License Plate Recognition Using a Novel Fuzzy Multilayer Neural Network

Osslan Osiris Vergara Villegas, Daniel González Balderrama, Humberto de Jesús Ochoa Domínguez and Vianey Guadalupe Cruz Sánchez

***Abstract*—**In this paper we present a proposal to solve the problem of license plate recognition using a three layer fuzzy neural network. In the first stage the plate is detected inside the digital image using rectangular perimeter detection and the finding of a pattern by pattern matching, after that, the characters are extracted from the plate by means of horizontal and vertical projections. Finally, a fuzzy neural network is used to recognize the license plate. The tests were made in an uncontrolled environment in a parking lot and using Mexican and American plates. The results show that the system is robust as compare to those systems reported in the literature.

***Keywords*—** Plate detection, Character extraction and recognition, Adaptive threshold, Neural networks, Projection, Filtering, Multiresolution.

I. Introduction

Intelligent Transportation Systems (ITS) are having a wide impact in people's life as their scope is to improve transportation safety and mobility and to enhance productivity through the use of advanced technologies [1].

An special area inside the ITS are the Automatic Target Recognition (ATR) systems. The main goal of an ATR system is to detect, to classify and to recognize an object inside a scene. Inside this field, the automatic License Plate Recognition (LPR) systems are located.

LPR is required for the purposes of enforcement, border surveillance, vehicle thefts, automatic toll collection and perhaps traffic control. LPR can be applied to access control in housing areas, automatic parking control and marketing tools in large shopping complexes, and for surveillance [2], [3], [4], [5], [6].

Manuscript received November 30, 2008: Revised version received December 31, 2008.

Osslan Vergara is with the Universidad Autonoma de Ciudad Juarez, Chihuahua, Mexico; (overgara@uacj.mx).

Daniel González is a mechatronics student at Universidad Autonoma de Ciudad Juarez, Chihuahua, Mexico; (al67239@uacj.mx).

Humberto Ochoa is with the Universidad Autonoma de Ciudad Juarez, Chihuahua, Mexico; (hochoa@uacj.mx).

Vianey Cruz is a Ph. D. student at Centro Nacional de Investigacion y Desarrollo Tecnologico (cenidet), Cuernavaca, Morelos, Mexico; (vianey@cenidet.edu.mx).

In any automatic plate recognition system, two main different stages can be distinguished; first, a particular region within the input image has to be identified as a car plate (localization), and then a character sequence inside the region has to be validated as a correct plate string following some grammatical rules (character recognition) [7].

These stages are always applied on controlled environments, static backgrounds or controlled light scenes [8], [9]. Under uncontrolled conditions, the process of detection should face a variety of situations like locating plates at different distances, different plate orientations, light conditions and noisy plate images. By these conditions, some characters are hardly recognizable or distinguishable from others.

In this paper we present a novel system for automatic license plate recognition that can be used under uncontrolled environments.

The rest of the paper is organized as follows. Section 2 briefly describes the proposed method to detect a plate inside a scene. The process to extract the characters from the plate is described in Section 3. The design of the fuzzy neural network to recognize a plate is show in Section 4. Section 5 presents the tests and the results obtained of our approach in different situations. The conclusion and future works are presented in Section 6.

II. Car Plate Detection

As the input we have a digital image of a car; then, the plate is searched in the image.

Essentially, the methodology to detect the plate is composed of two methods: detection of a rectangle, corresponding to the perimeter of the image of the license plate and by comparison of the normalized correlation coefficient (match).

In the figure 1 we show a flow chart to explain the stages of the two different methods used to detect the plate inside the input image.

A. Rectangle detection

In order to detect a rectangle inside the car scene, we performed the following steps: 1) Minimum Filter, 2) Gaussian Pyramidal Filter, 3) Thresholding, 4) Morphological Operations (erode and dilate), 5) Canny Filter and 6) Union Chain Contour Sequence.

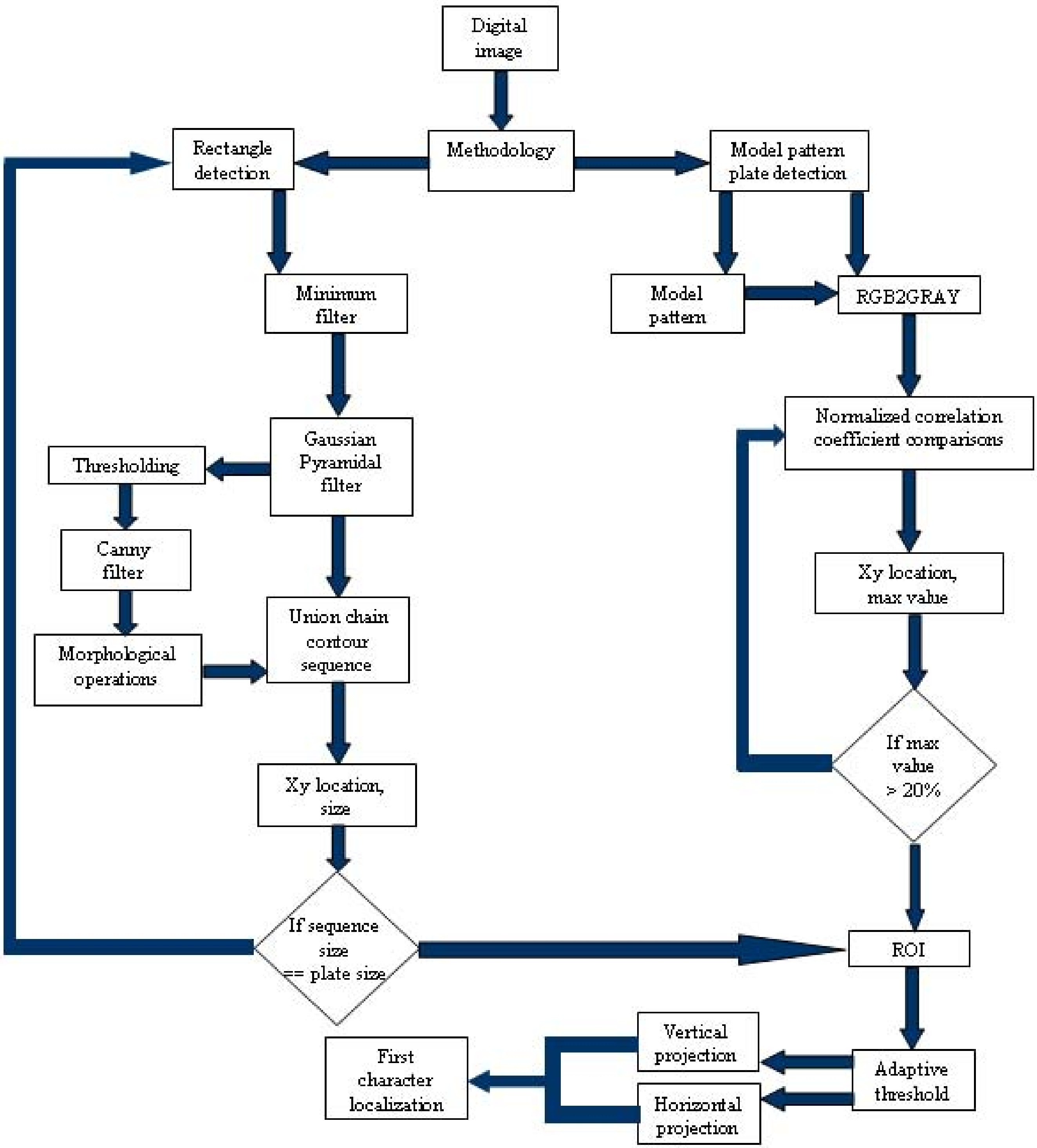


Fig. 1 flow chart to detect a plate inside the input image.

Minimum filter. We applied a minimum filter in order to assign to each pixel of the image, the minimum value extracted from all the values contained inside an *n x n* mask. The goal is to decrease the contrast of the input image. The Gaussian mask used is *A/16*, where *A = Gaussian Kernel*.

The weights pertaining to the mask have the shape of a Gauss bell described by means of equation 1, after this operation we convolve the input image using the equation 2.

$$f(x,y) = e^{\frac{-x^2+y^2}{s^2}} \tag{1}$$

$$g(x,y) = f(x,y) \otimes a(x,y) \tag{2}$$

The results obtained are showed on figure 2. Figure 2 a) is the original image and figure 2 b) is the resulting image, at this image the contrast of the plate is enhanced and the character inside the plate are prepared for the stage of recognition.

a)

b)

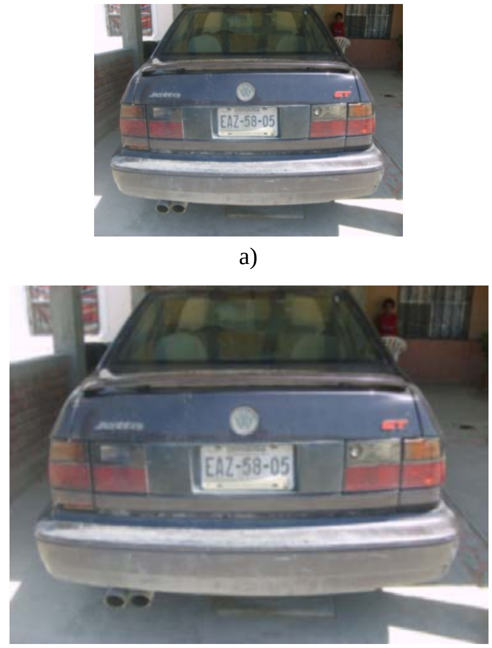
Fig. 2 filtering. a) original image and b) minimum filter results.

Gaussian Pyramidal Filter. This filter was applied to minimize the noise inside the image. Each level of the Gaussian pyramid is smoothed by a symmetric kernel and downsampled to obtain the next level of the pyramid as shown in equation 3.

$$P_{Gaussian}(I)^{j-1} = S \downarrow (G_\sigma P_{Gaussian}(I)) \tag{3}$$

The set of images obtained correspond to the multiresolution representation of an image [10].

The results obtained are shown in figure 3. The images were obtained by means of downsampling and upsampling operations on the input image.



a)

b)

Fig. 3 results obtained with Gaussian pyramid. a) downsampling and b) upsampling.

Thresholding. This operation is used to separate the background from the objects of the input image. This stage is very important for the success of the next phases.

Even with the great variety of thresholding methods proposed in the literature [11], [12], we propose an algorithm for adaptive thresholding. Equations 4 and 5 show the proposed algorithm.

$$v = \left[\frac{\sum_{k=1}^{m} \sum_{l=1}^{n} x_{kl}}{mn} \right] \mp \Psi \tag{4}$$

$$\Psi = \left[\frac{\sum_{k=1}^{m} \sum_{l=1}^{n} x_{kl}}{mn} \right] \left[\frac{\sum_{k=1}^{m} \sum_{l=1}^{n} x_{kl}}{e^{\alpha} mnx_{\max}} \right] \tag{5}$$

Where α < 1 and is calculated by image normalization. The comparison of the results obtained with Otsu method and our own method are shown in table1.

Table I results obtained with thresholding.

Original	***Otsu***	***Adaptive threshold***
		
		
		

Morphological operations. To reduce the noise introduced by the thresholding stage, we applied the well known morphological operations of dilate and erode. In figure 4 we present the results obtained after applying these operations.

Fig. 4 resulting image after dilate-erode.

Canny filter. Edge detection is one of the most commonly used operations in image analysis, and is a very important part of a process called segmentation [13].

We use the Canny filter in order to detect the edges inside the image, that was because the Canny filter offers a good union chain in edges in the plate area as it is shown in the car image of figure 5.

Union chain contour sequence. The union sequence is made by a stack. Stacks are known as Last In First Out (LIFO) data structures. We store in the stack the set of coordinates *xy* which defines the corners contained in the image in order to define rectangles.

We build inference rules in order to discard rectangles that can not be a plate. The results obtained are shown in figure 6.

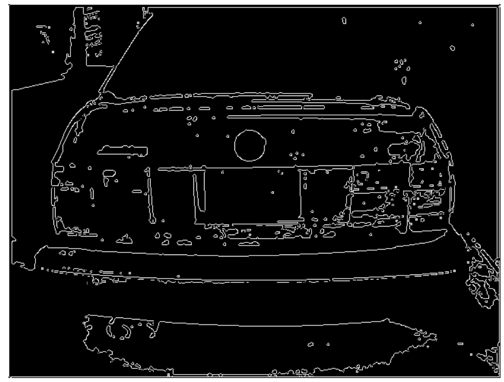


Fig. 5 image obtained with Canny filter.

Fig. 6 the rectangles detected by union chain.

B. Plate detection by model pattern comparison

After the first stage, we obtained several rectangles inside the image which can be interpreted as a plate. In order to ensure that a rectangle is the plate we perform the following stages: a) Definition of a Pattern Model, b) Application of the Normalized Correlation Coefficient, c) Definition of the Region of Interest and d) ROI thresholding.

Definition of a pattern model. We trained a pattern model to find the plate in the input image. The pattern is generated by a logic AND operation. For example, we used three plate images, then we apply the AND operation pixel by pixel.

The AND operation computes the conjunction of the operation *A AND B* and store the results in *C*. The results obtained are shown in figure 7 and in table 2 we present an example of how to compute this operation.

Fig. 7 template created to detect the plate.

Table II the AND operation between two images.

k-th bit of A_{ij}	***k-th bit of B_{ij}***	***k-th bit of C_{ij}***
0	0	0
0	1	0
1	0	0
1	1	1

Normalized correlation coefficient. *WxH* is the input image and a template size of *wxh.* Then the resulting image have $W - w + 1 \; x \; H - h + 1$ pixels. The value of a pixel at each location *xy* defines the similarity between the template and a rectangle inside an image with the upper left corner located at *xy* and the lower right corner located at $x + w - 1, y + h - 1$.

As a result of this process we obtain a point inside the image which is the similar place to the pattern model defined. The results obtained are shown in figure 8.

Fig. 8 results obtained with normalized correlation coefficient.

Definition of the region of interest. We selected two ROIs of an image. The first one is defined by the size of a rectangle; the second one is defined by the size of the pattern model. The ROI extraction is the action of subtract only the region or a part of an image which contain the necessary features to detect and identify the characters inside a plate. In figure 9 we show the ROIs detected.

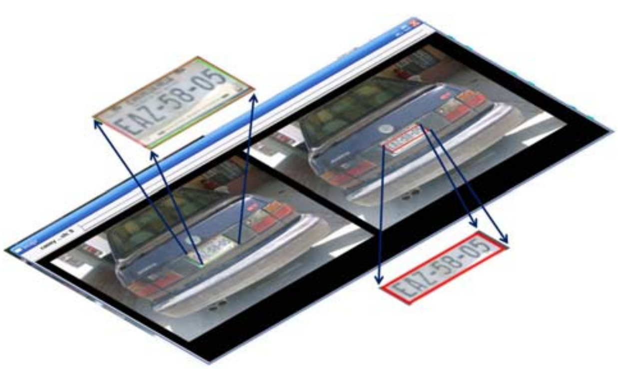


Fig. 9 the ROIs detected in the car image.

ROI thresholding. We need to perform the operation of thresholding inside the ROI detected which represent the plate. This operation is to detect the characters inside the ROI. In figure 10 we present the results obtained by thresholding the ROIs.

a) b)

Fig. 10 a) ROIs detected and b) the result of thresholding the ROIs detected.

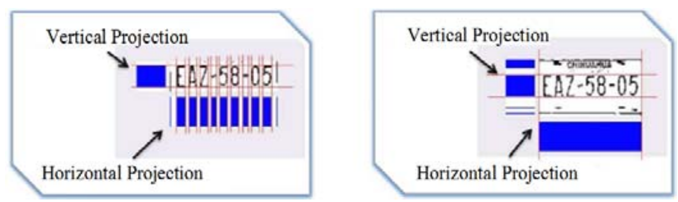
III. Character Extraction

To detect and separate the characters we used horizontal and vertical projections, and an operation to normalize the size of a character.

A. Horizontal and vertical projections

The operation of the projection is to compute the quantity of pixels pertaining to each row for the case of vertical projection, and the quantity of pixels contained at columns for the case of horizontal projection.

The minimum number of pixels in different vectors allows us to segment the characters contained in the plate. With this operation we obtain a square which enclose each character inside the image. The results obtained are shown in figure 11.



a) b)

Fig. 11 vertical and horizontal projections. a) from the ROI detected by match and b) from the ROI detected by rectangles.

The information obtained from the vertical and horizontal projections is used in a fuzzy set defined by equation 6 where Loc_{xy} define the position of the upper left corner of the first character contained in the plate.

$$Loc_{xy} = \begin{cases} x_1 y_1 & if \;\; x_1 y_{1n} \neq x_1 y_{1n-1} \\ x_2 y_2 & if \;\; x_2 y_{2n} \neq x_2 y_{2n-1} \\ \overline{x_1 y_1} & if \;\; x_1 y_{1n} = x_1 y_{1n-1} \\ \overline{x_2 y_2} & if \;\; x_2 y_{2n} = x_2 y_{2n-1} \\ \overline{x_1 y_1} \cap \overline{x_2 y_2} & otherwise \end{cases} \quad (6)$$

B. Normalizing the size of a character

At the final stage we are prepared to detect all the characters inside the plate. We need to apply a normalization process in order to obtain all the characters with the same size. We apply an operation to scale the characters inside the image; this is made by zoom in and zoom out the image. To do this an operation of downsampling and upsampling is executed. In the figure 12 we shown an example of this operation.

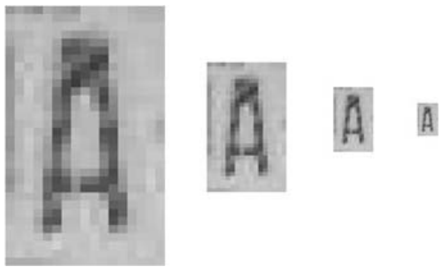


Fig. 12 the characters zoom in and zoom out by downsampling and upsampling.

IV. Plate Recognition

Once we have the coordinates of the characters contained inside the plate, we design a fuzzy neural network.

A. *Description of the fuzzy neural network*

The great difference between a connectionist machine and typical computer programs is that the first one processes the input information to obtain an output. For this work, we combine two classical neural networks: the perceptron and the adaptive linear neuron (Adaline) net.

The architecture of the perceptron neural network is very simple. We have a single layer structure containing a set of input cells and one or more output cells, according to the nature of the problem treated. The structure of a perceptron is shown in figure 13.

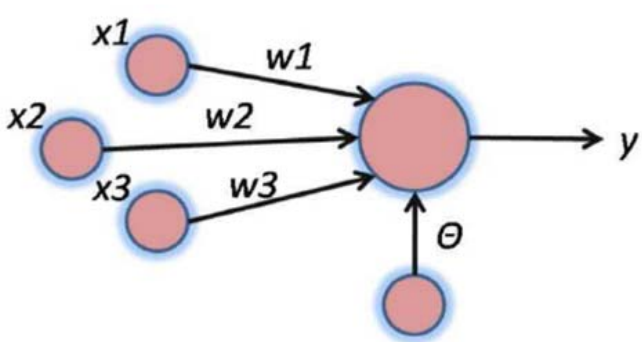


Fig. 13 the structure of a classic perceptron.

In order to compute the output *Y* of the perceptron neural network we use the equation 7.

$$y = \sum_{i=1}^{n} w_i x_i \tag{7}$$

Then, the first layer of the network created is a perceptron, here we have a set of matrices where the initial weights are contained.

The second layer, of the network created is again a perceptron, and then we have a multilayer network. In this layer the values are evaluated with a second stage of synaptic weights, as a result we obtain the closest value of the threshold.

The final layer is added in order to have a better discrimination mechanism. For this, we use the Adaline neural network to minimize the error using the real value produced by the difference of the output layer for a particular input pattern as it is shown in equation 8.

$$\tilde{y} = \sum_{i=1}^{n} w_i x_i + \phi \tag{8}$$

Each layer of the neural network designed is described as a sum which multiply the input by a synaptic weight for the stages of perceptron, and the difference of the values for the case of the Adaline layer as it is shown in equation 9.

$$y' = \sum_{i=1}^{n} w_i x_i \ AND \sum_{j=1}^{m} w_j x_j \ AND \sum_{k=1}^{s} w_k x_k + \phi \tag{9}$$

At the final of the last layer we obtain a fuzzy set which inference rules tries to minimize the error percentage and to offer the better result using the values obtained by the neural network. The topology of the final neural network is shown in figure 14.

B. *Methodology to fill the matrices of synaptic weights*

The methodology to fill the matrices is different from the classical methodology, where the weights must be random and small. We find the thresholds using the inputs and the matrix can pass the threshold without the training stage.

The input patterns are passed to the learning stage in order to compute the activation thresholds. The patterns are passed to each matrix by multiply the weights or by computing the minimum error for the case of Adaline layer.

First layer. To fill the first layer we use the input images which are called the group matrices. The matrix of synaptic weights is filled with the values of five characters using the OR logic function as it is shown in equation 10.

$$I_{wI} \sum_{i=1}^{n} I_{x1} \ OR \ I_{x2} \ OR \ I_{x3} \ OR \ I_{x4} \ OR \ I_{x5} \tag{10}$$

Where I_{wI} is the group matrix containing the sum of each I_{xn} and each I_{xn} is the input image as it is shown in figure 15.

We compute the values for each different activation threshold. After the creation of I_{wI} we pass each I_{xn} to compute each single threshold. We use a different threshold for each result obtained.

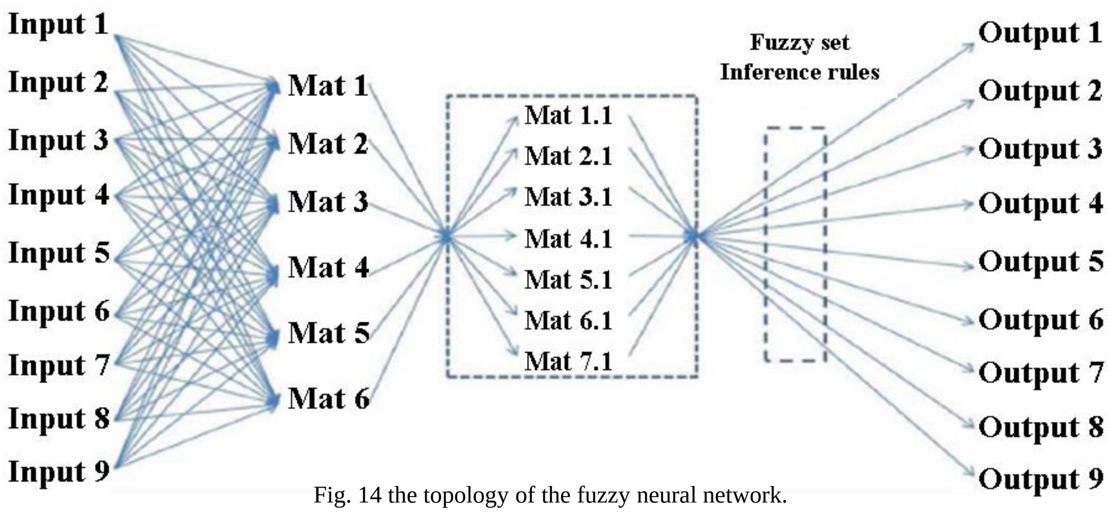


Fig. 14 the topology of the fuzzy neural network.

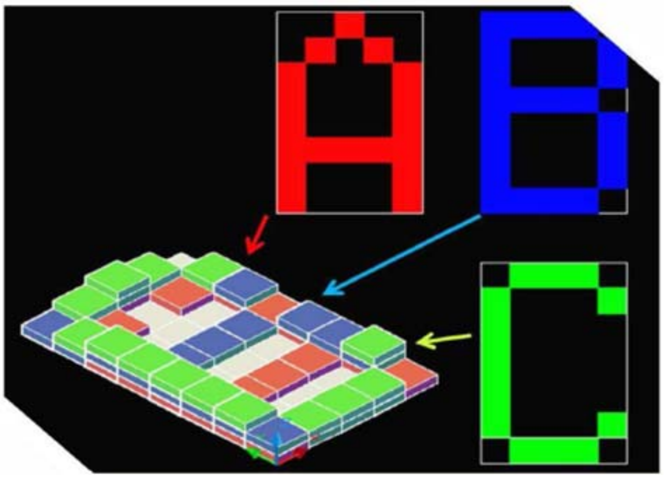


Fig. 15 structure of a group matrix, each square corresponds to a singular weight.

Second layer. To fill the matrix of the second layer we multiply it by a matrix defined for each character pertaining to the group matrix. We need to determine that each result is the closest to each threshold computed.

Each threshold is loaded in the time when the group matrix is created, here is necessary to consider the declaration of several inference rules to ensure that result obtained is the closest, even when a character can be similar to other characters pertaining to another group.

Third layer. We use a recursive methodology, when we pass trough the second layer an activation value is obtained. This value informs that the input have similarity with a specific character. At the third layer the trained pattern is called again, and the differences inside the Adaline network are computed.

If at the final of the second layer the input image has an activation value defining its similarity to the letter “B”, we call again the matrix of features for the “B” character. We analyze it using the fuzzy set to determine if the classification obtained is correct, this is valid even when we have two similar values.

C. Fuzzy sets

A set is defined as a collection of well defined elements. Always it is possible to determine when an element of the set is part or not of the particular set. The main decision is obviously to determine if the element is part of the set or not [14].

We use the methodology of inference fuzzy sets, whose structure is composed by several blocks as is shown in figure 16. Each value to be evaluated represent the grade of difference obtained by the third layer of the fuzzy neural network described.

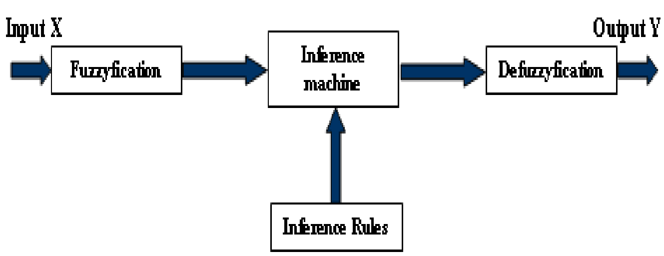


Fig. 16 diagram of the inference system.

The fuzzy block allow us to transform each normal value into a fuzzy value, the block of rules contain a set of linguistic rules used to synthesize the knowledge of an expert to solve the problem tackled.

The inference machine determines the grade of the value of each rule, at the final the output is transformed by the defuzzy block.

The inference rules inside the methodology proposed are created to insure that the character selected is really the most similar. In figure 17 we show the input of a random character and the path followed for each layer. At the stage of recognition we show several similar characters, and we can appreciate the patter models to detect.

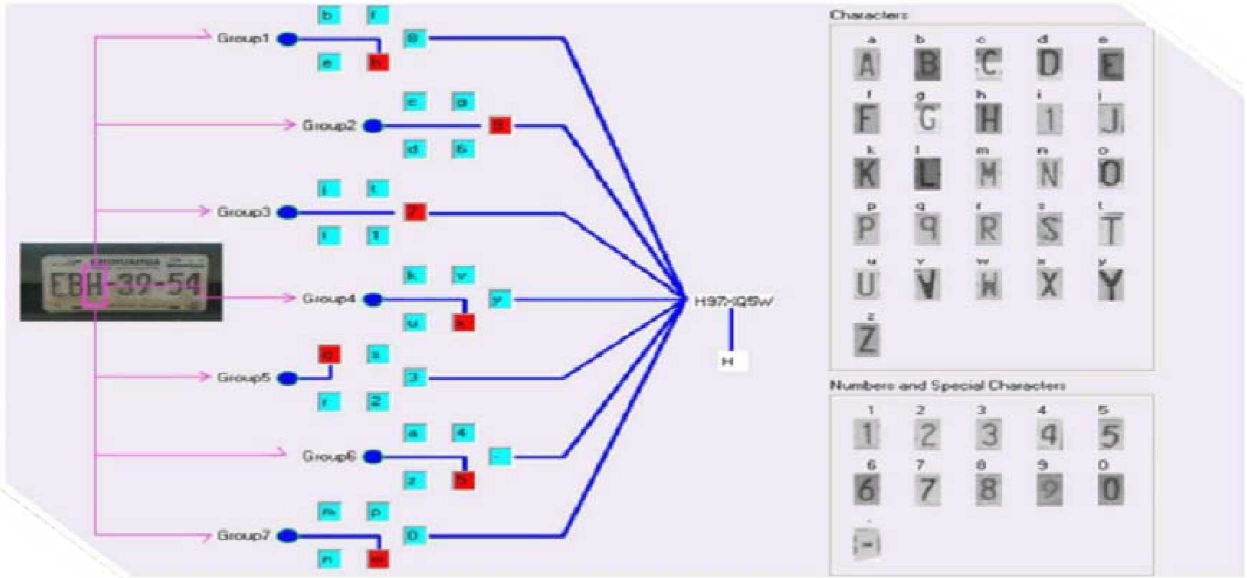


Fig. 17 the path followed for the character recognition.

V. Test and Results

The test images were taken on an uncontrolled environment at the parking lot of our University (Ciudad Juarez). We have three different kinds of plates: local plates of Ciudad Juarez defined by the gray color strip on the lower part, the plates of the border vehicles defined by a yellow strip on the lower part, and the plates of Texas (El Paso), characterized by larger characters.

As we mentioned before, the environment of the image acquisition was not controlled. In table 3 we presented some results for the case of plate detection. As one can see, the use of the two different stages is very important, because sometimes in one of them the plate is not detected. Therefore, this is a redundant process necessary to detect the plate.

One hundred percent of the plates of local and border vehicles were detected correctly; sometimes we have problems with the detection of Texas plates, because the size of the characters is different.

After plate detection, we carried out a test for character recognition; a key process of this stage is the process of adaptive thresholding. We have problems with noisy plates because we cannot separate the characters in a good way.

In table 4 we present a resume of the results obtained for the process of plate recognition. From Table 4, we can see that the main problem for the system to make a mistake was the dimension of the objects (characters) and the lighting.

We detect some cases of positive false that are those results whose detection of a vehicular plate where not correct even when the feature that we find are correct. The features are the size of a plate and that the rectangle detected must be similar at least for a percentage of 40% to be considered as a license plate.

This mistake occurs when we use gray scale images where the values are smaller than those of the RGB images.

The stage of image thresholding is the key to obtain valid characters. For the case of El Paso plates we do not have model patterns and this is the main problem to obtain the mistakes. In Table 5 we present the overall results obtained form the three stages.

As you can see sometimes we have problems when as input we have similar characters. This is due to the initial patterns were extracted from the original images.

The overall results are good compared with those reported in the literature, even with the uncontrolled environment.

VI. Conclusions and Further Works

In this paper we presented a novel methodology to solve the problem of car license plate recognition. The stage of plate detection is solved using two methods: rectangle detection and model pattern comparison. The stage of character detection is solved using an adaptive thresholding method and horizontal and vertical projections. The characters detected are resized in order to obtain always the same size of a character. Finally the problem of recognition was solved using a fuzzy three layer neural network.

The results obtained show that the system performs well even when the images were taken on uncontrolled environment and using three different kinds of plates.

In the future we are going to design an ideal character database to avoid the problem of character similarity. We are going to perform more tests to add better robustness to the system presented.

Table III results obtained for plate detection.

Source image	*Rectangle detection*	*Pattern model*	*Status*
			Accept
	Rectangle is not detected		Accept
			Accept
		Rectangle is not detected	Accept
		Rectangle is not detected	Accept
			Accept
	Rectangle is not detected		Reject
	Rectangle is not detected		Reject

Table IV results obtained for character recognition.

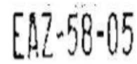
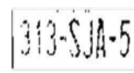
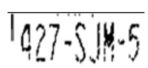
Source image	*Thresholding*	*Result*	*Status*
		Original: EAZ-58-05 Obtained: EAZ-58-05	Accept
		Original: 806-SKH-5 Obtained: 806-SKH-5	Accept
		Original: 313-SJA-5 Obtained: 313-SJA-5	Accept
		Original: 960-SFZ-5 Obtained: 960-SFZ-5	Accept
		Original: Y120 LPK Obtained: Unknown	Reject
		Original: EBH-39-54 Obtained: EBH-39-54	Accept
		Original: 427-SJM-5 Obtained: A27-5JM-5	Reject
		Original: EBD-36-99 Obtained: Unknown	Reject

Table V overall results obtained.

Number of images	*Plate detected*	*Failure*	*Acceptance percentage*
25	20	Plate size Light	80%
Number of Characters	*Non classified characters*	*Failure*	*Acceptance percentage*
180	23	Light	87.22 %

References

[1] Anagnostopoulos C. N., Anagnostopoulos I., Loumos V. and Kayafas E., "A License Plate Recognition Algorithm for Intelligent Transportation System Applications", *IEEE Transactions on Intelligent Transportations Systems*, vol. 7, no. 3, 2006, pp. 377-392.

[2] Sheikh Abdullah S. N. H., Khalid M., Yusof R. and Khairuddin O., "Comparison of Feature Extractors in License Plate Recognition", *Proceedings of the First International Conference on Modelling and Simulation (AMS)*, 2007, pp. 502-506.

[3] Tang Hsieh C. and Shan Juan Y., "Wavelet Transform Based License Plate Detection for Cluttered Scene", *WSEAS Transactions on Computers*, vol. 4, no. 1, January 2005, pp. 30-33.

[4] Karim Sulehria H., Zhang Y., Irfan D. and Karim Sulehria A., "Vehicle Number Plate Recognition Using Mathematical Morphology and Neural Networks", *WSEAS Transactions on Computers*, vol. 7, no. 6, June 2008, pp. 781-790.

[5] Kyung Kim K. and Ho Kim J., "Vehicle Plate Recognition System," *WSEAS Transactions on Signal Processing*, vol. 1, no.1, October 2005, pp. 119-124.

[6] Karim Sulehria H., Zhang Y., Irfan D., "Mathematical Morphology Methodology for Extraction of Vehicle Number Plates", *International Journal of computers*, vol. 1, no. 3, Summer, 2007 pp. 69-73.

[7] Lopez J. M., Gonzalez J. Galindo C. and Cabello J., "A Versatile Low-Cost Car Plate Recognition Systems", *Proceedings of the 9th International Symposium on Signal Processing and its Applications*, 2007, pp. 1-4.

[8] Draghici S., "A Neural Network Based Artificial Vision System for License Plate Recognition", *International Journal of Neural Systems*, vol. 8, no. 1, 1997, pp. 113-126.

[9] Kim K. K., Kim K. I., Kim J. B. and Kim H. J., "Learning-based Approach for License Plate Recognition", P*roceedings of the IEEE Signal Processing Society*, vol. 2, 2000, pp. 614-623.

[10] Mallat S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, 1989, pp. 674-693.

[11] Kittler J., Illingworth J. and Foglein J., "Threshold Selection Based on a Simple Image Statistic", *Computer Vision, Graphics, and Image Processing*, vol. 30, 1985, pp. 125–147.

[12] Otsu N., "A Threshold Selection Method from Gray-level Histograms", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, 1969, pp. 62-66.

[13] Gonzalez R. C. and Woods R. E., *Digital Image Processing*, Addison Wesley / Díaz de Santos, U. S. A., 2000.

[14] Dote Y., "Introduction to Fuzzy Logic", *21st International Conference on Industrial Electronics, Control and Instrumentation*, vol. 1, Orlando Florida, U.S.A, 1995, pp. 50-56.

Osslan Osiris Vergara Villegas was born in Cuernavaca Morelos, Mexico on July 3th, 1977. He received the B.S. degree in Computer Engineering from the "Instituto Tecnológico de Zacatepec (ITZ)," México in 2000, the M.Sc. in Computer Science at the "Center of Research and Technological Development (CENIDET)" in 2003, and the Ph. D. degree in computer Science from the National Center of Research and Technological Development (CENIDET), Morelos, México, in 2006. His research interests include pattern recognition, digital signal processing, computer vision, image compression and mechatronics.

He currently serves as a professor at the Autonomous University of Ciudad Juárez (UACJ), Chihuahua, México.

Dr. Vergara works for the Manufacturing an Engineering Department. He is member of the IEEE Computer Society. He is a member of the National Research Systems.

Daniel González Balderrama was born in Chihuahua, México on November 19th 1975. He received the B.S. degree in mechatronics from the "Universidad Autonoma de Ciudad Juarez", México in December 2008. His research interests include digital image processing, patter recognition, neural networks and mechatronics.

Humberto de Jesús Ochoa Domínguez received the B.S. degree in Industrial Engineering from The "Instituto Tecnológico de Veracruz", México, M.S. degree in Electronic Engineering from The "Instituto Tecnológico de Chihuahua", México, and Ph.D. degree in Electrical Engineering from "The University of Texas at Arlington", USA. In 1998 he was awarded the Chihuahua Prize for the "System to Classify Digital Mammograms into Normal and Abnormal using Texture Features and Microcalcifications Detection. He has conducted several tutorials/workshops on image/video coding/processing in Mexico, The United States, Italy, The Czech Republic and The Republic of Malta. He has published in refereed journals and has been a consultant to industry and academia.

He worked for the Mexican Merchant Marine as Electronic Officer and for the Nokia Research Center in Irving, Texas. He currently serves as a professor at the Autonomous University of Ciudad Juárez (UACJ), Chihuahua, México.

Dr. Ochoa works at the Electrical and Computer Engineering Department. He is member of the IEEE Computer Society and the IEEE Circuits and Systems Society. He is a member of the National Research Systems.

Vianey Guadalupe Cruz Sánchez was born in Cardenas Tabasco, México on September 14th, 1978. She received the B.S. degree in Computer Engineering from the "Instituto Tecnológico de Cerro Azul," México in 2000, the M.Sc. degree in Computer Science at the "Center of Research and Technological Development (CENIDET)" in 2004. She is a Ph. D. student of Computer Science at the National Center of Research and Technological Development (CENIDET), Morelos, México. Her research interests include hybrid systems, knowledge representation, neural networks, pattern recognition and digital image processing.

She currently serves as a professor at the Autonomous University of Ciudad Juarez (UACJ), Chihuahua, México. He works for the Electrical and Computer Engineering Department. Prof. Cruz is a member of the IEEE Computer Society.