World of Computer Science and Information Technology Journal (WCSIT)
ISSN: 2221-0741
Vol. 4, No. 4, 50-56, 2014

Arabic Digits Recognition Using Statistical Analysis for End/Conjunction Points and Fuzzy Logic for Pattern Recognition Techniques

Dr.Majdi Salameh
IT Department, King Abdul Azziz University

Abstract— Arabic Digits recognition has the lights spotted on lately, since it could be useful in a wild range of fields. This paper provides an easy and fast technique to recognizing Arabic digits. This paper presents two methods about enhancing recognition rate for typewritten Arabic digits (Hindi). First, is node method that calculates number of ends of the given shape and conjunction nodes as well, the second method is fuzzy logic for pattern recognition that studies each shape from the shape, and then classifies it into the numbers categories. Two stages are going to be done by the two given methods, to recognize the Arabic digit, each come out with its own result and afterward compounds these result to obtain the final solution and statistical analysis. Several steps are taken in the recognition system, starting with the image processing, then feature extraction and the last step is classification. The image processing includes converting into binary, cropping the digit in single image, and getting a skeleton of the shape by thinning it. Feature extraction includes number of terminal and conjunction nodes from nodes method and two characters to specify the curve lines group for shapes and third number to know the position of end nodes according to conjunction nodes in similar digit such as ٧ , ٨ . The recognition includes compound between two vectors, one from each method. The proposed technique was implemented and tested the experimental results give high recognition rate for some fonts and either less for other fonts because of due to irregularity of some fonts (Andalus) or failing for one of the methods. The dataset contains multi-size for the digits from ٠ to ٩.

Keywords-Arabic digit recognition; Arabic number recognition; node method; statistical analysis; multi size numeral recognition.

I. Introduction

Arabic language is widely used as more than billion people use Arabic in there life. Arabic language can mainly be under two categories: Arabic characters (أ.. ي) and numbers (٠ to ٩). Arabic are also two types: hand writing [6,7] and type written [15]. The improving of Arabic digits recognition can enhance the interact between the computer and human.[10] The easier and faster the recognitions perform, the more applications and benefits could be done several methods addressed the problem of recognition of digits depending on type of feature extract. There were many researches recognizing handwritten using different methods and techniques such as hand written digits recognition using two stage classification method [1] and hand written numeral recognition using fuzzy logic [2] which compos digits into lines, curves and circle also another research decomposed based on detection of set of feature points such as terminal , intersection, and bend points[3]. This paper focus on Arabic type written digits using hybrid of both classification method and detecting of set of feature points.

II. The Proposed Technique

This paper discusses the methodology of typewritten digits recognition technique, which will be used two proposed methods which lie in feature extraction step. The three steps of digits recognition are going to be described; each step is necessary and inevitable.

Various fonts are available these days, which added difficulty to recognize the typed digits, which make the feature extraction relatively hard where more font options are to be taken in consideration. Therefore feature extraction and classifying typing digits will be hard to deal with. However, for examination the capability for recognition in this study, 6 types of fonts are going to be studied for all digits from 0 to 9, such as, each image from type .bmp, and contains 10 digits, each digit describes specific different font type and size from 6 fonts.

٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩

Figure 1. Arabic digits

This paper proposed techniques is detailed in the following stages of the paper, while the following flow diagram shows all the basic stages in proposed techniques.

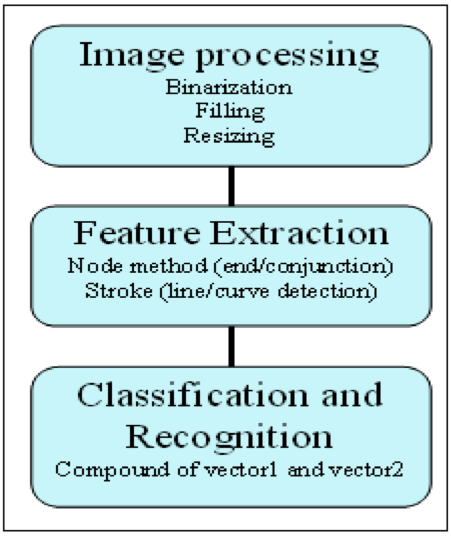

Figure 2. Flow diagram for proposed techniques for recognizing Arabic digits.

III. Image processing

The main idea of preprocessing is to segment the interesting pattern from the back ground.[19] In this step binarization, removing and filling the pixels should be done. Also the preprocessing defines a compact representation of the pattern, then acquiring the given image according to a set of characteristics such as image type “bmp” that contain set of digits as rows and columns, each row have a space in-between that is one line, and the digits are written in multi fonts – 6 fonts types – each digit in the imagine is a representation of a font type, and to determine the font types that be recognized in this study. This manner is going to be followed for numbers from 0 to9.

And then, we will be stratifying a set of operations on the image that aims to boost the shape of the digits, and to extract each digit in an single independent image. A set of morphological operations that will be applied on the input image to produce the final expected images. The next section of this paper will explain the steps of preprocessing by using an image containing 6 font types. This study had to deal with different size of fonts and any size of image. There are necessary operations to perform analysis for image prior to feature extraction and recognition of input image, the common operation as follows:

Binarization : is a method to reduce color image into two colors, black and white where only a single bit per pixel is required, the complexity is greatly reduced for the image, Matlab is used to read the image and convert it to black and white format. Digits recognition is color independent – a black digit is the same as the red digit [4]. First converting the image to grayscale, then the grayscale to binary image, threshold is determined by Otsu's method used in Matlab tools. The Matlab code to convert the input image to binary image.

Filling the pixels: Using square structure element to perform the binary dilation on binary image, the edge are being traces for each object that are on the image. The edge is dilated by using (imdilate) and square structure element 5x5, this step is one of the range of steps to define the digits from the given image, that each digit for example is isolated image with size 42x32. After performing dilation on the objects that are on the image, we fill the holes in the given image (a hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image).

Blobs Analysis In here, we are going to find all the objects in the given image and find the properties of each object toward delimitating the object.

Resizing: After determining all the blobs (Objects) by giving each object a label with unique index and surround it in a rectangle box. Then crop out the object in a single image. . Indexing of object and coordinating of bounding box will be used for cropping the object in the sub-image. Then, resize the sub-image (which contains isolated one digit) to standard size using resize method as follow

```
New_image_resized    =imresize(digit    image
,size)
```

IV. Feature extraction

For that each numeral is unique and have its own way, feature extraction is the important phase, thus distinguishing itself from other numerals.[ ] Hence, it's very important to extract features in such a way that the recognition of different numerals becomes easier on the basis of the individual feature of each numeral. Two proposed methods are taken in consideration in the feature extraction stage, a set of results is produced by each method and the correct output digit is one of them. Then a comparing step is taken of these two vectors of results to get the correct digit, where the intersection between two vectors occur in one element only, except duplicated states between,7and 8 . The duplicated states will treat with other technique (position of end point and intersected point). The advantages of these methods stem from flexibility to recognize several sizes of fonts, recognize many fonts with well recognition rate, and not needed to complex operation and calculations. Easy and fast way because no need to determine the entire shape; only the end and conjunction points.

Nodes method

The number endpoints of each digit will be calculated in this method. Node (or tip) is a pixel that has one or at most two neighbors concerning 8 pixel around it, from the experiment that we have tried, we reached to a conclusion, that recognizing a digit is possible if we know the number of nodes for that digit. We will use the number of nodes in the next step of this method to figure out the inputted image. The input image is sub-image with size 40x30. a thinning technique is going to be used in here, since its widely used to help recognizing the main shape (by giving it a main skeleton) , in our method we will use it in the pre-processing stage of pattern recognition system to compress data and to enhance feature extraction in the subsequent stage, and since it takes out a skeleton of the shape, all results going to be only 1 pixel thick, the thinning algorithm will be used in this study that can be found in MATLAB as the following [5]:

Thinned digit image =bwmorph(binary digit image, 'thin', 100)

Divide the image into two distinct subfields in a checkerboard pattern. In the first sub iteration, delete pixel p from the first subfield, if and only if, the conditions G1, G2, and G3 are all satisfied. In the second sub iteration, delete pixel p from the second subfield, if and only if, the conditions G1, G2, and G3' are all satisfied.

Condition G1:

$$X_H(p) = 1$$

Where

$$X_H(p) = \sum_{i=1}^{4} b_i$$

$$b_i = \begin{cases} 1, \text{ if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0, \text{ otherwise} \end{cases}$$

x1, x2, ..., x8 are the values of the eight neighbors of p, starting with the east neighbor and numbered in counter-clockwise order.

Condition G2:

$$2 \le \min\{n_1(p), n_2(p)\} \le 3$$

Where

$$n_1(p) = \sum_{k=1}^{4} x_{2k-1} \vee x_{2k}$$

$$n_2(p) = \sum_{k=1}^{4} x_{2k} \vee x_{2k+1}$$

Condition G3:

$$(x_2 \vee x_3 \vee \bar{x}_8) \wedge x_1 = 0$$

Condition G3':

$$(x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0$$

The number of nodes going to be calculated after the thinning step, And since each shape has a specific number of nodes, and in some cases the number might get repeated for that we used a large dataset that contained a several shapes for the same digit. For example, the number four rises in several shapes; each shape generates different number of nodes. To calculate the number of nodes, the following steps were taken:

- End points
- Conjunction points

Calculate the number of all nodes in the shape, endpoints, conjunction points, and then compare the total of nodes with the table 1 for producing the output vector, consisting of expected digits. The table 1 is prepared by experiential tests on dataset containing Arial font types as example.

TABLE 1: RESULTS FROM NODE METHOD FOR ALL NODE OF ARIAL FONT.

Number of total points	Elements
0	5
1	0
2	9 1
3	2 6 7 8
4	3
5	4

TABLE2: RESULTS FROM NODE METHOD FOR END NODE OF ARIAL FONT.

Number of end points	Elements
0	5
1	0 9
2	1 4
3	2 6 7 8
4	3

TABLE 3: RESULTS FROM NODE METHOD FOR CONJUNCTION NODE OF ARIAL FONT.

Number of conjunction points	Elements
0	0 1 5
1	2 6 7 8 9
2	3
3	4

TABLE 4: RESULTS FROM NODE METHOD FOR END AND CONJUNCTION POINTS NUMBERS P,(X,Y) OF ARIAL FONT.

Number	Elements of vector (p, x,y)
0	(1,0,1)
1	(2,0,2)
2	(3,1 ,4)
3	(4,2,6)
4	(2,3 ,5)
5	(0,0,0)
6	(3,1 ,4)
7	(3,1 ,4)
8	(3,1 ,4)
9	(1,1,2)

TABLE 5: END POINTS FOR ALL DIGITS WITH SAME SIZE(12)

	Conjunction points									
	0	1	2	3	4	5	6	7	8	9
Andalus	0	0	2	2	3	1	2	3	1	0
Arabic transparent	0	0	1	2	3	0	1	1	1	1
Simplified Arabic	0	0	1	2	3	0	1	1	1	1
Simplified fixed Arabic	0	0	1	2	3	0	1	1	1	1
Arial	0	0	1	2	3	0	1	1	1	1
Traditional Arabic	0	0	1	2	3	0	1	1	1	1

TABLE 6 : CONJUNCTION POINTS FOR ALL DIGITS WITH SAME SIZE(12)

	End points									
	0	1	2	3	4	5	6	7	8	9
Andalus	1	2	3	4	2	1	3	3	3	1
Arabic transparent	1	2	3	4	2	0	3	3	3	1
Simplified Arabic	1	2	3	4	2	0	3	3	3	1
Simplified fixed Arabic	1	2	3	4	2	0	3	3	3	1
Arial	1	2	3	4	2	0	3	3	3	1
Traditional Arabic	1	2	3	3	2	0	3	3	3	1

By contemplation in the table 1,2,3,4,5,6 we can observe 6 cases of nodes, which are concerned with five cases, and the last case (over 5 nodes) is excluded from this study, where those numbers which have over 5 nodes are irregular shapes, and will be needed for orientation with this study.

Figure 3. possible shapes for number 9.

Lines and curves detection

Each shape would represent a different digit that distinguishes it from others; an array will be created that contains the indexes for all the pixels in the given picture for each stroke of the shape.

Strokes determine: strokes is created by a set of all possibilities. In a vector will be stored each stoke extracted out of the character. Scanning the stroke will be from the left upper corner to the right down corner assign end points and the conjunction points.

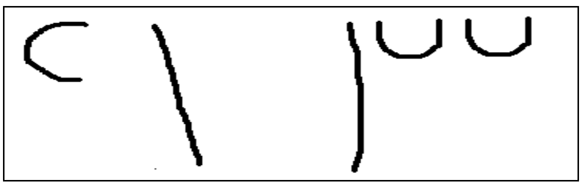

Figure 4. Shows the Strokes of numbers 9 and 3.

Lines determine: every stroke will be determined using tangent angle and mathematical operation and turn it into one kind of the lines (line, denoted as LL).

$$\begin{aligned}
MVL(x) &= 1 - \mid 1/m \mid fe && \text{if } |m| > 1 \\
&\quad 0 && \text{if } |m| <= 1 \\
MhL(x) &= 1 - \mid m \mid fe && \text{if } |m| < 1 \qquad (1) \\
&\quad 0 && \text{if } |m| >= 1 \\
Mob(x) &= 1 - \mid (\theta - 45)/45 \mid fe
\end{aligned}$$

Curves determine: The slop will be calculated for all the carves in order to determine in which direction the curve will be and to classify any in-between curves to the nearest curve.

Second: the following formula is used to determine the lines between the end and the beginning of the curve:

B *Middle Point of the curve Index order of element = (number of elements) /2 which denoted as (X3 ,Y3).*

$$M(a) = \Delta y / \Delta x = y2 - y1 / x2 - x1 \text{ , for any point on the line } (x,y)$$
$$\text{Let } (x, y) \text{ belong } A \text{ then}$$
$$Y - y1 = m (x - x1) , Y - m x = y1 - mx1$$
$$\text{Using } (x3, y3) \text{ in the pervious formula:}$$
$$\text{If } m(a) = 0 \text{ and } y3 < y1 \text{ then UC}$$
$$\text{If } m(a) = 0 \text{ and } y3 > y1 \text{ then DC}$$
$$\text{If } x1 = x2 \text{ or } \{m(a) > 1 \text{ or } M(A) < -1 \} \text{ and } X3 > x1 \text{ Then RC}$$
$$\text{If } x1 = x2 \text{ or } \{m(a) > 1 \text{ or } M(A) < -1 \} \text{ and } X3 < x1 \text{ then LC}$$

Third: if any element belongs to the line formula considering each curve, then it is a curve.

The following formula is going to be used to determine the curves that represents the strokes and will be denoted as (CC):

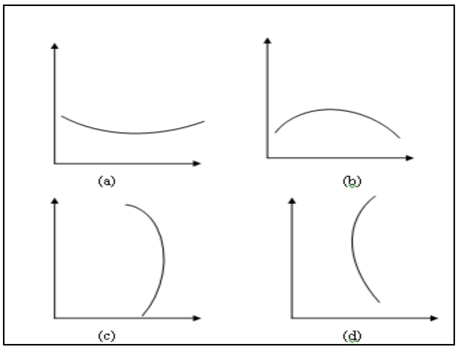

Figure 5. a,b,c,d Shows Basic Curves.

We will present all the digits by the following vector with four elements, first element is going to represent the number of end point in the digit, second element is for the number of conjunction point, third we are going to sum the total of end points and conjunction points of the digit, and the fourth element is for where the end points to the conjunction is located (1-above ,2-below,0-the same line), if more than one conjunction or no conjunction are found, the value will be N as null.

V1=(umber of endpoint , number of conjunction point, number of total points, number for position) from this method contains the elements which represents the probable digits.

TABLE 7: EXPECTED OUTPUT VECTORS OF SHAPE.

Characters of Shape	Output Vectors
1- LL (line)	1 4 6 7 8
2- CC (curves)	5
3- LC (both)	2 3 9

V. CLASSIFICATION AND RECOGNITION

Recognition system in this stage will use all features extracted from the shape of input digit to recognize and expect the correct number, the quality is the base of this stage where the strength of the output depends on, these features and methods used in the recognition system that we used. Two vectors are the output of the pervious two methods; each contains a set of elements. Thus, in the classification stage, those elements that we are going to match according to table 8 that shows all the possible result of all possible states, the dataset used in testing and experiment consists of 6 font type.

And to determine and obtain the wanted output, a comparing step between the two vectors are applied, the following steps are taken:

1. *Calculate the number of nodes for the shape in the given image (I) that will describe the expected elements in vectors of the nodes method, and there are 6 possibilities.*
2. *Starting form the right side of image (I) we will study the shape of the digit, we will use two characters to describe the shape of the digit in image (I), the two characters may be {LL,CC,LC } or in some cases "fails". Table 4 shows the elements of vector obtained from this method.*
3. *Then we will compound the two found vectors and union the elements that we found, to form the final vector.*

V3 =(number of endpoint(E), number of conjunction point (C), number of total points(T), number for position (P), stroke id number(S))

TABLE8: THE RESULTS FOR UNION THE TWO VECTORS.

Number	E	C	T	P	S
0	1	0	1	N	N
1	2	0	2	N	1
2	3	1	4	0	3
3	4	2	6	N	3
4	2	3	5	N	1
5	0	0	0	N	2
6	3	1	4	0	1
7	3	1	4	1	1
8	3	1	4	2	1
9	1	1	2	2	3

4. *From the previous steps, confusion states that we found needed to treated, that confusion occurred between the digits 6, 7, and 8. For that an extra operation are taken to clarify the confusion cases, the following table 8 present the adopted measures:*

Studying the position of the end point to the intersection point in the digits will help defer between the digits that caused confusion, which are 6, 7, and 8. if the end point above the intersection point then specific variable (4th parameter of the vector) will have the value of 1, if the endpoints below the conjunction points the value will be 2, if one of end point at the same level of conjunction then set to 0, otherwise set to N null. As shown in figure next.

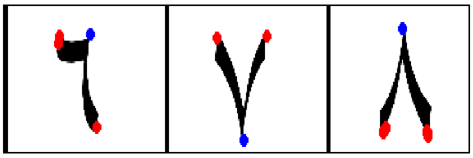

Figure 6. Duplicated case.

In case(6 7 8): if the return value is 1 then the number is 8, if the return value is 2 then the number is 7 , if the return value is 0 then the number is 6.

4. EXPERIMENTAL RESULTS

In this section the results retrieved through the application of the proposed technique are going to be presented as well as the analysis of the results. The illustration of dataset is used in the testing step. and the reason for the fonts are not being recognized or maybe having a low recognition rate, the recognition rate depends on the size of the given image and the type of the fonts that are used, and the efficiency of the method in the feature extraction, that shows the flexibility of these methods and how it can recognize the fonts with different sizes and fonts.

Image Data: The used dataset to evaluate the performance of the two proposed methods, is made up of set of images using bitmaps (.bmp Files) with range of sizes for the images, which the most important in this step is how to deal with isolated digits. This study assumes that the given image does not have any noise from the scanning or digital image, and to obtain the same results, we must observe the condition on the dataset.

Taking a lot of experiments to determine and describe the range of abilities for the proposed method in the step of recognition for the digit. Some of the results came out well, other was moderated, but some results failed. Table 9 shows the recognition level and clarifies number of recognition verse number of non-recognition fir 100 numerals. The observation from table 9; is that the number of recognition states that it's mostly larger than the number of non-recognition states. The errors of recognition results are due to error either in shape method or nodes method and some cases in both. Due to thinning of the shape; node error occur, shape which increases number of nodes or changed in number of nodes due to irregular shapes. The deformed shapes originally are caused by error reading for skeleton of the shape.

TABLE 9: RESULTS FOR RECOGNIZE 100 FONTS.

description / digits	Recognition Rate	Number of recognized	Number of non-recognize
1	100 %	100	0
2	92 %	92	8
3	94 %	94	6
4	100 %	100	0
5	88 %	88	12
6	94 %	94	6
7	100 %	100	0
8	100 %	100	0
9	94 %	94	6
0	100 %	100	0

In table 9 the recognition rate are calculated; by dividing the number of recognition state by the number of all states (96%). Each digit (from 0 to 9) has an image that contains 6 fonts. The font size is considered 12 for all the 100 fonts.

From the given results that are shown in table 9, we found out that some of the fonts have high recognition rate. The fonts recognized from 0 to 9; the recognition reach to 100%. Table 10 shows these fonts.

The recognition rate : is the number of recognized digits divided by 10.

TABLE10: DETAILS FOR RECOGNIZE 10 FONTS FROM 100 FONTS.

Description Fonts	Font Format	Font Size	Recognition Rate
Arial	Regular	12	100%
Arial	Bold	14	90%
Arabic transparent	Regular	12	100%
Arabic transparent	Bold	14	100%
simplified	Regular	12	100%
Simplified	Bold	14	100%
simplified fixed	Regular	12	100%
simplified fixed	Bold	14	100%
traditional Arabic	Regular	12	90%
traditional Arabic	Bold	14	90%
Andalus	Regular	12	50%
Andalus	Bold	14	50%
Overall	90%		

Some font recognition rate was lower that 50% and failed for some fonts, so, we extracted out the fonts which have the best rate of recognition, as well we found some fonts that have low recognition rate or in some cases digits are not recognized at all.

Overall, all these results are customary, because the original work did not have a 100% recognition result, plus the difference between the types of the fonts are massive. Table 10 shows the sizes for the fonts (12 or 14). Where the change might affect the rate of recognition for some fonts, for example, the font “Arial” got a 100% recognition rate with the font size 12, but if we used bold font, the rate recognition drops to 90%! The change in size or style type for some fonts doesn’t affect the recognition rate, because the shape and the taken skeleton of these fonts didn’t change through editing the size or the style of it.

The results for multi-sizes fonts: The experiment on several sizes of fonts was taken into account. Some cases of the results were good and not affected voluminously when the size changed, but in other cases, it's affected in different rates. Type of font and digit itself play a main role when changing the size of the font. For example, the digit "1" is not affected by size, where the results prove the recognition rate was 100% for sizes (12, 18, 24, 36, 54, and 72). Results for 6 fonts are described in table 11 below.

TABLE 11: RECOGNITION RATE FOR MULTI-SIZE

Types of fonts	Recognition rate
Arial	100%
Arabic transparent	100%
simplified	100%
simplified fixed	100%
traditional Arabic	90%
Andalus	50%

The rate for some reached to 100% or less than 50%. In some other font recognition rate; the result was low or failed, these fail and low cases, were either an error in studying the shape or in the number of nodes, are result of irregular shapes and disconnected edge of shapes that happen due to the difference of thinning results.

After studying the results in table 11 we found out that the results obtained from our technique on range of font sizes proved the flexibility for recognizing the multi size font. In principle for the two taken methods that have the ability to recognize a set of sizes, and we would attribute this ability into two reasons:

I. *The node method is detecting the number of nodes in tips (end points, conjunction points) of shape. Consequently, the tips for any digit mostly do not change if we change the size of the shape.*

II. *The skeleton shape is almost steady for any size of font. Therefore, we must expect good percentage recognition rate when using the two methods together.*

VI. CONCLUSIONS

The proposed technique for the digit recognition shows the ability to recognize the digits, as well as recognition of multi font numerals, the node method and feature extraction of the shape method were added and applied on large number of fonts.

Through studying and analyzing the experiments; results gained, we concluded that the proposed technique is suitable, and able to give a good result, and its qualified to recognize 100 fonts in 90% rate of recognition, and the recognize 90 multi fonts of different sizes varying from size 8 to 72. The shape study was utilized through a set of operations with low complexity; the classification stage relies on logical comparison operation.

The offered technique is flexible for change in size and type of fonts. The technique is simple and do not need much of calculations; that the “ if statement rules” is used to determine the wanted results, consequently, can use it in devices with small storage memory and low performing processor, and the ability to recognize irregular shape for some fonts.

FUTURE WORKS

Analysis for the method suggested to achieve by the details of problems faced by, and suggests some proposals to obtain results in the future, and to improve the efficiency this technique. Some of the proposals include increased rate of recognition and identify a significant number of fonts and sizes. Applied to the proposal of the proposed method to recognize numbers and letters handwritten isolated Arabic and English. The application of proposal and the proposed method to recognize the character of hand-written or printed. The promotion and development processing stage, in order to be able to recognize the shapes of irregular and deformed not affected with the change of scale.

ALGORITHM

Read image number
For each number Find conjunction and end nods
Store end node number as E, and conjunction numbers as C and total as T
Extract the shape for each number
Determine two letters for each shape(CC,LL,CL)
Let each group specify number (cc-1,ll-2,cl- 3) and store in S
Classify each number to its group number S
Create P as integer equal to 1 if numbers has conjunction node under the end node,2 if above, 0 same level, and null for more than one conjunction
Store set of parameters for each number
Number< E, C, T, P, S>
Compare new number vector with stored vector (find similarity)

REFERENCES

[1] J.s.lee o.j.kwon and s.y.bang,highly accurate recogntion of printed korean characters thrugh an improved two stage classification method, pattern recognition , vol. 32, no. 12 , 1999, pp.1935-1945.

[2] P.siy and c.s.chen fuzzy logic for handwritten numeral character recognition , IEEEtrans. System , klan and cybernetics,vol.4, 1974,pp.570-575

[3] G.baptista and k.m.kulkarni, a high accurance algorithm for recognition of hand written numerals, pattern recognition, vol.21, 1998, pp.287-291.

[4] Abhijit S. Pandya, Robert B. Macy, "Pattern recognition with neural networks in C++", CRC Press, SBN: 0849394627, Inc. Boca Raton, FL, USA ,1995.

[5] L.Lam, Seong-Whan Lee, and Y. Suen Ching, "Thinning Methodologies-A Comprehensive Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, No. 9, pp. 869-885, September 1992.

[6] C.Y. Suen, C. Nadal, R. Legault, T.A. Mai and L. Lam, "Computer recognition of unconstrined handwritten numerals", Proc. Of IEEE, vol. 80, no. 7, pp. 1162-1180, 1992.

[7] R. Plamondon, N. Srihari, "On-line and Off-line Hand writing Recognition: A Comperhensive Survey", IEEE Trans. On PAMI, vol. 22, no.1, pp. 63-84, January 2000.

[8] H. Ebrahimnezhad, GH. A. Montazer and N. Jafari, "Recognition of Persian numeral fonts by combining the entropy minimized fuzzifer and grammar", proceedings of the 6th WSEAS int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, February 16-9, 2007.

[9] Z.Chi, H.Yan,"ID3-Derived Fuzzy Rules and Optimized Defuzzification for Handwritten Numeral Recognition",IEEE transiction on fuzzy system,VOL, 4, February 1996.

[10] P Pyeoung Kee Kim," Improving Handwritten Recognition Using Fuzzy Logic", Pusan Women's University,Korea, 2004.

[11] N. Belkhamza, A. Akkouche, "Fuzzy Logic Character Recognition", Madinah College of Technology, Saudi Arabia, 2004.

[12] B.V.Dhandra, V.S.Malemath, H.Mallikarjun, and H. Mallikarjun, "Multi-font Numeral Recognition without Thinning based on Directional Density of Pixels", Gulbarga Univ., Digital Information Management, 1st IEEE International Conference, 2006.

[13] N.S.Arjun, G.Navaneetha, G.V.Preethi and T.K.Babu, "An approach to multi-font numeral recognition", TENCON 2007 - 2007 IEEE Region 10 Conference , Taipei , 2007 .

[14] Frank Y.Shih, "Image Processing and Pattern Recognition Fundamentals and techniques", John Whily and Sons, Directory of IEEE Book and Information services, pp.3-12, 2010.

[15] Aj Palkovic, "Improving Optical Character Recognition", Villanove University, CSRS, Vol. 4, No. 2, pp. 109-114, December 2001.

[16] N.S. Arjun, G. Navaneetha, G.V.Preethi, T.K.Babu "An approach to multi-font numeral recognition", TENCON 2007 - 2007 IEEE Region 10 Conference, pp. 1-4, Taipei, 2007.

[17] Frank Y.Shih, "Image processing and Mathematical morphology fundamentals and techniques", Wiley-IEEE, CRC Press, United State of America, pp. 4-17, 2009.

[18] R. C .Gonzalez, and Woods, "Digital image processing", second edition, New Jersey: Prentice Hall, pp. 34-69, 2002.

[19] W. Abu-Ulbeh, "New Algorithm to Noise Reduction using Morphological Operations", Msc. Thisis, Al Al-Bayt University, Mafraq, Jordan,2009.

[20] M. K. Al.bakkar, M. rababah ' A Hybrid Technique for Multifonts Numeral Recognition' al albyt university, jordan,master thesis, 2011.