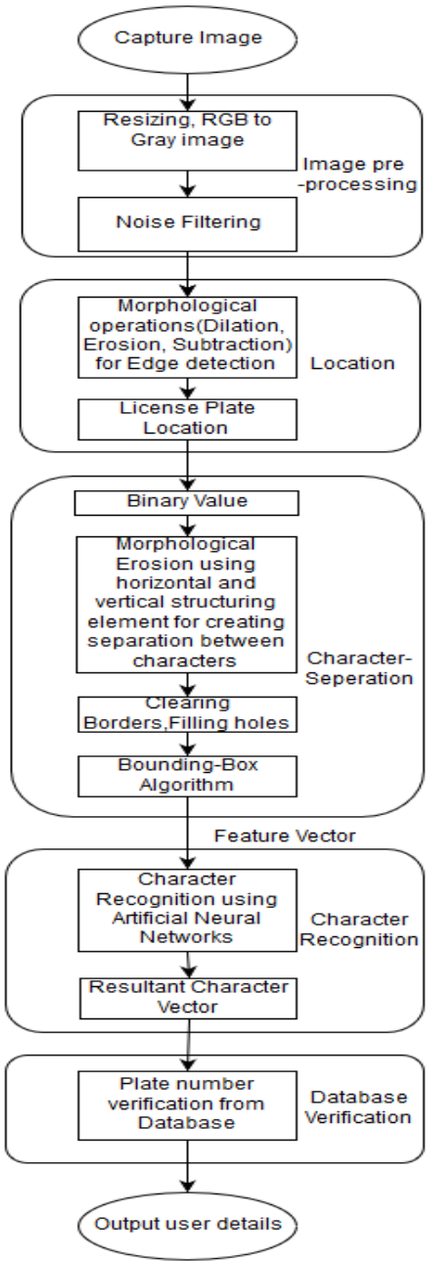
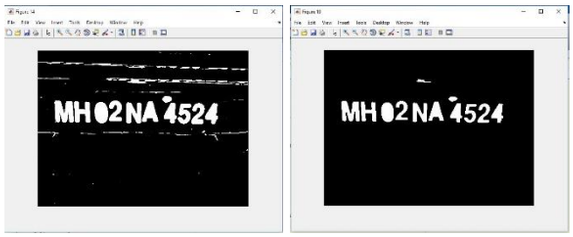
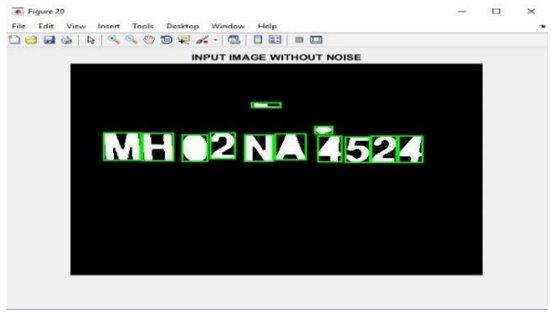


Figure 5 Testing Diagram



a) b)



c)

Figure 6.a) Filling of Images b) Removal of objects c) Bounding Box Algorithm

3.3 Character Training And Recognition Using Auto-Associative Memory

Pattern association involves associating a new pattern with a stored pattern. It is a "simplified" model of human memory. Auto-Associative Memory is used here for pattern association.

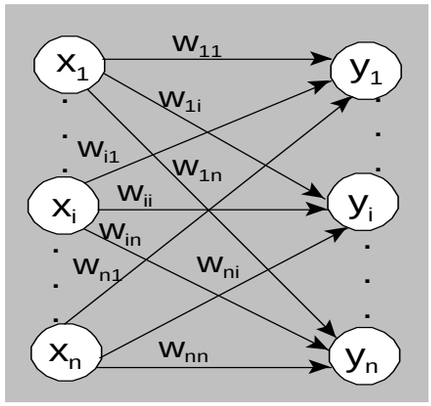


Figure 7 Auto Associative Memory Architecture

3.3.1 Training:

The inputs and output vectors s and t are the same. The Hebb rule is used as a learning algorithm or calculate the weight matrix by summing the outer products of each input-output pair. The input image of each character of size 80*80 is converted to a 1*6400 vector. These 6400 features of each character (A-Z and 0-9) forms the input pair (36*6400) to calculate the weight matrix. The input is normalized so that all values lie between [0,1].
The Hebb algorithm for training is as follows:

Initialize the weights to zero, wij =0, where i = 1,.. n and j = 1, …, n.

For each training case s:t repeat:

$x_i = s_i$, where i=1,...,n

$y_i = t_j$, where j = 1,…,n

Adjust weights $w_{ij}(new) = w_{ij}(old) + x_i y_j$, where i = 1, .., n and j = 1, .., n

Here the weight vector is of size 6400*6400. The weight vector is also normalized so that the values lie between [0,1].

3.4 Testing

The trained Auto-Associative memory is used to recognize the characters as below:

1. After training the weight vector is generated which is represented by w_i.
2. Each character is represented by vector x_i.
3. Determine the output vector by using the relation y_i by using the relation:

$$y_{mj} = \sum_{i=1}^{n} x_i w_{ij}$$

4. The output vector y_i obtained is not normalized. So the activation function is applied to normalize it as below:

$$y_{j=} f(y_{mj}) = \begin{cases} +1, & y_{mj} > 0 \\ 0, & y_{mj} \leq 0 \end{cases}$$

The obtained output vector (1*6400) is normalized in the range [0,1]. The minimum distance between this output vector(Yi) and the 36 rows of the weight matrix (W) is calculated and the corresponding row number gives us the recognized character.

3.5 Database Verification

A vector of row numbers is formed in the character recognition (testing) phase [13 8 27 29…]. Mapping of these numbers to characters is performed to obtain the recognized license plate (eg:MH02NA…). This string is then securely matched with the database in XML format to determine whether the car is registered and the history of car and user details are obtained. The corresponding color of the car is also verified and if the license plate and car color both match; user is granted access.

3.6 Tools used

The proposed algorithm discussed above is implemented using MATLAB R2016a on i5 processor, Windows 10,4GB RAM, mobile camera of 8 megapixel and webcam of 12 megapixel.

4. RESULTS

This system converts the complex tasks of parking management systems into a simplified environment. The license plates are captured through the camera which are then matched from the database. When the matched car is found, the details of the user is displayed.

4.1 Training:

The database has been trained using a variety of fonts. The training efficiency is around 85 percent with 30-31 out of the 36 characters for each font being detected successfully. The fonts used are Arial, Times New Roman, Sans-Serif and Serif. The characters like B, D, O, Q and Numerals like 6, 8 and 9 are being detected as zero for some fonts.

4.2 Testing:

Testing has yielded the following results where 16 plates have been correctly identified out of 20 plates i.e. 80% efficiency.

If number plate is recognized and user is granted access, his details are displayed on GUI as shown in Figure 8.

Figure 8 GUI Snapshot 1

If the number plates do no match, then a cross will be shown for no match as shown in Figure 9.

Figure 9 GUI Snapshot 2

The results obtained are displayed as follows:

Table 1 Result Matrix

Results obtained	Training Phase(No. of characters)	Testing Phase(No. of license plates used)
Training set used	180	20
Correctly classified	151	16
Accuracy	~85%	80%

Nodal point direction coding technique used for character recognition [3] gives an efficiency of around 85 % in testing phase but is largely dependent on image quality and fonts used since it recognizes pixel values of character and the number of endpoints in a character.

Template matching technique for character recognition [1], [2] gives 90 % efficiency during testing but can be used only for a font and is also affected by image quality and luminance. It does work well with different forms of license plates which we generally encounter in real-time. It is more useful for static images.

5. CONCLUSION

The objective of this paper is to develop a secure license plate recognition system**.** The algorithm used for character recognition is neural networks that basically tries to mimic the working of a human brain, it is always adaptable to learn with new datasets.

Some of the major challenges faced:

1. Edge detection and character recognition using morphology is a big challenge.
2. Character separation using Bounding Box poses some problems if the characters are not properly filled which can be improved using horizontal and vertical projection.
3. Number plates not adhering to the specifications.
4. Proper uniformity of characters and higher resolution should to be maintained.
5. Broken number plate cannot be properly detected.
6. Blurry images cause a problem in character extraction.

Efficiency can be improved by overcoming these challenges and using dynamic algorithms along with high resolution cameras with better shutter speed and better night vision can be used for making the system more robust.

6. ACKNOWLEDGEMENT

Our special thanks to the Department of Information Technology for their cooperation and unconditional support. As our teachers, they provided us with their useful insights and extended a helping hand whenever it was required. We would also like to thank our Institute which provided the platform and infrastructure without which this project would have been a distant reality.

7. REFERENCES

[1] Rathore, Manisha, and Saroj Kumari. "Tracking number plate from vehicle using Matlab." International Journal in Foundations of Computer Science & Technology (IJFCST) 4.3 (2014).

[2] Bhat, Ragini, and Bijinder Mehandia. "Recognition of Vehicle Number Plate Using Matlab." International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering 2 (2014).

[3] Fan, Chih-Hai, and Yu-Hang Peng. "Vehicle License Plate Recognition System Design." Chung Hua Journal of Science and Engineering 7.2 (2009): 47-52.

[4] Badr, Amr, et al. "Automatic number plate recognition system." Annals of the University of Craiova-Mathematics and Computer Science Series 38.1 (2011): 62-71.

[5] Žiga Zadnik, Handwritten character Recognition: Training a Simple NN for classification using MATLAB)