*International Journal of Research in Engineering and Science (IJRES)*
*ISSN (Online): 2320-9364, ISSN (Print): 2320-9356*
*www.ijres.org Volume 3 Issue 12 ǁ December. 2015 ǁ PP.41-52*

# Identification system of characters in vehicular plates

S. Garcia[1], C. Aquino[2], D. Samorano[3]
[1]ESIME Culhuacan, National Polytechnic Institute.
[2]ESIME Culhuacan, National Polytechnic Institute.
[3]ESIME Culhuacan, National Polytechnic Institute.

**Abstract**: In this paper, is proposed a system that identifies the location and dimensions of the characters relative to the image of a vehicular plate, when the location of the plate has not been accurate. The system is divided into 4 stages, each with a specific purpose. Which are: binarization by thresholding, morphological filtering, identification of the largest area and segmentation similarity. The first stage is used to find at the extent if possible, the region occupied by the plate relative to the rest of the image. Then, in the filtering step, it seeks to eliminate as far as possible the noise that interferes with the identification of the characters. The third stage is used to identify the region occupied by the plate. Finally, in the fourth and last stage the segmentation by similarity is used to identify position and dimension of the characters in the image, in stage a Kohonen neural network is used.
**Keywords**: Kohonen, lqv, Otsu, morphological

## I. Introduction

Nowadays technology has grown favorably and has developed new solutions and alternatives for the improvement of society, although it is noteworthy that technology does not advance in the same way in all countries either by lack of financial resources, lack of education and training or lack of initiative and sometimes the lack of government support. In the case of Mexico optical character recognition has not been fully investigated which is a very broad topic and of most importance for alphanumeric reading of any text or vehicular plates. Some previous papers are cited in [1] - [2].

## II. Binarization

To emphasize at the possible extent the region that occupies the plate with the rest of the image the binarization technique was used. Binarization is an image processing technique that consists of the reduction of a digital image information with L + 1 levels of gray, where $0 \leq L \leq 255$, at two values: 0 (black) and 255 (white). The process is very simple, and involves comparing each pixel of the image with a determined threshold which is the limit value that determines whether a pixel would be black or white in color. The values of the image that are larger than the threshold take a value of the 255 for white and the other pixels take the value 0 for black.

### 2.1. Thresholding

In order to obtain the threshold, the Otsu's thresholding method was used to determine the optimum threshold $t_{opt}$, by maximizing a parameter that Otsu defined as the variance between classes, and is given by equation (1).

$$\sigma_B^2 = \omega_1(\mu_1 - \mu_T)^2 + \omega_2(\mu_2 - \mu_T)^2 \qquad (1)$$

where,

$\mu_1 = \sum_{k=1}^{t} \frac{k \cdot p_k}{\omega_1}$, is the mean of class 1

$\mu_2 = \sum_{k=1+1}^{L} \frac{k \cdot p_k}{\omega_2}$ , is the mean of class 2

$\mu_T = \sum_{k=1}^{L} k \cdot p_k$ , is the mean of the distribution (image)

The parameters $\omega_1$ y $\omega_2$ represent the sum of probabilities $p_k = f_k/N$, where $f_k$ is the frequency in which appears level of intensity k with $0 \leq k \leq L$ and N the number of pixels in the image, from 0 up to t and t + 1 to L respectively. That is,

$\omega_1 = \sum_{k=0}^{t} p_k$ y $\omega_2 = \sum_{k=t+1}^{L} p_k$

To determine the optimal threshold $t_{opt}$, an iterative process is followed, that starts at t = 1 and ends at t = L-1 since if t = 0 or L would only exist a class (class 1 or class 2).Thus, after performing the calculations of the variance between classes $\sigma_B^2$for each t, the optimal threshold is obtained $t_{opt}$ when$\sigma_B^2$reaches its maximum value.

In example, consider the digital image with 8 levels of grey shown in Figure 1.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 4 | 4 |
| 0 | 0 | 0 | 0 | 4 | 4 |
| 0 | 1 | 7 | 0 | 0 | 0 |
| 0 | 1 | 6 | 1 | 2 | 2 |
| 0 | 1 | 5 | 6 | 7 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |

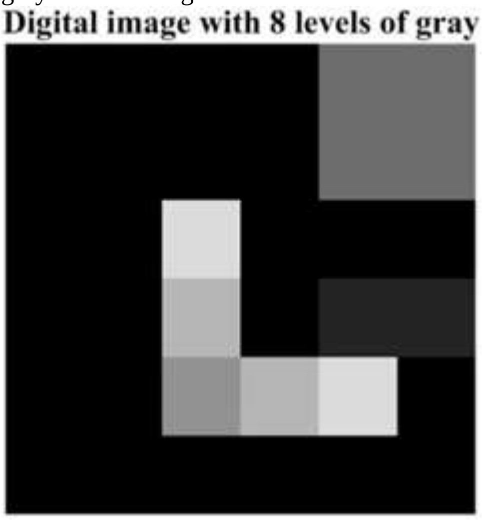


**Figure 1**. Digital image with 8 levels of gray (right). The matrix on the left represents the level of intensity of each pixel.

In table 1the values obtained from the variance between classes are shown when applying (1) for each candidate t.The maximum is obtained when t = 2. In this way, the pixels with intensity values greater than 2 are placed in Class 2 (intensity level 255) and the rest in class 1 (intensity level 1) as shown in Figure 2.

**Table 1**. Variance between classes for each candidate t value of the image in Figure 1.

| **t** | 1 | **2** | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $\boldsymbol{\sigma_B^2}$ | 4.95 | **5.33** | 3.71 | 5.14 | 3.06 | 1.75 |

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 255 | 255 |
| 0 | 0 | 0 | 0 | 255 | 255 |
| 0 | 0 | 255 | 0 | 0 | 0 |
| 0 | 0 | 255 | 0 | 0 | 0 |
| 0 | 0 | 255 | 255 | 255 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |


**Figure 2**. The binarized image (right) with t=2 of the image in figure 1. At the left the associated matrix is shown.

In Figure 3, the binarization of the digital image of a plate with 256 levels of grey is shown. In the process the Matlab **graythresh** in structions and **im2bw** were employed. The **graythresh** function determines the optimal threshold $t_{opt}$ and **im2bw** function performs the image binarization.


**Figure 3**. Gray scale image (left). Binarized image (right).

Sometimes it is not possible to differentiate the region occupied by the plate with the rest of the image. In Figure 4, it is observed how after the binarization, the region occupied by the plate is indistinguishable from the rest of the image, because the vehicle is white color.

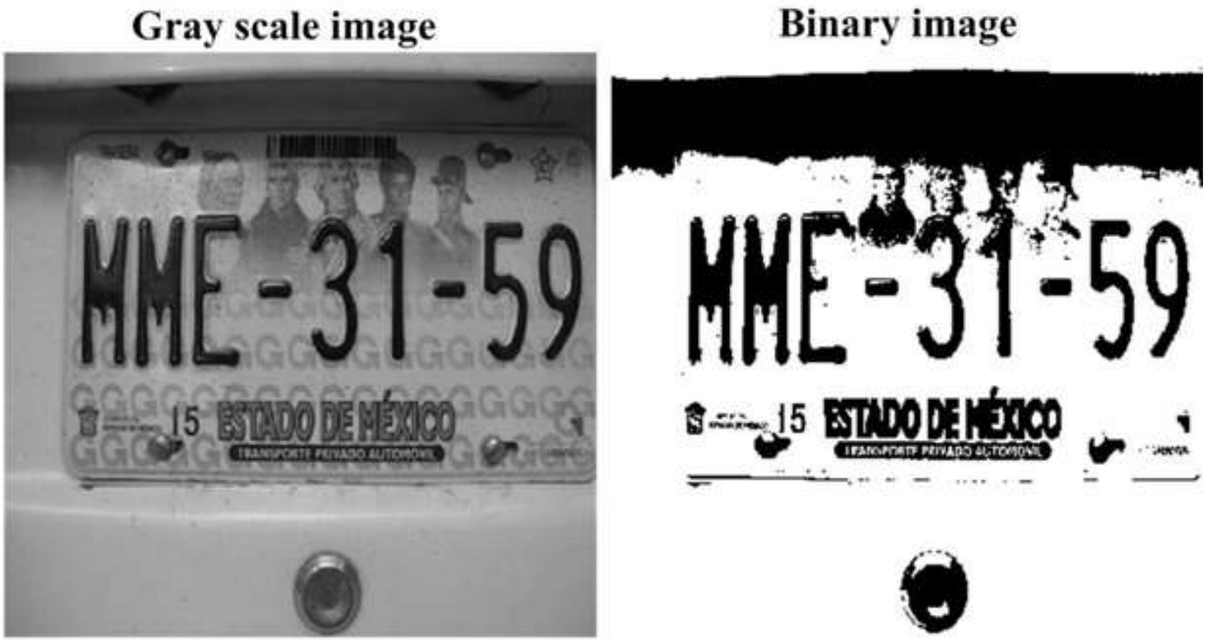


**Figure 4**. After binarization, it is not possible to differentiate the region occupied by the plate from the rest of the image.

## III. Morphological Filtering

Mathematical morphology is a widely used tool in image processing. The morphological operations may simplify the image data, preserving the essential characteristics and eliminating irrelevant aspects. In this work it is used after the binarization step, with the purpose to eliminate the weak connectivity dark areas within the plate and which overlap to the region occupied by the characters.

### 3.1 binary image

A binary image can be seen as a mapping set of $Z^2 \lozenge Z_2$, where $Z^2=Z\times Z$ with Z the integer set and $Z_2=\{0,1\}$. Consider the binary matrix in Figure 5 that represents a binary image. In the figure is stated the origin of the coordinates as well as the positive directions of the axes $m,n\in Z$. Each position (m, n) is mapped to a value of the set $Z_2$. Call this function X (m, n) the binary image that associates each position (pixel) a 0 or a 1. For example X (1,1) = 0.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Figure 5**. Binary matrix that represents a binary image.

### 3.2. Estructural element

The structural element B is a binary matrix with much smaller dimension than the image, which defines the shape and size of the pixel's neighborhood that would be analyzed. The transposed of the structural element B, denoted by B, is defined as the reflection on the origin of each item of B, ie, $\widehat{B} = -B$. In figure 6 a rectangular structural element and its transposed are shown, the origin of the coordinates is marked with an underscore.

| | |
|---|---|
| $\underline{1}$ | 1 |

| | |
|---|---|
| 1 | $\underline{1}$ |

**Figure 6**. Structural element B with rectangular form (left) and it's transposed$\widehat{B}$ (right).

### 3.3. Dilation and erosion

Dilation and erosion are the two basic morphological operations. Dilation of a binary image X by the structural element B, can be described as follows:

Step 1. A pixel (m, n) of X is selected to be analyzed.
Step 2. The origin of$\widehat{B}$, is positioned at (m,n).
Step 3. The value of the pixel (m, n) is changed to the maximum value of the pixel's neighborhood defined by $\widehat{B}$.

On the other hand, the erosion of a binary image X by the structural element B is described as follows:

Step 1. A pixel (m, n) of X is selected to be analyzed.
Step 2. The origin of the transposed of B,is positioned, that is, B in (m,n).
Step 3. The value of the pixel (m, n) is changed to the minimum value of the pixel's neighborhood defined by B.

In Figures 7 and 8 is shown the effect of a dilation followed by an erosion over the binary matrix of Figure 5 by the structural element of Figure 6.

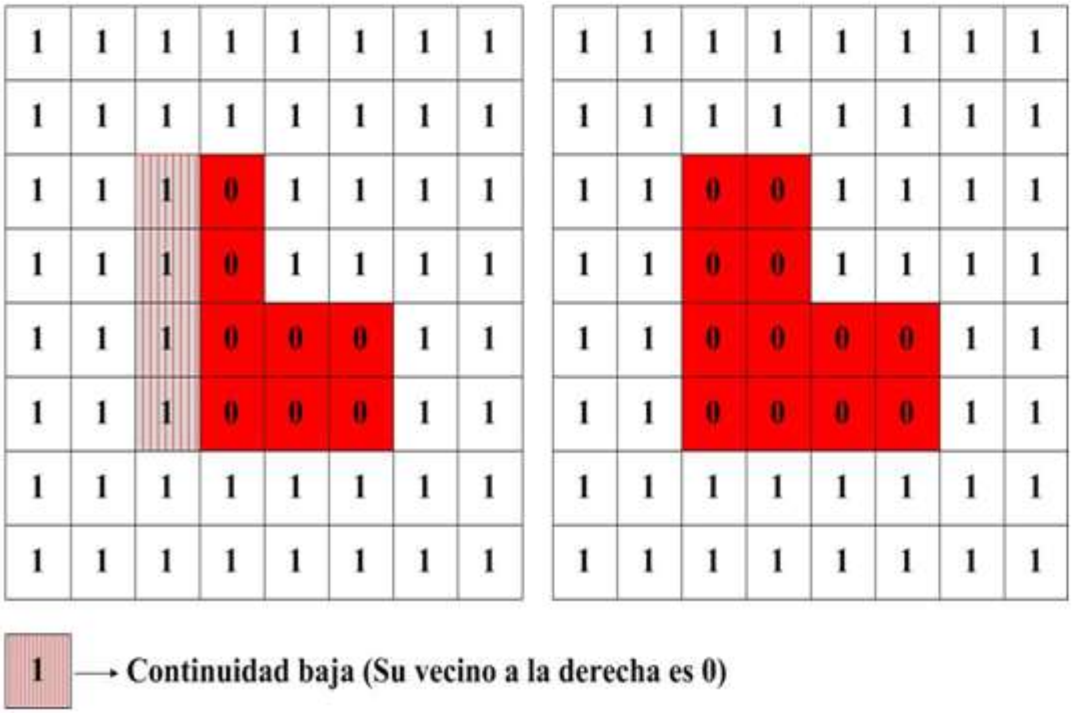


**Figure 7**. Dilation operation. Input matrix (left), output matrix (right).

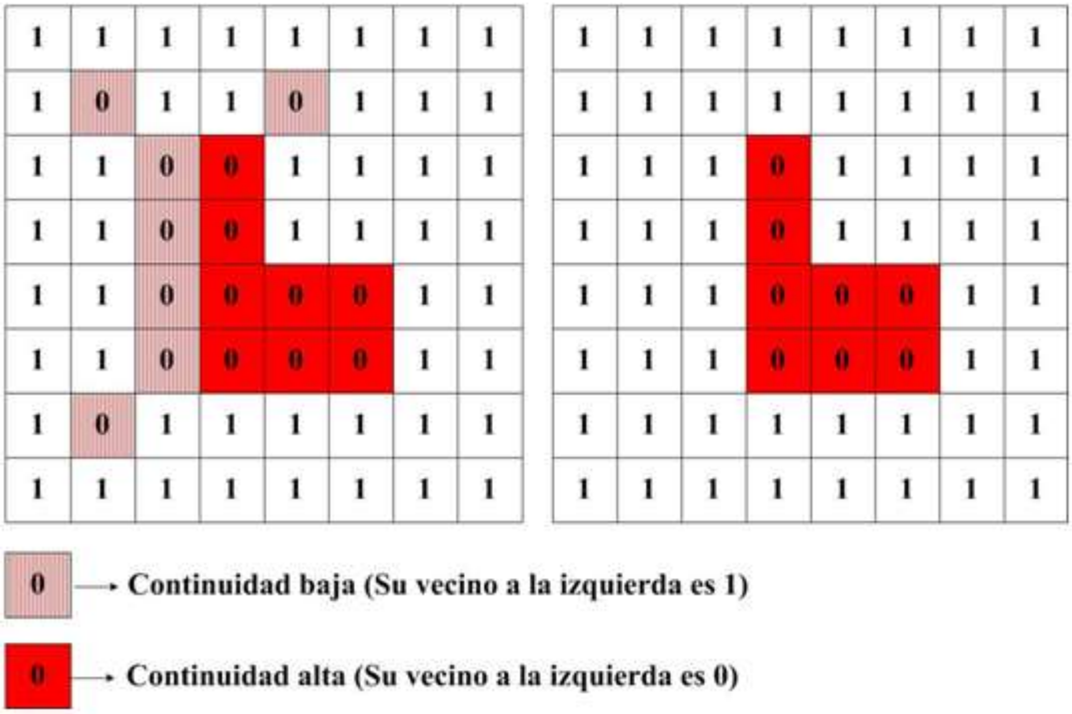


**Figure 8**. Erosion operation. Input matrix (left), output matrix (right).

Is observed how the effect of the dilation is to disconnect 0's regions with low continuity. Then, with the erosion, 0's region strongly connected and that suffered a slight disconnect with dilation operation is reconnected and returns to its original state. The combination of these operations and at this order is called **closing** operation. Ideally, this is what you would expect| to happen with the plate characters. However, in real conditions it is expected that most of the dark regions that overlaps and interfere with the plate would disappear, so as to allow proper identification of the characters in the following process stages.

A square structural element of size 5 was used in this study, so that the effect of dilation wouldn't be so agresive as to disconnect the characters. To implement the closing operation, the Matlab's **imclose** instruction was used. In Figure 9, two cases are shown where the operation gives good results and a third case in which the result is not so satisfactory, due that the interference region is strongly connected. An alternative to this case would be to decrease the binarization threshold before filtering.


**Figure 9.** Filtering over plates containing regions that interfere with characters

## IV. Identification of the larger area

In stage 3 of the work, an identification of "the larger area" within the image is made, for the purpose of getting closer to the region occupied by the plate. For this, the Matlab's function **regionprops** was used. This function allows identifying connected components within the image, to later obtain basic properties of each component such as: the area, perimeter, centroid, just to name a few.

When the **regionprops** function is directly applied to a binary image, implicitly uses the function**bwconncomp**, this function identifies connected components and that generates a data structure that provides among other things, the number of components and the location of the corresponding pixels to each component.

Then, a very general description of a procedure is given that determines the connected components, that is not necessarily the one followed by the **bwconncomp** function.

STEP 1. A pixel P with 1 as a value is selected and stored in the LP list.

STEP 2. A neighborhood of rectangular shape centered in P is generated, by default square shaped of dimension 3. In this case it is said that the window is of connectivity 8 (number of neighbors of P).

STEP 3. If there are neighbors of P with value of 1, these belong to the same component and are stored then in the LP list.

In figure 10, the implementation of the **bwconncomp** function is shown, followed by the **labelmatrix** function to a binary matrix, the **labelmatrix** function assigns a different label to each of the lists previously obtained by the **bwconncomp** function. At the time of implementing a default 8 connectivity is used.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 2 | 2 | 2 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 2 | 0 |
| 0 | 0 | 1 | 0 | 2 | 2 | 2 | 0 |

**Figure 10**. Identification of connected components. The original matrix (left) and the matrix with labels (right)

To identify the "larger area" within the image, the maximum of the "**Areas**" was obtained of the connected components obtained by applying the **regionprops** function and using the "**Area**" property, that as such does not correspond to the geometric definition. This property provides the number of pixels with value 1 of the component.

After identifying the "larger area", is proceeded to identify the rectangle of least dimension that contains that area. For this, the property BouningBox of the function regionprops was used, which gives the position of the upper left corner (origin) as well as the length and width of the rectangle of least dimension that contains that connected component. For example, the component 2 of figure 10 has it origin at position (1,5), length and width of 3.

Under ideall conditions, one would expect that the component of greater area would be the plate, because it is a connected component with most of its pixels with value of 1. However, unwanted scenarios may occur, for example, that the plate be a connected subcomponent (the plate doesn’t have a frame and the car is light in color) or that connected component exists with a larger area ( the plate having a frame and the car is light in color).

In Figure 11 two cases are shown, the first close to the ideal, and at the second, the plate doesn’t have a frame and the car is light in color.


**Figure 11**. Obtaining the larger area.

## V. Identification of characters

To identify the region occupied by the characters, the negative of the image obtained in the previous stage was used. The negative of a binary image, consists in changing the pixels 1 to 0 and vice versa. Afterwards the rectangle of least dimension containing each connected component is obtained. The length and width of each rectangle were used as training patterns of a Kohonen neural network.

### 5.1. Kohonen neural network

There are two variants of the model, called LVQ (Learning Vector Quantization) one-dimensional and SOM (Self Organization Map) usually two-dimensional but could even be three-dimensional. The LQV variant was used in this work.

#### 5.1.1. Architecture LQV

The original version (LVQ) consists of two layers, with N input neurons and M output neurons. Each of the N input neurons are connected to the M output neurons through forward connections (feedforward), as illustrated in Figure 12.

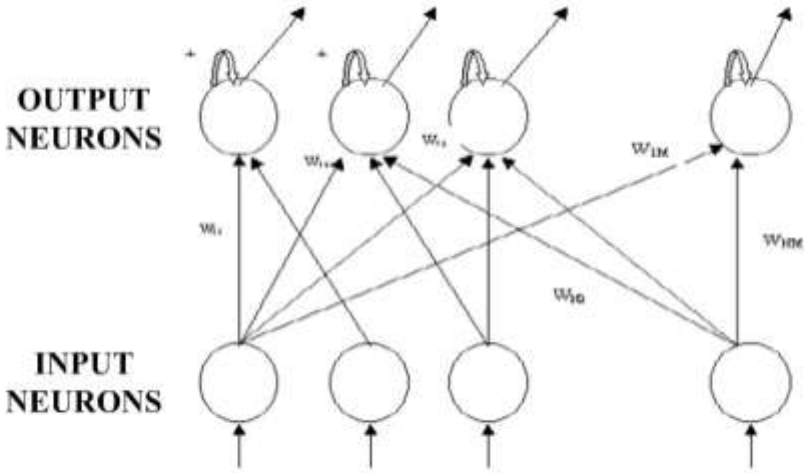


**Figure 12**. LQV Architecture

**Operation**

1. The synaptic weights $W_j$ are initialized with default or random values for every j=1,…,M, where;

$$W_j = [W_{1j}, W_{2j}, \cdots, W_{Nj}] \quad (2)$$

2. For every t = 1,…,T and for every k=1,…P, where T is the number of epochs and P is number of training patterns. Steps 3 and 4 are realized.

3. The input pattern $E^k$is presented to the net, where;

$$E^k = [E_1^k, E_2^k, \cdots, E_N^k] \quad (3)$$

and the euclidean distance is calculated for every j=1,..,M, that is:

$$d_{kj} = \left\| E^k - W_j \right\| \quad (4)$$

4. The winner neuron j* is the one that has the least euclidan distance. Then, the weight $W_{j*}$ is updated, following the learning rule in (4), while the rest of the weights are maintained without change.

$$W_{j*} = W_{j*} + \beta(t)\left[E^k - W_{j*}\right] \quad (5)$$

where the parameter β(t) is denominated as a learning parameter and is a bounded value between 0 and 1 that decreases with t. The next expression may be used:

$$\beta(t) = \frac{1}{t} \quad (6)$$

In this work the number of input neurons is N=2, since the input patterns are defined like points in the plane $E^k=[w^k, h^k]$, where $w^k$, $h^k$represent the width and length respectively of the container rectangle of the connected component k. Besides, the dimensions $w^k$, $h^k$were normalized respect to the maximum width and length respectively, that is:

$$\widehat{w^k} = \frac{w^k}{w_{max}} \widehat{h^k} = \frac{h^k}{h_{max}} \quad (7)$$

On the other hand, the number of neurons in output core is M=4. In figure 13 the prefixed default values are indicated with the ones that initialized the synaptic weights. At ideal conditions, it would be expected that in the map would be defined the 4 zones that are indicated in figure 13. The zone 4 is the zone that has less probability to classify the characters, due that this zone classifies components with proportion w/h > 1, andthe characters maintain a proportion w/h <1. However, this zone is necessary to classify and to discard these type of components. In any of the other zones, the characters can be classified.

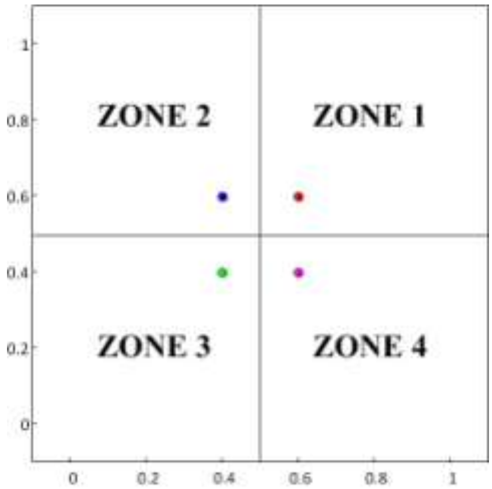


**Figure 13**. Initialization of weights and ideal zones of classification

In Figure 14, three cases are exhibited, where the zone is indicated where the characters were classified.

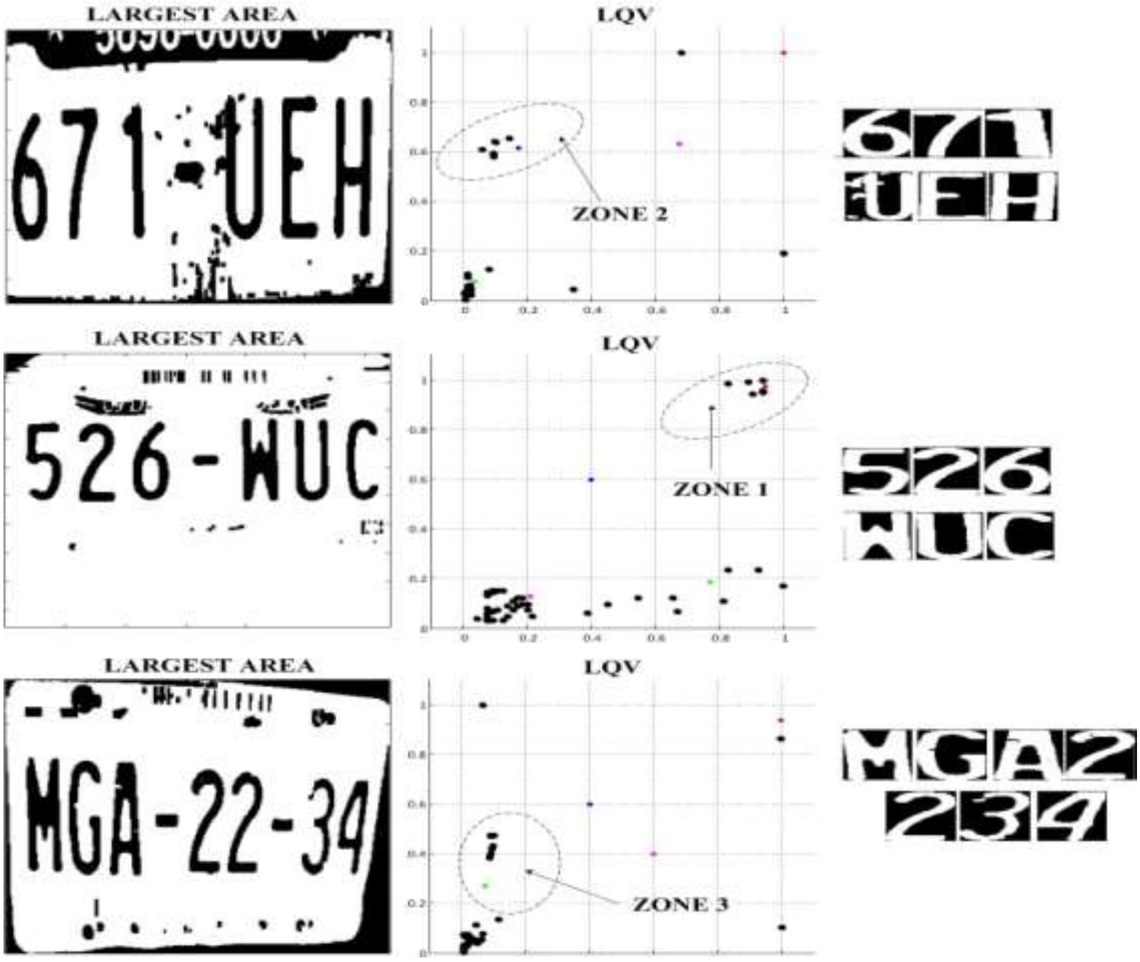


**Figure 14**. Classification zones of the characters.

## VI. Conclusions

This paper presents a method for identifying the position, length and width of the rectangle of least dimension that contains the characters in an image taken from a vehicular plate, when the plate's location has not been accurate. It should be mentioned that from the identification in some cases is not possible to realize the extraction, since it should be applied to a geometric correction, plus it is also possible that still exists noise in the region occupied by some characters. However, the objective is not to have a prior stage to the extraction, instead useful information for the identification of the region occupied by the characters in the image and necessary geometric correction.

## References


[1] Andrade Miranda Gustavo, López Encalada José y Chávez Burbano Patricia,Vehicle control system using optical character recognition, http://www.dspace.espol.edu.ec/bitstream/123456789/1458/1/2973 .pdf,

[2] Richard Gutiérrez, Ma. Fernanda Frydson y Phd. Boris Vintimilla, Application of computer vision for automatic recognition of vehicular plates using conventional OCR's www.dspace.espol.edu.ec/.../Paper_**Gutierrez**_Frydson_Vintimilla.pdf

[3] Angulo Jesús, Segmentación de imágenes http://cmm.ensmp.fr/~angulo/publicat/AnguloSerra_CyS_05.pdf

[4] Duarte Domingo, Image segmentation using its histogram http://www.unne.edu.ar/unnevieja/Web/cyt/cyt/2001/7-Tecnologicas/T-084.pdf

[5] Cáceres Tello Jesús, Artificial vision and morphological operations in binary images, http://www2.uah.es/jcaceres/uploaded/articulos/FormatoVisonArtificial.pdf,

[6] National University of Quilmes http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Operaciones%20Morfol%C3%B3gicas%20en%20Im%C3%A1genes%20Binarias%20-%20parte%203.pdf

[7] The Kohonen model,http://thales.cica.es/rd/Recursos/rd98/TecInfo/07/capitulo6.html,

[8] www.mathworks.com/help/images/ref/


## AUTHOR PROFILE:

**Silvestre Ascención García Sánchez**. Born in Mexico DF on Jul 11, 1975, holds a degree in Mathematical Physicistfrom the National Autonomous University of Mexicoin and earned his Master of Science in Engineering with specialization in Microelectronics Digital Signal Processing by the National Polytechnic Institute in 2005.

**Carlos Aquino Ruiz** was born in Mexico on April 2, 1973 is in Communications and Electronics Engineering from the National Polytechnic Institute IPN, a research professor and areas of development are applied computing, data networking and security.

**Dolores Zamorano Saavedra.** She is a methodological research and has master's degree in Educational Sciences. Nowadays it is a research professor at the School of Mechanical and Electrical Engineering (ESIME) in Mexico. Her areas of development are assessment and project development, scientific writing and methodology of science