*Stochastic Modelling and Computational Sciences*
*Vol. 2 No. 2 (December, 2022)*

*Received:12th June 2022* *Revised: 15th August 2022* *Accepted: 16th October 2022*

# Handwritten Numeral Recognition for mistake free Fuzzy System Grammar

*Rudra Pratap, Department of Mathematics, Meerut University, INDIA*

**Abstract**

The different methods for automatic pattern recognition are motivated by the way in which pattern classes are characterized and defined. In this paper, handwritten numerals are preprocessed and segmented into primitives. These primitives are measured and labeled using *fuzzy logic*. Labeled-strings of characters are formed from the labeled primitives. The handwritten characters (Numerals) are recognized through a *Modified Parser* generated from the *Error-free Fuzzy Context-Free Grammar.*

## 1. INTRODUCTION

A lot of research effort has been dedicated to handwritten character recognition [4]. Number of schemes are available for this purpose. Some of the areas where the handwritten character recognition is being carried are Fuzzy Methods [19] Knowledge-based techniques using Neural Networks [14, 26] and Markovian Model [17]. The different methods for automatic pattern recognition are motivated by the ways in which pattern classes are characterized and defined [16]. The idea in syntactic pattern recognition is to describe a complex pattern in terms of a hierarchical composition of simple sub-patterns [11]. In syntactic pattern recognition a basic set of primitives forms the terminal set of grammar [6]. The pattern class is the set of strings generated by the pattern grammar. But the concept of formal grammar is too rigid to be used for the representation of real-life patterns such as handwritten documents.

This rigidity can be changed if certain fuzziness is introduced which describes the vagueness of such patterns. Accordingly a fuzzy language can handle imprecise patterns when the indeterminacy is due to inherent vagueness [9]. The conventional approaches to knowledge representation usually lack the means to represent the imprecise concepts. Due to Zadeh [27], Fuzzy sets offer a theoretical basis to cope with the vagueness of patterns, which we have exploited in the proposed method. First the motivation for this method is given. How Fuzzy Context-free Grammar is applied on the handwritten numerals is presented along with results.

## 2. MOTIVATION OF THIS METHOD

In recent years, development tools in fuzzy software and hardware such as Fuzzy Clips [15], FUNN-Lab [8] have been introduced. These tools provide a convenient way to configure the membership functions, defining rules, input and output functions etc. But they are not suitable for highly structured pattern recognition. The symbolic and structural description of a pattern is more useful for analysis and recognition [21]. The allograph-based method to recognize cursive handwritten words with fuzzy logic has been proposed by Parizeau et al. [18]. The drawback of this method is that, there is no direct way of generating handwriting feature allographs automatically. Malaviya et al. [15] have proposed FOHDEL a new fuzzy language for automatic generation of a pattern description in a rule-base and the representation of patterns in a linguistic form. The problem with this method is that, the large number of input features make the rule-base incomprehensible and consumes more time for recognition. The theory of Fuzzy grammars and quantitative fuzzy semantics [7] give very interesting ideas like the connection between contextfree grammar and natural grammar through transformational grammar and the derivation trees (structural descriptions or pattern markers). The idea here is to construct labeled strings using Fuzzy Logic. The labeled strings are being compared with prototype strings to recognize the pattern (handwritten character).

The purpose of this paper to offer a system which infers a complete Error-handling Fuzzy Context-Free Grammar (FCFG) from samples and generates a fuzzy language as sets of trees and match the strings for recognition.

## 3. APPLICATION OF FUZZY CONTEXT-FREE GRAMMAR TO NUMERAL PATTERN RECOGNITION

### 3.1 Handwritten Numerals Recognition

Character recognition requires a preprocessing, learning and recognition stages. The preprocessing takes different stages. Learning Stage in most of the early work in character recognition were based on the use of correlation techniques and probabilistic concepts. But, Fuzzy Technique has been used here for the primitive identification and recognition. The general flow of the work in this paper is shown in the figure 1.

Handwritten numerals are considered here as a case study. Handwritten characters are having biological origin, since depending on the mood of a person his handwriting varies and hence variability in all sense is possible in the input image. So it is considered to recognize such patterns. Examples of handwritten digits are given in figure 2 and 3.

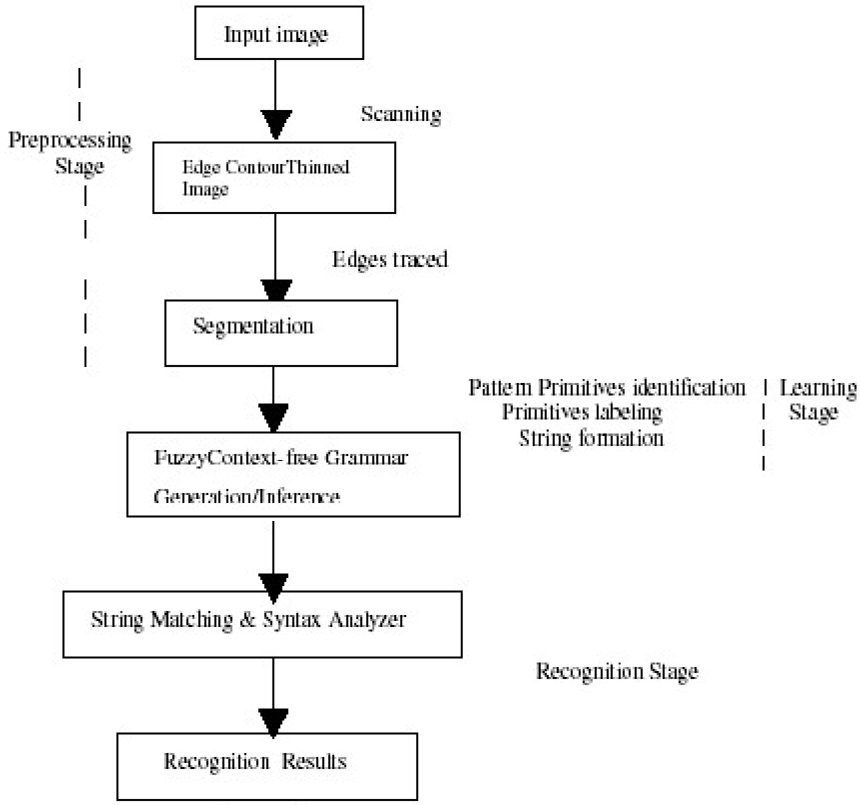



**Figure 1: Flow of Recognition**

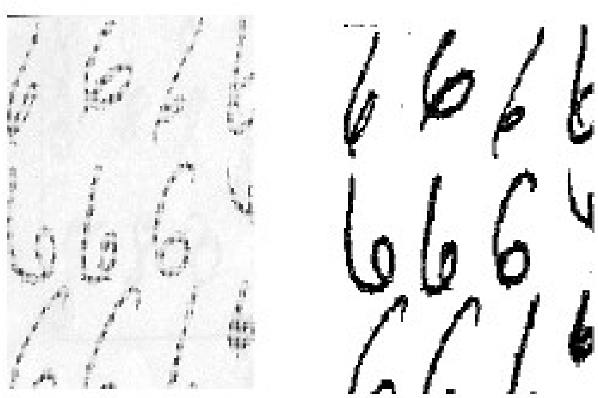


**Figure 2: (a) Gray Image of Handwritten Digit Six (b) Two tone Image of Figure 2 a)**

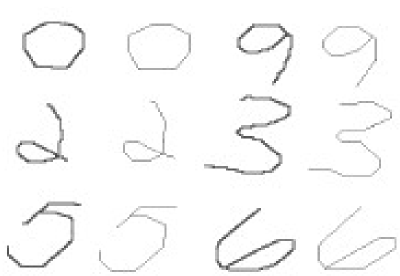
### 3.2. Preprocessing (Edge Detection Smoothing and Thinning)

The scanned (digitized) input is preprocessed using the steps as discussed in [20]. The preprocessed result for some sample images is shown in Figure 3.

**Figure 3: Edge detected, Smoothened and Thinned Numerals**

### 3.3. Polygonal Approximation

The efficient representation of irregular curves is an important problem in picture processing. Approximation of such irregular digitized curve by straight line segments is known as polygonal approximation or piecewise linear approximation[9]. Any digitized curve can be approximated by a polygon with any desired degree of accuracy. The curves are interpreted as polygons in which the vertices lie on corner points and edges coincide with pixels. The thinned image of input characters given in figure 2.3 looks like a polygon. But then, real polygonal approximation is necessary for the numbers ‘2’, ‘3',’6' and ‘9’. Several papers on the subject of smoothing, quantized contours, polygonal approximation [1] have described various approaches and demonstrated the techniques. We have developed an algorithm to perform polygonal approximation with a small number of edges for arbitrary twodimensional digitized curves and the result of that is given below



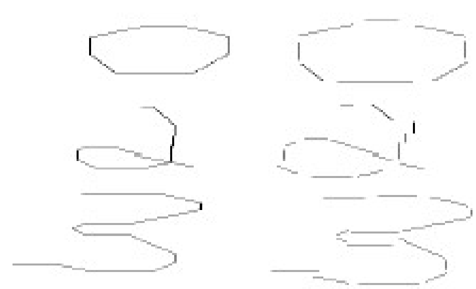


**Figure 4: Polygons for some of the Numerals** **Figure 5: Segmented Primitives of Numerals**

### 3.4. Feature Selection (Segmentation)/Primitive Identification (Recognition)

Feature selection and extraction are the most significant aspects of any pattern recognition problem. The features should be selected in such a way that the resulting description is independent of skew, contrast, deformation or other style of writing. In handwritten character, the normal variation due to style and other aspect of writers should not affect the feature. In other words the feature selected should be insensitive to the deformation of the character. Detailed discussion on feature selection and extraction with their importance can be found in [20]. Structural details like endpoints, intersections of line segments, loops, curvatures, segment lengths, etc. describing the geometry of the pattern structure are used as features. The details of feature's discriminative power is well documented in literature [25]. Many authors have used number of such features like endpoints, branches, junctions, corners, curvatures, line lengths, curve shapes etc. [22] to describe characters. However, none of them employed a flexible and unified representation. In the present work the iterative procedure has been applied on the structure for polygonal approximation of plane curve. A feature is defined as a set of vertices on or near the pattern boundary (line) and the segmented line lengths are obtained from them. The structure of a character is represented by this feature. Also it is easy to reconstruct a character from them. The pattern primitives are identified, recognized (Fuzzy functions) and labeled using the procedures developed by us and the result is shown below.

***Algorithm1: Fuzzy Context-Free Grammar Inference***

1. The set of sample strings are considered for determining the cycle in the strings. The order of frequencies of cycles in the strings are listed.
2. For each cycle a, a production rule $A\rightarrow \alpha A$ is derived.
3. For any intermediate substring $\beta$ occurring between two consecutive cycles a production rule is formed as $A\rightarrow\beta A$.
4. For a substring $\gamma$, at the end of a string (after the deletion of the cycles) a rule $A\rightarrow \gamma$ is formed;
5. For any string $\eta$ at the beginning of a string (before a the cycles start) a rule is formed as $S\rightarrow \eta A$.
6. For any string $\varphi$ which does not contain a cycle, a rule $B\rightarrow \varphi$ is formed.
7. Any redundancy in the production rule is avoided.

***Algorithm 2: Fuzzy Membership value determination***

Input: The production rules.

Output: The membership values for each production rule.

1. Identify the set of production rules (derived using the above algorithm) required for the entire sample set.
2. Obtain the frequency of each of them
3. The possibility of occurrence of a production rule is determined and defined as a membership value of that rule.

   The inferred grammar is as follows:

$$\overset{1.0}{S\rightarrow A},\quad \overset{0.6}{S\rightarrow hA},\quad \overset{0.6}{S\rightarrow hlA},\quad \overset{0.7}{S\rightarrow B},\quad \overset{0.8}{S\rightarrow AB},$$

$$\overset{0.9}{A\rightarrow hrvA},\overset{0.9}{A\rightarrow hlrA},\overset{0.8}{A\rightarrow rhA},\ \overset{1.0}{A\rightarrow vA},\ \overset{1.0}{A\rightarrow lA},\ \overset{0.6}{A\rightarrow hrh},\ \overset{0.6}{A\rightarrow rv},\ \overset{0.8}{A\rightarrow \varepsilon},$$

$$\overset{0.6}{B\rightarrow rvr},\ \overset{0.6}{B\rightarrow hll},\ \overset{0.6}{B\rightarrow hl}$$

### 3.5. Error-Free Grammar Generation

However, there is a possibility that the grammar inferred need to accommodate strings of the following types.

(i) Strings which differ from a prototype string in exactly one place.

(ii) Strings which can be obtained from a prototype string by deleting one symbol.

(iii) Strings which can be obtained from a prototype string by inserting one symbol.

To accomplish this, a new error-free grammar is generated from the inferred grammar $G$. To accommodate strings of type (i), the set of $V_N$ of non terminals of $G$ is replaced by $V_N \cup \{ A' \mid A' \in V_N\}$. For each production rule of the form $A \xrightarrow{\tau} B$, a new production rule $A' \xrightarrow{\tau} B'$ is added. For a production of the form $A \xrightarrow{\tau} BC$, two new production rules $A' \xrightarrow{\tau} B'C$ and $A' \xrightarrow{\tau} BC'$ are added. To accommodate strings of type (ii), for each $A \in VN$, two production rules of the form $A \xrightarrow{\tau} \varepsilon$ and $A' \xrightarrow{\tau} \varepsilon$ are added. To accommodate strings of type (iii), for each production rule of the form $A \xrightarrow{\tau} \varepsilon\, a$, new production rules of the form $A' \xrightarrow{\tau} ab$ are added for all $b \in V_T$.

The new error-free grammar with additional productions rules to accommodate all the strings is given below:

G’ = ({ S, A, B, S’, A’, B’},

{ h, v, l, r}

$$S \xrightarrow{1.0} A,\quad S \xrightarrow{0.6} hA,\quad S \xrightarrow{0.6} hlA,\quad S \xrightarrow{0.7} B,\quad S \xrightarrow{0.8} AB,$$

$$S' \xrightarrow{1.0} A',\quad S' \xrightarrow{0.7} B',\quad S' \xrightarrow{0.8} A'B,\quad S' \xrightarrow{0.8} AB'$$

$$A \xrightarrow{0.9} hrvA,\quad A \xrightarrow{0.9} hlrA,\quad A \xrightarrow{0.8} rhA,\quad A \xrightarrow{1.0} vA,\quad A \xrightarrow{1.0} lA,\quad A \xrightarrow{0.6} hrh,\quad A \xrightarrow{0.6} rv,\quad A \xrightarrow{0.8} e,$$

$$A' \xrightarrow{1.0} hA',\quad A' \xrightarrow{1.0} rA',\quad A' \xrightarrow{0.9} e$$

$$B \xrightarrow{0.6} rvr,\quad B \xrightarrow{0.6} hll,\quad B \xrightarrow{0.6} hl,\quad B' \xrightarrow{0.6} \varepsilon.$$

{S})

### 3.6. String matching and Recognition

The methodology applied here for Handwritten Numeral Recognition is a two stage recognition technique, having the first stage as string matching with the help of Error-free Fuzzy context-free grammar and the second stage as with membership values of the string if there is match with more than one strings. Some of the earlier methods are described in [20].

The prototype Numeral generated using the method described in section 1 is decomposed into elements and labeled using Fuzzy logic. The labeled elements are combined and represented as strings. The string of each prototype character is stored in the database. The learned codes are used to classify the unknown handwritten numeral’s string into a class of the matching prototype. If there is any chance in getting two matching numeral then their membership values are compared for correct classification.

An algorithm has been developed for the string searching and recognition. This algorithm is quite efficient for the handwritten input characters considered for recognition.

***Algorithm 3: String Recognition Algorithm***

Input: Any handwritten character's string with membership value
Output: Classification/Recognition class of Numeral with membership value.

1. The binary search has been implemented for search of a string
2. In the search, the label-wise (character-wise) search is being performed.
3. If there is a match for two numerals then the membership values are compared.
4. Otherwise the correct numeral is declared.
5. If the membership values are also not matched with the existing, then the pattern is said to be misclassified or error.

## 4. RESULTS AND CONCLUSION

In this paper unconstrained handwritten numerals with invariant position and size are considered. The strings are obtained by considering the trace in clockwise direction. A 20 x 20 frame is used for writing alphabets. Prototype Numerals and Tamil characters are used to infer the grammar. Apart from the prototype generation module a set of handwritten Numerals were collected from various persons and the experiment has been conducted for testing purpose. The percentage of recognition varies from 91 to 99.5.

**Table 1**
**Result of the sample Numerals**

| *Numerals* | *String (worst case example)* | *No. of Contour* | *No. of Samples* | *% of re-cognition* | *% of Error or Mismatch* |
|---|---|---|---|---|---|
| 0 | *hrvlhrvl* | 2 | 220 | 92.2 | 7.8 |
| 1 | *rvr* | 1 | 212 | 99.5 | 0.5 |
| 2 | *hrvvvlhrvlhr* | 2 | 235 | 91.0 | 9.0 |
| 3 | *hhrvlhlrhrvlhrh* | 1 | 240 | 91.2 | 8.8 |
| 4 | *Vllhll* | 1 | 250 | 97.4 | 2.6 |
| 5 | *hlhrvvlhrv* | 2 | 243 | 95.5 | 4.5 |
| 6 | *lvrhlvhl* | 2 | 234 | 92.0 | 8.0 |
| 7 | *Hlvlv* | 1 | 237 | 95.0 | 5.0 |
| 8 | *hrvlvrhlvrv* | 3 | 255 | 94.0 | 6.0 |
| 9 | *hrvlllhrvl* | 2 | 218 | 91.0 | 9.0 |

**Table 2.2**
**Comparison of our Work with the Other Works found in the literature**

| *Methods* | *Recognition %* | *Rejection %* | *Training* | *Testing* |
|---|---|---|---|---|
| Ahmed et al. [2] | 87.85 | 7.25 | 5000 | 3540 |
| Cohen et al. [3] | 95.54 | 2.47 | | 2711 |
| Gader et al. [5] | 96.35 | 2.65 | | 6000 |
| Krzyzak et al. [10] | 86.40 | 12.60 | 4000 | 2000 |
| Lam et al. [12] | 93.10 | 3.95 | 4000 | 2007 |
| Le Cun et al. [13] | 90.00 | 9.00 | 7291 | 2000 |
| Stringa [23] | 92.60 | 2.80 | 19377 | 19377 |
| Suen et al. [24] | 93.05 | 6.95 | 4000 | 2000 |
| Suresh et al. [16] | 93.88 | 6.22 | 2500 | 2500 |

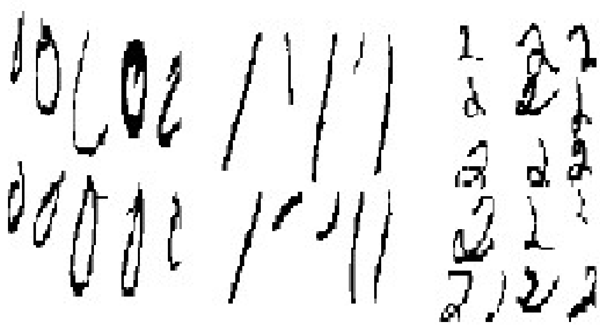


**Figure: (a) Numerals 0, 1 & 2**

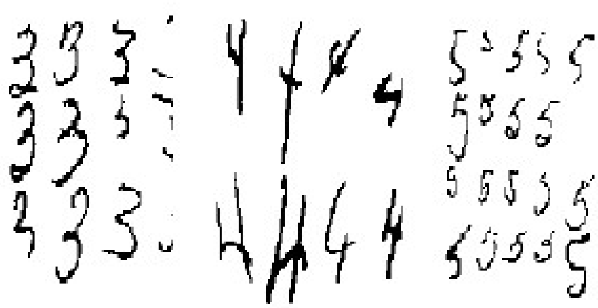


**Figure: (b) Numerals 3, 4 & 5**

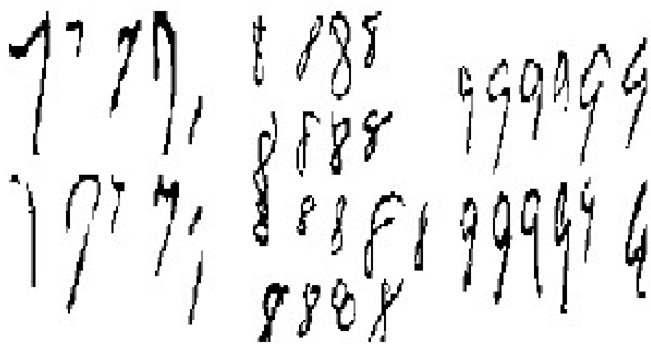


**Figure: (c) Numerals 7, 8 & 9**

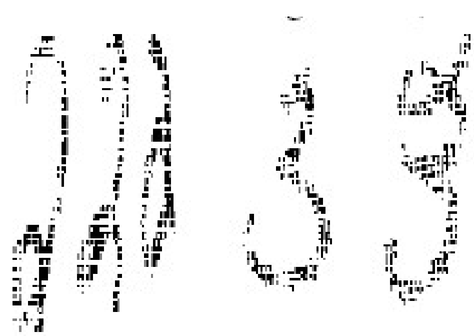


**Figure: (d) Numerals 2, & 3**

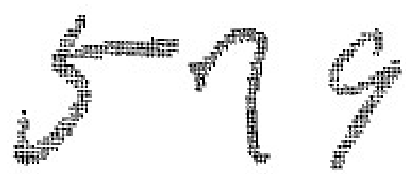


**Figure: (e) Numerals 5, 7 & 9**


## ACKNOWLEDGEMENT

The author wishes to express his sense of gratitude to **Shri. R.S. Munirathinam** Chairman, **Shri R. Jothi Naidu**, Director and **Dr. M.R. Jayateertha Rao**, Principal, ***RMK Engineering College***, *Kavaraipettai* for providing all types of support in all respect.

## REFERENCES


[1] Arcelli C., Pattern thinning by contour tracing, Comp. Graphics Image Process 17, pp. 130 144, 1981.

[2] Ahmed P. and C.Y. Suen, Computer recognition of totally unconstrained handwritten zip codes, Int. J. Patt. Reco. and Artificial Intell. **1**, pp 1-15, 1987.

[3] Cohen E. et al., Understanding handwritten text in a structured environment: Determining zip codes from address, Int. J. Pattern Recognition and Artificial Intelligence, **5**, pp. 221-264, 1991.

[4] Downton A.C. et al., Progress in Handwriting Recognition, World Scientific, Colchester, 1996.

[5] Gader D., et al., Pipelined system for recognition of handwritten digits in USPS zip codes, Proc. US postal service Advanced Technology Conf., Nov. pp. 539-546. 1990

[6] Gruska J, Some classifications of Context-Free Languages, Inf. and Cont. 14, pp. 152-179, 1969.

[7] Honda N and M.Nasu, Recognition of Fuzzy Languages, in Fuzzy Sets and their Applications Cognitive and Decision Processes, L.A. Zadeh, K.S. Fu, K.Tanaka and M.Shimura Ed., Netherlands: Kulwer Academics, pp. 279-299, 1974.

[8] Jamshidi M., Fuzzy logic Software and Hardware, in M. Jamshidi et al. (Eds), Fuzzy Logic and Control, Prentice-Hall, Englewood Cliffs, NJ, pp. 112-148, 1993.

[9] Klir G.J., and Bo Yuan, Fuzzy Sets and Fuzzy Logic Theory and Applications Prentice Hall of India, New Delhi 1997.

[10] Krzyzak, A, W. Dai, and C.Y. Suen, Unconstrained handwritten character classification using modified back propagation model, Proc. Int. workshop on Front. In Hand. Rec., Concordia University, Montreal, Apr. pp. 155-166, 1990.

[11] Kuroda K. et al, Large Scale on-line handwritten Chinese character recognition using successor method based on stochastic regular grammar, Patt. Recog. **32** pp. 1307-1315, 1999.

[12] Lam L. and C.Y. Suen, Structural classification and relaxation matching of totally unconstrained handwritten ZIP-code numbers, Pat. Reco. 21, pp.19-31, 1988.

[13] Le Cun et al., Constraint Neural Network for unconstrained handwritten digit recognition, Proc. Int. Workshop on Frontiers in Handwriting Recognition, Concordia University, Montreal, Apr, pp 145-154, 1990.

[14] Leung C.H., Y.S.Chenug ad Y.L.Wong, A knowledge based stroke-matching method for Chines Character recognition via neural networks, Pat.Reco. Let.7 no.1, pp. 19-25, 1988.

[15] Malaviya A. and L. Peters, Fuzzy Handwriting Description Language: FOHDEL, Patt. Reco. Vol. 33, pp. 119-131, 2000.

[16] Mantas J, An Overview of character Recognition Methodologies, Pattern Recog. , No. 6, pp. 425-430, 1986.

[17] Mohamed M. and P. Gader, Hand-written word recognition using segmentation free hidden Markov modeling and segmentation based dynamic programming technique, IEEE Trans. on Pattern Anal. and Machine Intell., **18**, pp. 548-554, 1996.

[18] Parizeau M.et al., A Fuzzy-syntactic approach to allograph modeling for cursive script recognition, IEEE PAMI, **17**, No. 7, pp. 707-712, 1995.

[19] Pedrycz W, Fuzzy Sets in Pattern Recognition Methodology and Methods, Patt. Recog, Vol 23, No. ½, pp. 121-146, 1990.

[20] R.M. Suresh, "Applications of Fuzzy Techniques to Pattern Recognition Problems", Ph.D. Thesis, M.S. University, Thirunelveli 2000.

[21] Shaw A.C., A Formal picture description scheme as a basis for picture processing systems, Info. Control, **14**, pp. 9-52, 1969.

[22] Siy P. and C.S. Chen, Fuzzy Logic for handwritten Numerical Character recognition, IEEE SMC 1,1, pp. 61-66, 1971.

[23] Stringa L., Efficient classification of totally unconstrained handwritten numerals with a trainable multilayer network, PRL, **10**, pp. 273-280, 1989.

[24] Suen C.Y. et al., Computer Recognition of unconstrained handwritten numerals, Invited paper, Special issue of Proceed. IEEE on OCR, **80**, No 7, pp. 1162-1180, 1992.

[25] Swonger C.W., An evaluation of character normalization, feature extraction and classification techniques for postal mail reading, Proc. Automatic Pattern Recognition, Washington, D.C., USA, May 6, 1969, pp. 67-87.

[26] Young Y., Handprinted Chinese character recognition via neural networks, Pat. Reco. Let. 7, no.1, pp.19-25, 1988.

[27] Zadeh L.A., Fuzzy sets, Inform. Control, **8**, pp. 338-353, June 1965.