Turkish Journal of Science & Technology
Volume 12(2), 71-77, 2017

Handwriting Character Recognision by using Fuzzy Logic

Enes VARDAR, Kaplan KAPLAN, H. Metin ERTUNÇ
Department of Mechatronic Engineering, Kocaeli University, Izmit / Kocaeli
kaplan.kaplan@kocaeli.edu.tr

(Received: 19.02.2017; Accepted: 05.04.2017)

Abstract

In Handwriting character recognision can be used to seek texts in big documents, take notes on tablet or decide whether or not internet user is a human or a computer in terms of Web security. In this study, a handwriting recognition system is studied by using fuzzy rules. The system includes 4 parts, namely image processing, feature extraction, fuzzification of the inputs, and defuzzification. In the first stage, image processing based on morphological operations are used to perform the handwriting recognisition under the same conditions. The feature extraction process is employed to find the total number of white pixels in each column. Then these pixel numbers are assigned to arrays. The next step is to find the local maximum and minimum values by considering this arrays as an increasing-decreasing mathematical function. Therefore, it is observed that the handwritten letters of these values are divided into various groups. In the next operation, fuzzy classification membership functions and rule tables of text groups are generated by using extracted feature data. For a better recognition perfromance, the letters group have to be known in order to use image fuzzy logic algorithm. Consequently, this group of letters was succesfully classified with fuzzy logic rules.

Keywords: Handwriting recognition, character recognisition, fuzzy logic approach, image processing algorithms.

Bulanık Mantık ile El Yazısı Tanıma

Özet

El yazısı karakter tanıma, büyük bel gelerde metinleri araştırmak, tablet üzerinde not almak veya İnternet kullanıcısının bir insan veya bir bilgisayar olup olmadığını Web güvenliği açısından karar vermek için kullanılabilmektedir. Bu çalışmada, el yazısının tanınmasında bulanık kuralların kullanıldığı bir sistem incelenmektedir. Sistem görüntü işleme, özellik çıkarma, verilerin bulanıklaştırılması ve bulanık çıkarım olmak üzere 4 bölüme içermektedir. İlk aşamada, morfolojik işlemlere dayalı görüntü işleme, aynı koşullar altında el yazısı tanımayı gerçekleştirmek için kullanılmaktadır. Özellik çıkarma işlemi ise her sütunundaki toplam beyaz piksel sayısını bulmak için kullanılmıştır. Bulunan toplam piksel sayıları okunan sütun sırası ile dizilerde saklanır. Sonraki işlem ise bu diziyi artan-azalan bir matematiksel fonksiyon olarak göz önünden bulundurup, yerel maksimum ve minimum değerlerini bulmaktır. Böylece, bulunan bu değerlerden oluşan el yazısı harflerinin çeşitli gruplara ayrıldığı görülmüştür. Sonraki adımda çıkarılan özellik verileri kullanılarak harf gruplarının bulanık kümeleri üyelik fonksiyonları ve kural tabloları oluşturulmuştur. Daha iyi bir tanuıma işlemi için, görüntüye bulanık mantık algoritması uygulanmadan önce hangi harf grubunda olduğu bulunması gerekmektedir. Daha sonra bu harf grubları bulanık mantık kuralları ile başarıyla sınıflandırılmıştır.

Anahtar Kelimeler: El yazısı tanıma, karakter tanıma, bulanık mantık yaklaşımı, görüntü işleme algortimalar

1. Introduction

Nowadays, studies on handwriting recognition have found numerous application area in technological development. Handwriting recognition systems can greatly facilitate human life by speeding up many operations such as reading tax statements, directing mail, reading bank checks and so on and so forth. On the other hands, these systems may reduce the need for human interaction. Therefore, the academicals and commercial studies about handwriting characters have recently become an important research topic in pattern recognition. The human handwritings have been tried to be recognized by many artificial intelligence methods such as Artificial Neural Networks (ANN) [1], K-Nearest Neighbors (KNN) [2], and Linear Differential Analysis (LDA) [3]. One of the methods in application is Fuzzy Logic approach. Fuzzy logic is very suitable for handwriting recognition systems because of its capability for processing of uncertain data. The difficulty in handwriting recognition systems is that handwriting characteristic varies greatly from person to person. For this reason, handwriting recognition systems are quite complex.

In this study, we have studied on the set of images containing big handwritten letters with straight lines. Image processing and feature extraction algorithms in the system were coded in C++ environment. Fuzzy logic algorithms have been constructed and tested in MATLAB environment for handwritten letters. Once the fuzzy logic algorithms were tested in MATLAB to check the recognition performance, then they are written in C++. In the final stage, all the algorithms are combined in a single software, and, therefore handwritten characters are recognized with a reliable performance.

2. Materyal ve Metot

2.1. Image processing

Most of the images require preliminary image processing algorithms before applying any recognition technique. In order to increase the accuracy of the classification, it is very important to include descriptive features when extracting features. The steps of image processing algorithms used in this study are explained as in the following subsections.

2.1.1. Convert the image into a binary image

The image to be classified in this section is converted from RGB space to gray space by using *OpenCV* in C ++ environment. The grayscale histogram density is automatically calculated and converted to a binary image with the applied threshold value. In the obtained binary image, all the pixel values above the threshold value are assigned to 1 (white) value and all the pixel values below the threshold value are assigned to 0 (black) value. Thus, the image to be classified is divided into two parts that are 1 and 0 with the threshold value. Since feature extraction is related to the number of pixels that have value of 1, the division of parts must be done properly. For this, the Otsu Method, which has a library on OpenCV that automatically calculates the threshold value, is used. This method assumes that the image is composed of only two color classes, namely, the background and the foreground. The variance values for all threshold values for these two color class is calculated. The threshold value, which has the smallest variance value, is determined as the optimum threshold value

2.1.2. Finding more than one character in the image

In this section, binary image applied with threshold value is used as input. The binary image can have more than one character. Firstly, the characters must be separated from the whole image and considered as single image, and each process referred to in this work must be applied. As a first step, the boundary lines is found by the *findContours* function in OpenCV library [7]. The boundary lines are stored in a array with two memory files. The first memory of the array is the object and the second is the point of the object boundary line. The object boundaries are dropped to an array by using a simple for loop. Then, the smallest rectangles surrounding these boundary lines are found with the help of OpenCV rectangle as in Fig. 1 [7]. This rectangular image is cropped sequentially from the left of the image, and each cropped rectangle is regarded as an image to be classified.

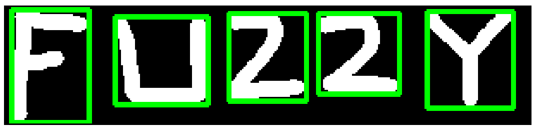


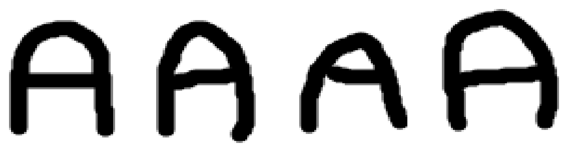
Figure 1. The handwriting image divided into character groups

2.1.3. Image standardization

The last step of image processing procedure is the size standardization of the image characters. Since character images are in different sizes from each other, they need to be reduced to the same size for comparison and recognition of images adequately. By means of this algorithm, the images to be classified in different sizes are standardized to (40 × 30) pixel dimensions.

2.1.4. Feature extraction

In feature extraction stage, the approach is to scan all the columns of the digitized imager respectively and keep the total number of white pixels in each column in a string form. Then classification of the characters is performed by comparing the total number of white pixels in this column. Figure 2 shows the *A* letters drawn with different handwriting and Table 1 lists the column pixel data after image processing procedure is applied to these characters. Note that the pixel values of the characters are different according to each column in Table 1. The main reason of the characters' difference is that they are written by different handwritings.



(S_1) (S_2) (S_3) (S_4)

Figure 2. The characters '*A*' written by different handwriting

As the columns are observed carefully, even if the characters looks like in the form of a different character, the Table 1 gives a hint about classification sheme. In all four cases, the coloumn data shows that the values of pixels increase to nearly 21, decrease to nearly 11, increase nearly 39 again, and finally decrease to 0, respectively. For this reason, a mathematical function can be evaluated according to the local maximum and local minimum values of the data in Table 1. The difference between the current local maximum (or minimum) and the previous local minimum (or maximum) gives us the transition values. These transition values will form the inputs of the fuzzy system. It is possible to verbalize these pixel values in the column with fuzzy logic. Then, the characters can be defined by these verbal statements with fuzzy rules. It is important to determine transition values correctly in terms of accuracy of system.

Step 1 (Finding extreme values): In this process, the program takes into account the data in the column and compares it with the previous data set in order to see whether or not it extracts a local maximum or local minimum point. Figure 3 shows the flow diagram of the algorithm for computing the extremum (transition) values written in C++. In Table 2, the transition values are listed for the letter '*A*' with this block diagram. In the diagram, CD is the current state data and PD is the data in the previous column. Outputs of this algorithm is a transition number and a transition value associated with it. For instance, the parameters $T_1 = 23$ indicates that the transition number is 1 and the transition size is 23. The algorithm also includes determining directional change. When a directional change (transition value) is determined, there must be more difference than ±3 value between CD and PD. Negative DIT (-DIT) (Direction of transition) means incrementation and positive DIT (+DIT) means the decrementation. If there is a difference equal to 2 or less between CD and PD, the big one is selected. If CD is big, the direction sign is considered. If the direction sign is positive, then L_max is equal to CD, otherwise the algorithm will search the next column. If PD is big and the direction sign is negative, then L_min becomes as CD. If the direction sign is positive, the other column is passed.

Table 1. S_1, S_2, S_3 and S_4 pixel data

Coloumn	S_1	S_2	S_3	S_4
1	5	8	0	0
2	14	14	9	11
3	20	19	14	17
4	23	16	17	24
5	22	16	20	19
6	15	16	17	15
7	13	14	14	13
8	11	13	12	12
9	12	14	13	13
10	13	14	14	13
11	13	14	15	12
12	14	15	32	12
13	14	32	32	11
14	14	38	30	12
15	38	39	29	13
16	39	22	25	38
17	39	16	13	40
18	14	13	13	40
19	14	14	13	27
20	14	14	13	13
21	14	14	14	13
22	14	14	14	12
23	14	14	13	12
24	14	14	13	12
25	14	14	13	12
26	14	13	12	13
27	14	13	12	12
28	12	10	10	11
29	8	5	5	9
30	0	0	0	0

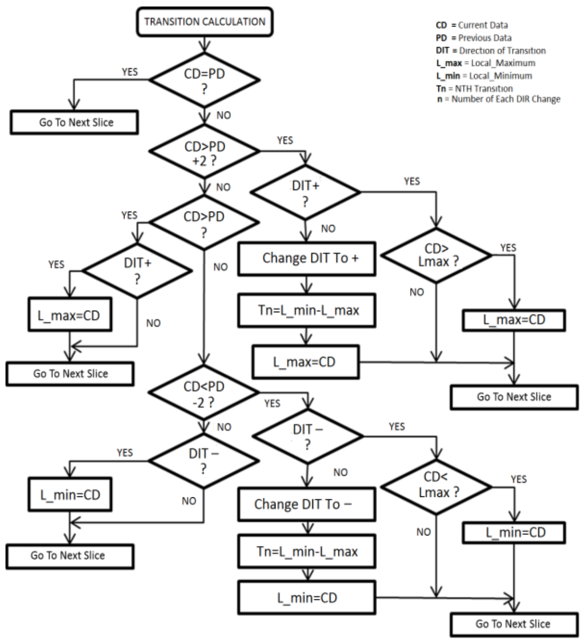


Figure 3. Transition value calculation algorithm

The transition values of the character A (S_1) found using the algorithm in Fig. 3 are shown in Table 2. As shown in Table 2, there are 4 transition points of character A ($T_1 = 23$, $T_2 = –12$, $T_3 = 28$, $T_4 = –39$). The feature extraction in order to characterize each character is applied to all the letters. In feature extraction operations, characters are divided into 3 character sets with 2, 4, or 6 transition values. B, E, G, S have six transition values; A, C, F, J, O, P, R, Z, D have four transition values and H, I, K, L, M, N, T, U, V, Y letters have two transition values character sets, respectively. Firstly, the character to be recognized is determined by which of these letter class, and so this operation facilitates decision of classes with and also flow of the system. These transition values can not provide sufficient accuracy in each character set in order to recognize the characters. For this reason, the total number of pixels, the first maximum and minimum column number, and the characteristic length specific for the character set are extracted by the software to increase the accuracy of the program.

Step 2 (Finding total pixel data): In this process, the pixel of images that are standardized and considered to be drawn in certain dimensions are scanned in C++ in order to find the total number of white pixels. The total number of pixels is not used because it does not vary much in the class which has only 4 transition values. The total number of pixels found is written in the fuzzy logic rule table and in the membership functions with TPS abbreviation.

Step 3 (Finding the first maximum and minimum points): It was observed that the column numbers in which the first and maximum transition values were found during the observation of the data differ between the letters and that the same letter has close values even in different handwritten letters. The first maximum and minimum column numbers found differ in letter sets with 2 transition and 4 transition values. They are not used since these values do not vary sufficiently in the 6-pass value set.

Table 2. Display of transition variables

Coloumn	CR	PR	DIR	T	L_min	L_max	Tn
0	0	0	+		0	0	0
1	5	0	+		0	5	0
2	14	5	+		0	14	0
3	20	14	+		0	20	0
4	23	20	+		0	23	0
5	22	23	+		0	23	0
6	15	22	-	T_1	15	23	23
7	13	15	-		13	23	0
8	11	13	-		11	23	0
9	12	11	-		11	23	0
10	13	12	-		11	23	0
11	13	13	-		11	23	0
12	14	13	-		11	23	0
13	14	14	-		11	23	0
14	14	14	-		11	23	0
15	38	14	+	T_2	11	23	-12
16	39	38	+		11	39	0
17	39	39	+		11	39	0
18	14	39	-	T_3	11	39	28
19	14	14	-		14	39	0
20	14	14	-		14	39	0
21	14	14	-		14	39	0
22	14	14	-		14	39	0
23	14	14	-		14	39	0
24	14	14	-		14	39	0
25	14	14	-		14	39	0
26	14	14	-		14	39	0
27	14	14	-		14	39	0
27	12	14	-		14	39	0
29	8	12	-		8	39	0
30	0	8	-	T_4	0	39	-39

Step 3 (Feature extraction of a set of letters with four transition values): In this process, the character lengths of this set of letters shown in Fig. 4 are determined to increase the accuracy of the system. The logic of the written algorithm is to count the number of black pixels remaining between the first white pixel and the next white

pixel. To find the length between the vertical and horizontol red lines, the scan operation is performed from (20,0) to (20,30), and from (0,28) to (40,28), respectively. The accuracy of recognizing letters has been increased thanks to differences in letters (C, F, J, P, Z) that do not have horizontal length (HU), and the letters (C, F, J, P, Z) do not have vertical length (VU).

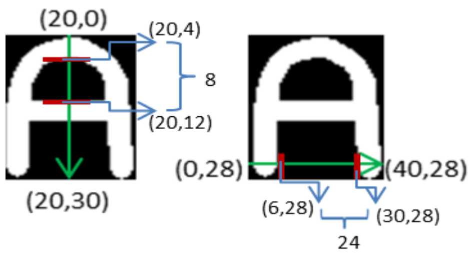


Figure 4. The characteristic length extraction

Step 4 (Feature extraction of a set of letters with two transition values): Characteristic length extraction in the case of two transition values is to determine the length between the horizontal and vertical red lines shown in Fig. 5. The reason for obtaining these lengths is the same as in the previous feature extraction process. The difference only in the scanning process is that the scanning direction is from (33,0) to (33,30) for the vertical lengths, from (12,3) to (28,3) for horizontal lengths. In this letter set, only the letter K is recognized by using fuzzy logic with the length of the letter K in the vertical direction, the differences in the horizontal length between the letters with the transition values, the first maximum column number, and the total pixel values.

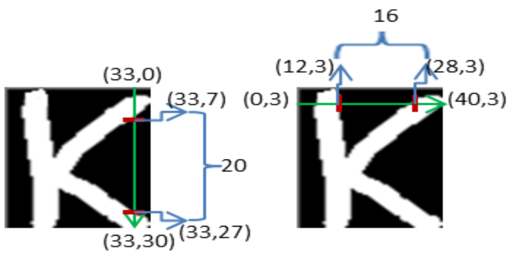


Figure 5. The characteristic length extraction

2.1.5. Fuzzification of data

Fuzzification is the process of converting information received from the system into symbolic values, which are linguistic qualifiers. By taking advantage of the membership process, it determines the fuzzy set and membership level to which the input information belongs, and assigns numerical values to the entered numerical values such as “small, smallest”. In order to determine each verbal expression, firstly the intervals where the values of the data change must be found. In this study, many different handwritten characters have been evaluated for each character; feature data were examined, and variation intervals of values were found. Each variable interval is divided into sub-regions and each sub-region is labeled with verbal expression. The rule tables of each character group are determined via the MATLAB FIS Editor interface by using the variable data found in C++ software in order to find the membership level of these verbal expressions. Then the variable data are labeled as verbal expressions; fuzzy clusters and membership functions are created in the most accurate way by trial and error method. The fuzzy clusters and membership functions that have been verified by MATLAB program are created in C++ and integrated into image processing and fuzzy logic software. The fuzzy sets in the system consist of 3 membership functions that are small (S), medium (M) and large (L). These membership functions are defined as triangular and trapezoidal functions as shown in Fig. 6. The trapezoid membership function consists of 4 points, n_1, n_2, n_3 and n_4, and the triangular membership function consists of 3 points, n_1, n_2 ve n_3. Points of triangular and trapezoidal functions of all letter sets are given in Table 3, Table 4 and Table 5.

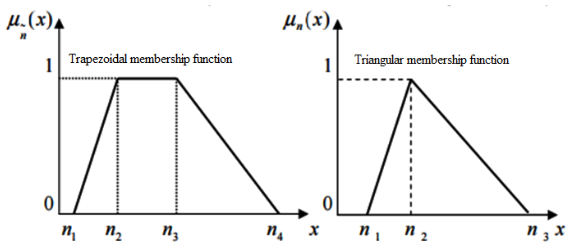


Figure 6. Trapezoidal and triangular membership functions

Table 3. Membership functions of letters with 6 transition values

	T_1	T_2	T_3	T_4	T_5	T_6	TPS
S	[0 0 23 30]	[0 0 17 23]	[0 0 15 22]	[0 0 15 22]	[0 0 15 22]	[0 0 20 30]	[0,300,449,500]
M	[23 30 40 40]	[17 23 30]	[15 22 25 30]	[15 22 25 30]	[15 22 27 30]	[20 30 40 40]	[449 500 536 600]
L	------------	[23 30 40 40]	[25 30 40 40]	[25 30 40 40]	[27 30 40 40]	------------	[536 600 775 800]

Table 4. Membership functions of letters with 4 transition values

	T_1	T_2	T_3	T_4	HU	VU	Min_1
S	[0 0 20 31]	[0 0 15 20]	[0 0 14 23,5]	[0 0 22 33]	[0 0 2 3]	[0 0 18 20]	[0 0 10 18]
M	[20 31 40 40]	[15 20 25 30]	[14 23,5 25 30]	[22 33 40 40]	[2 3 40 40]	[18 20 40 40]	[10 18 30 30]
L	------------	[25 30 40 40]	[25 30 40 40]	------------	------------	------------	------------

Table 5. Membership functions of letters with 2 transition values

	T_1	T_2	VU	HU	Max1	TPS
S	[0 0 15 20]	[0 0 14 20]	[0 0 2 5]	[0 0 15 20]	[0 0 5 10]	[0 0 300 400]
M	[15 20 24 30]	[14 20 24 30]	[2 5 25 40]	[15 20 40 40]	[5 10 13 21]	[300 400 500 600]
L	[24 30 40 40]	[24 30 40 40	[25 30 40 40]	------------	[13 21 25 40]	[500 600 700 800]

Table 6. Fuzzy rule tables of letters with 6 transition values

Letter	T_1	T_2	T_3	T_4	T_5	T_6	TPS
S	Medium	Medium	Small	Small	Medium	Medium	Small
B	Medium	Small	Small	Small	Medium	Medium	Large
G	Medium	Medium	Large	Medium	Small	Medium	Medium
E	Medium	Medium	Medium	Medium	Large	Medium	Medium

Table 7. Fuzzy rule tables of letters with 4 transition values

Letter	T_1	T_2	T_3	T_4	HU	VU	Min_1
A	Medium	Small	Medium	Large	Medium	Small	Small
J	Large	Large	Medium	Medium	Small	Medium	Medium
F	Large	Large	Medium	Large	Small	Small	Small
O	Medium	Small	Small	Medium	Medium	Medium	Medium
P	Large	Small	Small	Large	Small	Small	Small
R	Medium	Small	Small	Large	Medium	Small	Small
Z	Large	Medium	Large	Large	Small	Small	Medium
C	Medium	Medium	Medium	Medium	Small	Medium	Medium
D	Medium	Small	Small	Medium	Small	Medium	Medium

Table 8. Fuzzy rule tables of letters with 2 transition values

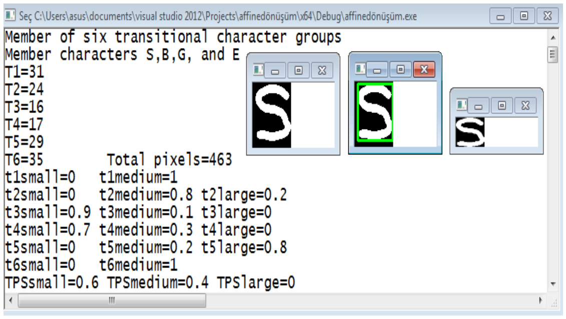
Letter	T_1	T_2	VU	HU	Max_1	TPS
H	Large	Large	Small	Medium	Medium	Medium
I	Large	Large	Small	Small	Large	Large
K	Medium	Medium	Medium	Small	Medium	Medium
M	Large	Large	Small	Small	Medium	Large
N	Medium	Medium	Small	Medium	Large	Medium
T	Large	Large	Small	Small	Small	Small
U	Large	Large	Small	Medium	Large	Medium
V	Small	Small	Small	Medium	Large	Small
Y	Small	Small	Small	Medium	Medium	Small
L	Large	Large	Small	Small	Large	Small

2.1.6. Fuzzy inference

Fuzzy inference is the process and inference of the fuzzy concepts in a way similar to the ability of people to make decisions and make inference. In fuzzy inference, there are several methods such as min-max, max-prod and Tsukamoto. Min-max was used in this study. The number of transition values of the letter and the letter set given above will be evaluated according to in the rule table. In the rule table, the number of the transition values of the image to be recognized and the character set of letters given above is found. For each letter rule in the rule table, the membership function value of the fuzzy set which the letter is the member is found, respectively. The minimum values of the membership degree of each rule found are stored in an array. The maximum value obtained between these minimum ratings specifies which letter is the image.

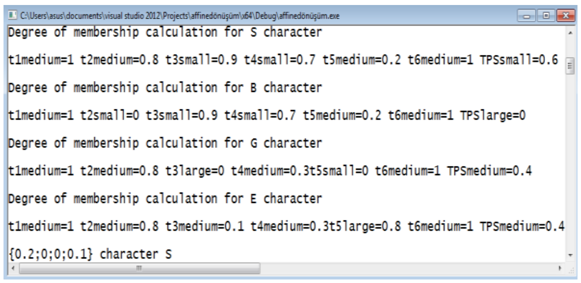
3. Testing of Handwriting Recognition System

Figure 7 shows the output of the program with the image of the letter *S* handwritten to be classified. The letter *S* image was first converted to binary image by thresholding method, and then the boundary points were found in the OpenCV library. The character to be recognized is regarded as a separate image by the smallest rectangle formed by these boundary points, and this image has been translated into (40 × 30) pixel dimensions for standardization data in the program. As shown in the program output, this character is a member of a set of letters with 6 transition value. The fuzzy logic inputs in this set of letters are the transition values and the total number of pixels. The total pixel values and transition values found and the membership degree of the fuzzy clusters are calculated. In the next step, the membership degrees as shown in Figure 8 are placed in each letter rule table as in Table 6. The minimum membership degree between the membership degrees listed in rule table is found. As shown in Figure 8, *S* character has 0.2, *B* character has 0, the character *G* has 0 and the character *E* has a membership rate of 0.1. based on taking into account the maximum membership level between these minimum membership degree, it can be determined which letter rule is the Max degree member of the image to be classified. As seen from the Min values, this character belongs to two rules. The letter *E* is a member with a membership level of 0.14 and the letter *S* is also a member with a membership level of 0.2. As mentioned before, the maximum value (0,2) between these two membership levels indicate the letter *S*.


Seç C:\Users\asus\documents\visual studio 2012\Projects\affinedönüşüm\x64\Debug\affinedönüşüm.exe

```
Member of six transitional character groups
Member characters S,B,G, and E
T1=31
T2=24
T3=16
T4=17
T5=29
T6=35        Total pixels=463
t1small=0   t1medium=1
t2small=0   t2medium=0.8 t2large=0.2
t3small=0.9 t3medium=0.1 t3large=0
t4small=0.7 t4medium=0.3 t4large=0
t5small=0   t5medium=0.2 t5large=0.8
t6small=0   t6medium=1
TPSsmall=0.6 TPSmedium=0.4 TPSlarge=0
```

Figure 7. C ++ program output


C:\Users\asus\documents\visual studio 2012\Projects\affinedönüşüm\x64\Debug\affinedönüşüm.exe

```
Degree of membership calculation for S character

t1medium=1 t2medium=0.8 t3small=0.9 t4small=0.7 t5medium=0.2 t6medium=1 TPSsmall=0.6

Degree of membership calculation for B character

t1medium=1 t2small=0 t3small=0.9 t4small=0.7 t5medium=0.2 t6medium=1 TPSlarge=0

Degree of membership calculation for G character

t1medium=1 t2medium=0.8 t3large=0 t4medium=0.3t5small=0 t6medium=1 TPSmedium=0.4

Degree of membership calculation for E character

t1medium=1 t2medium=0.8 t3medium=0.1 t4medium=0.3t5large=0.8 t6medium=1 TPSmedium=0.4

{0.2;0;0;0.1} character S
```

Figure 8. C ++ program output

4. Conclusion

In this study, a handwriting recognition system is realized by using image processing, feature extraction and fuzzy logic algorithms in C++ environment. Character images written in different handwriting, image processing algorithms, feature extraction are examined. The accuracy of the system has been increased by examining more than one samples for each letter. As a result of this examination, it is found that the same data in each letter does not vary, that each letter does not have the same number of variable data numbers, and that each letter does not contain the same variable data. Therefore, letters are separated into character groups and fuzzy logic algorithm is created by using variable data of each group of letters. Handwritten letters are succesfully classified by using fuzzy logic algorithm. In this study, the dotted letters (ö, ü, i) were not classified due to the uncertainties of the data. In future work, the authors will be concentrated on a recognition system based on more advanced intelligent algortihms such as artificial neural networks, neuro-fuzzy inference system and support vector machines.

5. References

1. Erdem O. A., Uzun E. (2005). Turkish Times New Roman, Arial, And Handwriting Characters Recognition By Neural Network: Journal of the Faculty of Engineering and Architecture of Gazi University, Ankara, 20: 13-19.

2. Weijie S., Jin X. (2011). Hidden Markov Model with Parameter-Optimized K-Means Clustering for Handwriting Recognition. IEE E2011 International Conference on Internet Computing & Information Services (ICICIS), Hong Kong, 235-438.

3. Prasad, M. M., Sukumar M. (2013). 2D-LDA based online handwritten kannada character recognition. Int. Jl. of Computer Science and Telecommunications **4**(1): 14-18.

4. Jasim M. K., Al-Saleh A. M., Aljanaby A. (2013). A Fuzzy Based Feature Extraction Approach for Handwritten Characters. International Journal of Computer Science Issues (IJCSI). **10**: 208-2015.

5. Gowan W. A. (1995). Optical character recognition using fuzzy logic. Microprocessors and Microsystems, **19**: 423-434.