N93-29536

Character Recognition Using a Neural Network Model with Fuzzy Representation

S20-63
161820
P-10

NASSRIN TAVAKOLI AND DAVID SENIW
Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC 28223

Abstract

The degree to which digital images are recognized correctly by computerized algorithms is highly dependant upon the representation and the classification processes. Fuzzy techniques play an important role in both processes. In this paper the role of fuzzy representation and classification on the recognition of digital characters is investigated. An experimental Neural Network model with application to character recognition has been developed. Through a set of experiments, the effect of fuzzy representation on the recognition accuracy of this model is presented.

Keywords: Statistical, Syntactical, Neural Network, Fuzzy Techniques

1. Introduction

Three primary processes are utilized in most pattern recognition systems. 1) The representation process in which the raw digitized data is mapped into a higher level form, such as a feature vector (statistical techniques) or pattern elements constituting pattern grammars (syntactical techniques). 2) The generation of a known base containing the high-level representation of all known patterns in a problem domain. 3) The identification/classification process which classifies the unknown pattern, given its high-level representation and the known base. A block diagram of a general pattern recognition system is given in Figure 1 [16].

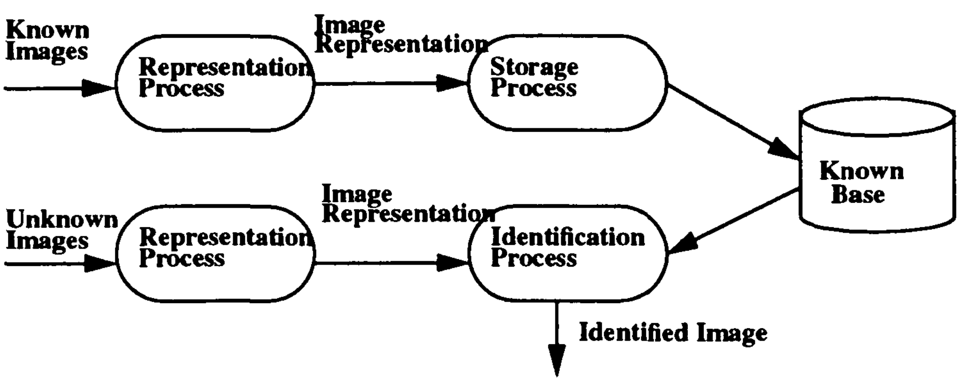

FIGURE 1. A general pattern recognition system.

In both the storage and the identification processes, representation of the image plays a very important role. In fact, the techniques and algorithms used to store the image representation, and the selection of identification techniques are strongly tied to the methods used for representation.

This paper starts by a short introduction to pattern recognition techniques and the role of fuzzy theory on these techniques. We have selected and implemented a neural network model as the identification process. This model is regarded as a fuzzy classifier since it provides the degree of membership rather than an exact match. For the representation process we start with raw images and through pyramid reductions, provide different levels of resolutions (exactness/fuzziness). Representations of images at each level are used as inputs to the classifier. The purpose is to find an appropriate representation level for the this model and gain some understating on the level of fuzziness required in general (regardless of the classifier) that could result in the best recognition. For comparison, the same character representations were used in conjunction with a template matching identification process.

The results of experiments on a set of 702 unknown digitized characters are given.

2. Pattern Recognition Techniques

Although no unified approach exists for pattern recognition, the majority of techniques that have been developed are in general categorized into two major approaches, namely, the statistical [1,2,10,17] and the syntactical [3,4,6,7,12,14,15] approaches. A distinguishing factor between the syntactical and statistical approaches is the representation and identification processes. Fuzzy techniques play an important role in both the syntactical and Statistical approaches. A thorough discussion and review of fuzzy techniques in pattern recognition is given by Kandel [8].

Statistical approaches use a feature component vector, where the vector contains representations of independent pattern elements that are extracted from the image. The identification/classification process is based on a similarity measure that in turn is expressed in terms of a distance measure or a discriminant function. Fu [5] provides a discussion of several important discriminant functions.

Syntactical approaches represent the image as a tree or graph of pattern elements and their relationships. A set of syntax rules, called pattern grammar, is used to represent this relationship. This type of representation would require the identification process to use syntax parsing techniques.

The *fuzzy set theory* introduced by Zadeh [19] has played an important role in both statistical and syntactical approaches. The main purpose of using fuzzy sets has been to represent the inexactness of patterns belonging to certain categories. In statistical methods, the classification algorithm yields the degree of membership of an object in a particular class. In syntactical methods, fuzzy formal languages [11]and parsing methods have been introduced.

Neural Network models used in pattern recognition can be considered as a statistical approach in which the classifier (i.e. the neural net) provides the degree of the membership of the unknown object in each of the known classes. Hence the neural network model can be considered as a fuzzy classifier.

3. The Representation Process

The input to the representation process consists of a known base of 26 digitized characters with 5 instances of each character and an unknown base of 702 characters used for recognition. A sample of these characters are shown in Figure 2. These characters were extracted from a digitized text scanned at 240 pixels/inch.

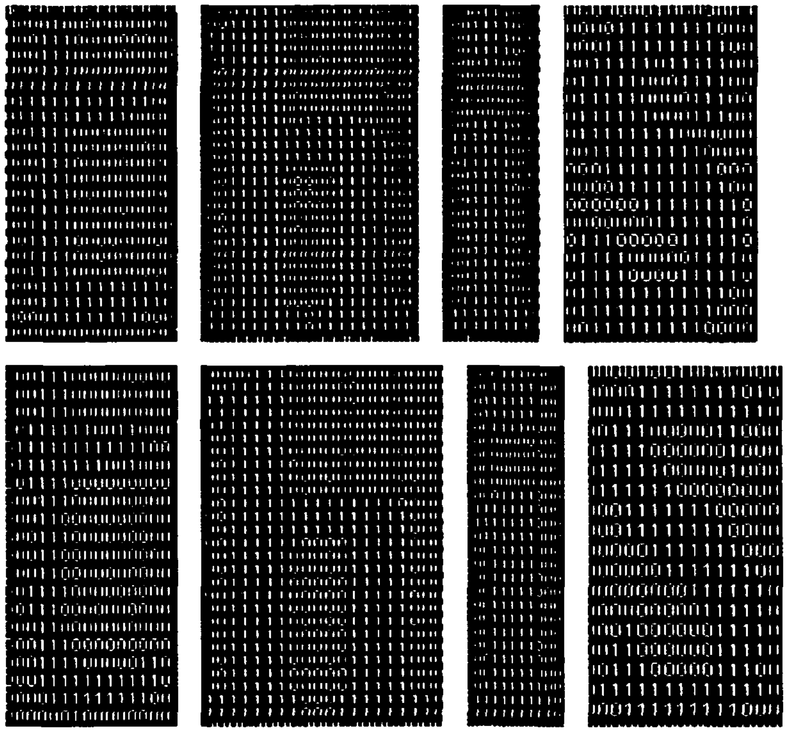

FIGURE 2. Sample of Digitized Characters

As noticed from this sample, the characters are noisy, with ragged edges and the digitized representations of the same character are not identical.

In the experimentations, different representations of these characters are used. These representations are:

1. The raw image. As shown in figure 2, the raw image is a digitized character converted into a binary form of zeros (spaces) and ones (body of the character).
2. A pyramid reduction of 2
3. A pyramid reduction of 3
4. A pyramid reduction of 4
5. A pyramid reduction of 5
6. A pyramid reduction 6
7. A pyramid reduction of 7
8. A pyramid reduction of 8

A pyramid is a successive reduction of an image to a lower resolution by representing a block of an image with one pixel. The value of this pixel is determined by the ratio of dark to light pixels (i.e. the threshold factor). We have used a threshold factor of .45 in all pyramid reductions. The selection of this threshold factor was due to a series of experimentations for finding the most optimal value. Figure 3 shows the result of the pyramid reduction.

```
00000000111111111110000000000000
00000111111111111111111000000000
00000111111111111111111000000000
00011111111111111111111110000000
00011111111111111111111110000000
01111111111101111111111111000000
01111111110000000111111111000000
01111111110000000111111111000000
11111111000000000111111111000000
11111111000000000111111111000000
01111111000000011111111111000000
01111111000000011111111111000000
00011100000001111111111111000000
00000000111111111111111111000000
00000000111111111111111111000000
00000111111111111111111111000000
00000111111111111111111111000000
00011111111110000011111111000000
01111111110000000111111111000000
01111111110000000111111111000000
11111111000000000111111111000000
11111111000000000111111111000000
11111111000000000111111111000000
11111111000000000111111111000000
11111111000000011111111111111100
11111111111111111111111111111111
11111111111111111111111111111111
01111111111111111111111111111100
01111111111111111111111111111100
00011111111110000000001111000000 0
00011111111110000000001111000000 0
```

No reduction

```
001111111110000
011111111111000
111111111111100
111110001111100
111100001111100
111100011111100
011011111111100
001111111111100
011111111111100
111110001111100
111100001111100
111100001111100
111111111111111
111111111111111
111111111111111
```

Pyramid reduction of 2

```
0011111000
0111111100
1110001100
1110011100
0001111100
0111111100
1110001100
1110001100
1111111111
1111111111
```

Pyramid reduction of 3

```
0111110
1111110
1100110
0011110
1110110
1100110
1111111
```

Pyramid reduction of 4

```
011110
110110
011110
011110
110110
111111
```

Pyramid reduction of 5

```
01110
11010
01110
11010
11111
```

Pyramid reduction of 6

```
1110
1011
1111
1111
```

Pyramid reduction of 7

```
111
111
101
```

Pyramid reduction of 8

FIGURE 3. A sample of reduced characters

4. The Neural Network Model

The Neural Network Model implemented for these experiments is based on the Bidirectional Associative Memory Model[9,18] with two layers, without the feed back mechanism. In terms of recognition accuracy this model may not be the most optimal model for this application. However, our purpose was to examine the effect of different levels of representation on the recognition accuracy. Figure 4 shows a high level representation of this model.

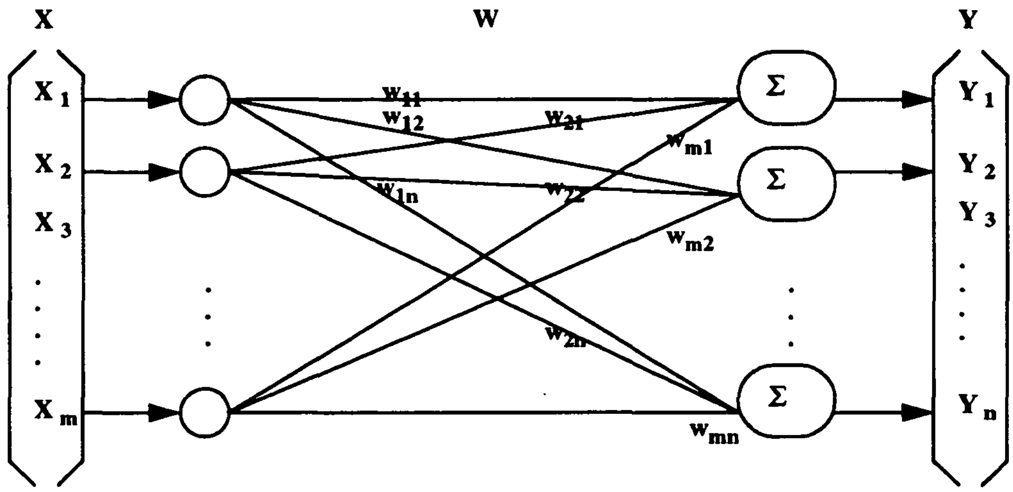

FIGURE 4. The Neural Network Model

This model uses two layers, an input and an output layer. No weighting is performed at the input layer. The links between each of the input nodes and each of the output nodes is weighted with an integer value (positive or negative). The output nodes sum each of the input node values multiplied by it's associated weight. The result of this summation is represented by the vector Y.

The implementation of the model using the character representations are shown in the following section.

4.1 Implementation of the Neural Network Model

The input layer consists of a node for each feature of the character being recognized. Initially each feature is equivalent to one pixel element. The value of this feature is either zero or one depending on whether the pixel is light or dark. However, after the image is subjected to a pyramid reduction, each feature now represents several pixels. Initially, each character is represented by 31*31 or 961 pixels. The output layer is made up of 26 different nodes (one for each possible character). Figure 5 shows an implementation model of this model using the input vector X, the output vector Y and the weight matrix W.

$$[x_1, x_2, \ldots x_m] \begin{array}{c} \begin{array}{ccccc} a & b & c & & z \end{array} \\ \begin{pmatrix} w_{11} & w_{12} & w_{13} & \cdots\cdots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \cdots\cdots & w_{2n} \\ w_{31} & w_{32} & w_{33} & \cdots\cdots & w_{3n} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ w_{m1} & w_{m2} & w_{m3} & \cdots\cdots & w_{mn} \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} a & b & z \end{array} \\ = [y_1, y_2, \ldots y_n] \end{array}$$

$$X \qquad\qquad\qquad W \qquad\qquad\qquad Y$$

FIGURE 5. The Neural Network Implementation Model

In this application, X, Y, and W contain the following values.

X = a vector of pixels belonging to one character.

Y = a vector of positive and negative integers where the index to the vector represents a character such that:

$Y(a) = y_1$

$Y(b) = y_2$

.

.

$Y(z) = y_{26}$

W = a weight matrix where the columns are associated with different characters and rows are associated with the weight per pixel of each character.

$W(a) = w_{11}$ to w_{m1}

$W(b) = w_{12}$ to w_{m2}

.

.

$W(z) = w_{1n}$ to w_{mn}

m = number of pixels per character

n = number of classes of character = 26

Each character is 'taught' to the network by modifying the weight matrix. During the identification phase, the output node with the largest (or the most positive) value indicates the class in which the unknown character belongs to. Note that output nodes with the second, third, etc. largest values indicate characters that are similar to that particular character. In an actual Neural Network hardware solution, the input layer would be 961 processors, each collecting information about their pixel. The output layer would be made up of 26 processors each containing the 961 weights for modifying the signals coming from the input layer nodes. These weights would be created during the learning phase. In our simulation the weights are all stored in a two dimensional matrix.

4.2 Teaching the Neural Network

Teaching the Neural Network is the process of recursively modifying the values of the weight matrix W. This process consists of the following steps:

1. Convert all characters into a one dimensional vector of 0's and 1's.
2. Convert the vector of 0's and 1's into a bipolar vector with negative ones (-1) representing the zeroes. This constitutes the vector X of length m as shown in figure 5. The value of m varies from 961 (no pyramid reduction) to 9 (pyramid reduction of 8).
3. Initialize matrix W to 0.
4. Calculate the weight matrix using the following algorithm:

```
For l = a to z
   Y(l) = -1
End
For Instances = 1 to 5
   For k = a to z
     Y = -1                    /* initialize Y vector
     Y(k) = 1
     For i = 1 to m
       For j = 1 to n
         wij = wij + xi * yj
       End
     End
   End
End
```

Figure 6 shows two instances of the weight matrix after characters "a" and "b" with a pyramid reduction of 5 have been taught to the network.

```
             Bipolar
Character    Vector
'a'          'a'    The Weight Matrix after learning 'a'

011110       -1     -1 1 1 ... 1
110110        1      1-1-1 ...-1
011110        1      1-1-1 ...-1
011110        1      1-1-1 ...-1
110110        1      1-1-1 ...-1
111111       -1     -1 1 1 ... 1
              1      1-1-1 ...-1
              1      1-1-1 ...-1
              .      .
              .      .
              1      1-1-1 ...-1
              1      1-1-1 ...-1
              1      1-1-1 ...-1
              1      1-1-1 ...-1

                     1-1-1 ...-1 <--- the Y vector
```

```
Character Bipolar Vector
'b'          'b'    The Weight Matrix after learning 'a' and 'b'

110000        1     -2 2 0 ... 0
010000        1      0 0-2 ...-2
011111       -1      2-2 0 ... 0
010001       -1      2-2 0 ... 0
010001       -1      2-2 0 ... 0
011110       -1      0 0 2 ... 2
             -1      2-2 0 ... 0
              1      0 0-2 ...-2
             -1      0 0 2 ... 2
              .      .
              .      .
              1      0 0-2 ...-2
              1      0 0-2 ...-2
              1      0 0-2 ...-2
             -1      2-2 0 ... 0

                    -1 1-1 ...-1 <--- the Y vector
```

FIGURE 6. Snapshot of the weight matrix after teaching characters "a" and "b"

4.3 Recognizing Characters with Neural Network

The recognition of a character requires the linearization and bipolarization of each of the image features (just as in the teaching section). Then simply perform a matrix multiplication of the vector X on the weight matrix W, placing the result in the vector Y. The closest match is obtained by finding the largest positive value in Y (or the smallest negative value, as Y is usually composed of all negative values). The character associated with index of the node of the largest value is selected. For example, if the third node in Y (i.e. Y(c)) had the largest positive value or the smallest negative value, then the recognized character is "C". Note that other close matches may be discovered by finding the second, third, fourth, etc. largest values in Y. Figure 7 shows a snapshot of the state of network when character "C" is recognized.

```
Unknown        Bipolar
Character      Vector  The Weight Matrix after all characters have been taught

011111         -1      -18-6-26...-16
110011          1      -78-78-82...-88
100000          1      -14-34-14...-24
110000          1      -14-34-14...-24
110011          1      -56-76-56...-66
011110          1       4 4 16 ... 14
                1       24 4 20 ...14
                1      -54-54-54...-64
               -1       36 36 36 ...46
                .       .
                .       .
                1      -76-76-76...-86
                1      -54-58-54...-64
                1      -48-48-48...-58
               -1       18 -2 -2 ... 8

                       -658 -742 -454...-730 <--The results of the matrix multiplication of
                        Y(a)Y(b)Y(c) ...Y(y)    unknown character and the weight matrix.
```

FIGURE 7. Snapshot of the network during recognition

In this example, the least negative value was the integer associated with Y(c), indicating that the neural nets choice for the image was a 'c'.

5. Experimental Results

A series of experiments were conducted for recognition of the 702 digitized characters. These experiments were varied over the following parameters:

1. The identification process (a. the neural network model, b. the template matching).
2. The number of instances (1,2,3,4,5) of each known character for teaching the identification process.
3. The image representation (raw, pyramid reductions of 2, 3, 4, 5, 6, 7, and 8).

Figure 8 shows the effect of each varying parameter on the recognition accuracy of the 702 unknown characters.

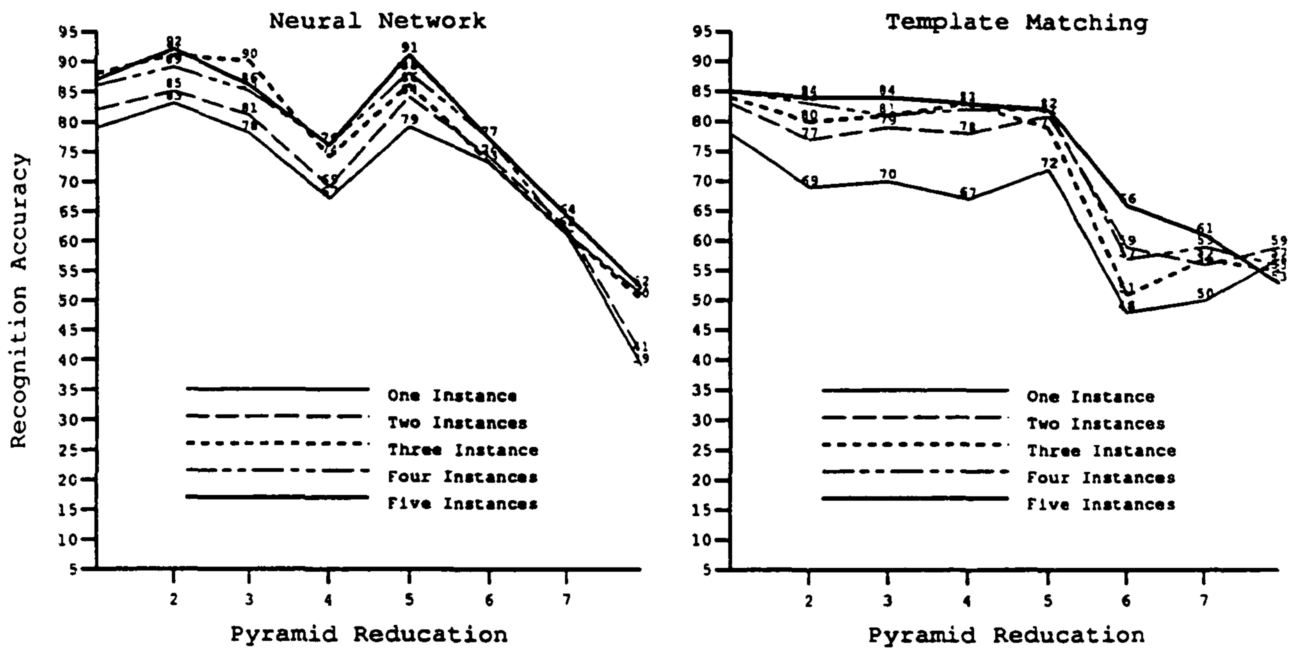

FIGURE 8. Comparison of Neural Network and Template Matching approaches.

The following conclusions can be drawn from the above figure.

a) The Neural Network approach can provide a higher recognition accuracy than the template matching approach. This is due to the fact that the Neural Network approach is a fuzzy classifier whereas the template matching approach is an exact classifier.

b) In general as the number of known instances for each character increases, better recognition is achieved in both approaches. The Neural Network approach however, slightly deviates from this fact. In one case (i.e. pyramid reduction of 3) three instances provide better recognition than five instances.

c) In template matching, the raw image (i.e. no pyramid reduction) is better than any pyramid reduction. As noticed, the curve is almost flat, implying that this approach is less sensitive to the representation process. In Neural Network approach, peaks are shown at pyramid reductions of 2 and 5.

d) In general, a pyramid reduction of 5 seems to provide a good recognition in both approaches. This is an interesting phenomenon since at this representation level, it is more difficult for the human vision to recognize accurately (see figure 3).

6. Conclusion

The purpose of this research was to gain some understanding on the level of fuzziness required in the representation and the identification processes for better recognition of digitized characters. We also were interested in the degree of the interdependency between the representation process and identification process.

A Neural Network model and a Template Matching model for recognition of digitized characters were implemented. For the representation process, we used pyramid reductions of 1 to 8. Through a series of experiments we concluded that the optimal amount of fuzziness to be introduced by the representation process is totally dependant upon the identification process. As expected some level of fuzziness in both the representation and the identification processes contributed to better recognition. The Neural Network approach in general proved to be a better identifier due to its fuzzy classification property. While no general representation can be found to be the optimal, it seems that a pyramid reduction of 5 provided good recognition (i.e. above 80%) in both models.

With further experiments, we found that different approaches for the representation and identification, resulted in the recognition of a different sets of character. By combining two different techniques therefore, a recognition of 100% on the same set of characters were achieved.

References

[1] Devijver, P. A., and Kittler, J., *Pattern Recognition, A Statistical Approach*, New York:Academic Press,1982.

[2] Fukunaga, K., *Introduction to Statistical Pattern Recognition*, New York:Academic Press, 1972.

[3] Fu, K. S., *Syntactic Methods in Pattern Recognition*, New York:Academic Press, 1974.

[4] Fu, K. S., "Recent Advances in Syntactic Pattern Recognition." *Proc. 4th Int. Joint Conf. on Pattern Recognition*, 1978, pp. 13-18.

[5] Fu, K. S. (Ed.), *Digital Pattern Recognition*, New York:Springer-Verlag, 1980.

[6] Fu, K. S., *Syntactic Pattern Recognition and Applications*, Englewook Cliffs:Prentice-Hall, 1982.

[7] Gonzalez, R. C. and Thompson, M. G., *Syntactic Pattern Recognition*, An Introduction, Reading:Addison Wesley, 1978.

[8] Kandel, A., *Fuzzy Techniques in Pattern Recognition*, John Wiley & Sons, 1982.

[9] Kosko, B., "Constructing an Associative Memory", *Byte*, Sep. 1987, pp. 137-144.

[10] Kovalevsk, V. A., "Recent Advances in Statistical Pattern Recognition." *Proc. 4th Int. Joint Conf. on Pattern Recognition*, 1978, pp. 2-12.

[11] Lee, E.T., "Fuzzy Languages and Their Relation to Automata", *Ph.D. Thesis*, Dept. of Elect. Eng. and Comput. Sci., Univ. of California, Berkeley, 1972.

[12] Pavlidis, T., *Structural Pattern Recognition*, Springer-Verlag, 1977.

[13] Segen, J., "Learning Structural Descriptions of Shape.," *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 1985, pp. 96-99.

[14] Tanimoto, S. and Klinger, A. (Eds.), *Structural Computer Vision*, New York:Academic press, 1980.

[15] Thomason, M. and Gonzalez, R. C., *Syntactic pattern recognition: an introduction*, Addison-Wesley, 1981.

[16] Tavakoli, N., "A New Approach to Pattern Recognition.", *Doctoral dissertation*, Kansas State University, 1986.

[17] Young,T. Y. and Fu, K. S., (Eds.), *Handbook of Pattern Recognition and Image Processing*, New York:Academic Press,1986.

[18] Wasserman, P. D., *Neural Computing: Theory and Practice*, Van Nostrand Reinhold, New York, 1989.

[19] Zadeh, L. A., "Fuzzy sets", *Inf. and Control*, vol. 8, 1965, pp. 338-353