Journal of ELECTRICAL ENGINEERING, VOL. 57, NO. 5, 2006, 258–267

TEXT DETECTION AND CHARACTER RECOGNITION USING FUZZY IMAGE PROCESSING

Mohanad Alata — Mohammad Al-Shabi *

The current investigation presents an algorithm and software to detect and recognize character in an image. Three types of fonts were under investigation, namely, case (I): Verdana, case (II): Arial and case (III): Lucida Console. The font size will be within the range of 17–29 font size. These types were chosen because the characters have low variance and there is less redundancy in the single character. Also, they have no breakpoints in the single character or merge in group of characters as noticed in Times New Roman type. The proposed algorithm assumed that there are at least three characters of same color in a single line, the character is on its best view which means no broken point in a single character and no merge between group of characters and at last, a single character has only one color. The basic algorithm is to use the 8-connected component to binirize the image, then to find the characters in the image and recognize them. Comparing this method with other methods, we note that the main difference is in the binarizing technique. Previous work depends mainly on histogram shape. They calculate the histogram, and then smooth it to find the threshold points. These methods are not working well when the text and background colors are very similar, and also if it may create an area of negative text. The negative text may be treated as a graphic region which may affect the system efficiency. The presented approach will take each color alone. This will make the merge between the text and the background unlikely to happen. Also there will be no negative text in the whole image because the negative text to a color will be normal text for another. The shown and discussed results will show the efficiency of the proposed algorithm and how it is different compared with other algorithms.

K e y w o r d s: text detection, character recognition, fuzzy image processing, optical character recognition

1 INTRODUCTION

Machine replication of human functions, like reading, is an ancient dream. However, over the last five decades, machine reading has grown from a dream to reality. Text detection and character recognition, which is known as Optical Character Recognition (OCR) has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence. Many commercial systems for OCR exist for a variety of applications.

1.1 Introduction to image processing

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows. In an (8-bit) grayscale image each picture element has an assigned intensity that ranges from 0 to 255. A grayscale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of gray. A normal grayscale image has 8 bit color depth = 256 grayscales. A "true color" image has 24 bit color depth = $8\times8\times8$ bits = $256\times256\times256$ colors =$\sim$ 16 million colors. Some grayscale images have more grayscales, for instance 16 bit = 65536 grayscales. There are two general groups of 'images': vector graphics (or line art) and bitmaps (pixel-based or 'images').

1.2 Literature Review

There has been a growing interest in the development of methods for detecting, localizing and segmenting text from images and video. Here we present the literature on detection, localization, and extraction of text regions from images and video.

Q. Yuan, C.L. Tan [1] presented a method that makes use of edge information to extract textual blocks from gray scale document images. M. Pietikäinen and O. Okun [2] proposed a simple method based on edge detectors, such as the Sobel operator, for document image filtering and text extraction. S.J. Perantonis, B. Gatos and V. Maragos [3] presented a novel Web image processing algorithm for text area identification. Wei-Yuan Chen and Shu-Yuan Chen [4] developed a page segmentation algorithm to locate text blocks for a journal's cover page, where texts may be overlaid onto graphics or images. C. Strouthopoulos and N. Papamarkos [5] proposed a new method to automatically detect and extract text in mixed type color documents. The proposed method was based on a combination of an Adaptive Color Reduction (ACR) technique and a Page Layout Analysis (PLA) approach. K. Atul Negi, Nikhil Shanker and Chandra Kanth Chereddi [6] presented a system to locate, extract and recognize Telugu text. The circular nature of Telugu script was exploited for segmenting text regions using the Hough Transform. Shoichiro Hara [8] proposed new means of separating cursive characters into

* Mechanical Engineering Department, Jordan University of Science and Technology, Irbid P.O. Box (3030), Jordan

ISSN 1335-3632

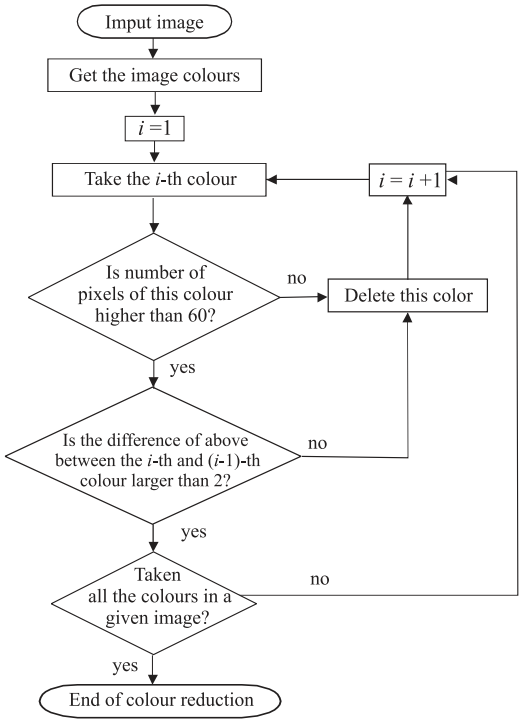

Fig. 1. Reducing colors flowchart

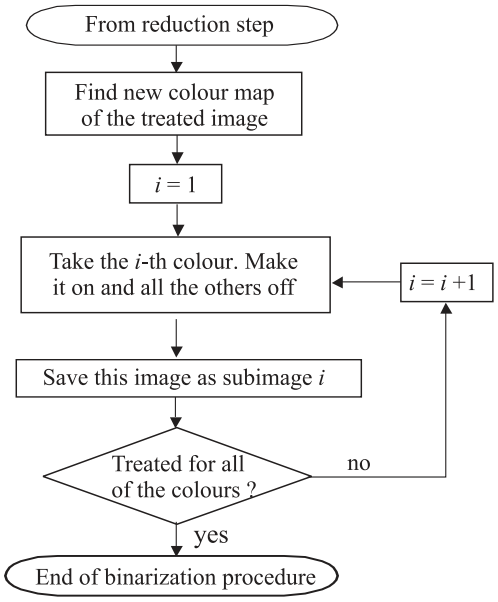

Fig. 2. Binarization of sub–images

separate and distinctive characters for further OCR processing. Christian Wolf, Jean-Michel Jolion and Francoise Chassaing [9] presented an algorithm to localize artificial text in images and videos using a measure of accumulated gradients and morphological post processing to detect the text. Oleg G. Okun [10] proposed a new method for text extraction from binary images with a textured background. He basically used a morphological filter using the top-hat transform. P. Clark and M. Mirmehdi [11] presented a method based on statistical properties of local image neighborhoods for the location of text in real-scene images. C. Datong, B. Hervé and Jean-Philippe T. [12] presented a fast and robust algorithm to identify text in image or video frames with complex backgrounds and compression effects. Qixiang Ye, Wen Gao, Weiqiang Wang and Wei Zeng [13] proposed an algorithm for detecting text in images and video frames. The algorithm contained two steps: initial detection and verification. Maija Koivusaari, Jaakko Sauvola and Matti Pietikäinen [14] proposed a new approach to automate document image layout extraction for an object-oriented database feature population using rapid low level feature analysis, pre-classification and predictive coding. P. Clark and M. Mirmehdi [15] presented a method for the fronto-parallel recovery of paragraphs of text under full perspective transformation. Hanchuan Peng, Fuhui Long, Wan-Chi Siu, Zheru Chi, and David Dagan Feng [16] presented a new matching algorithm based on document component block list and component block tree.

2 THE PROPOSED APPROACH

The developed algorithm is divided into three groups:

1) Detection algorithm
2) Recognition algorithm
3) Fuzzy System (which is part of recognition group)

2.1 Detection algorithm

2.1.1 Reducing the image's colors

We are dealing with images which can be classified as 8-bit type, which means that 256 different colors. This makes dealing with each single color harder and takes some time. Also, a thresholding point can't be taken to convert the image to binary as all previous works did. Because if the color of the text is almost as that of the background it will be lost, and also it may cause a broken point in the single character or merging in a group of characters which consider them as noise. So the objective here is to reduce the number of colors used to a minimum. This will keep the basic features while removing the colors that do not appear much in the image, *ie* colors that are shown in less than 60 pixels. Now because of this method, if the text color almost matches the background color, the text will not be considered as part of background. This will not affect the text features. The reducing colors flowchart is shown in Fig. 1.

2.1.2 Converting the image into sub-images and binarize them

A number of images will be created which have the same height and width of the original image and in each one, we will consider one of the colors to be on and the others are off which will give us a binary image each time.

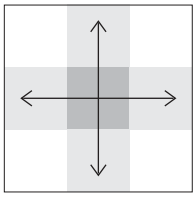

Fig. 3. 4-connected components

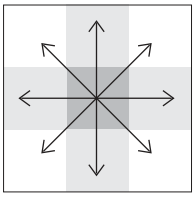

Fig. 4. 8-connected components

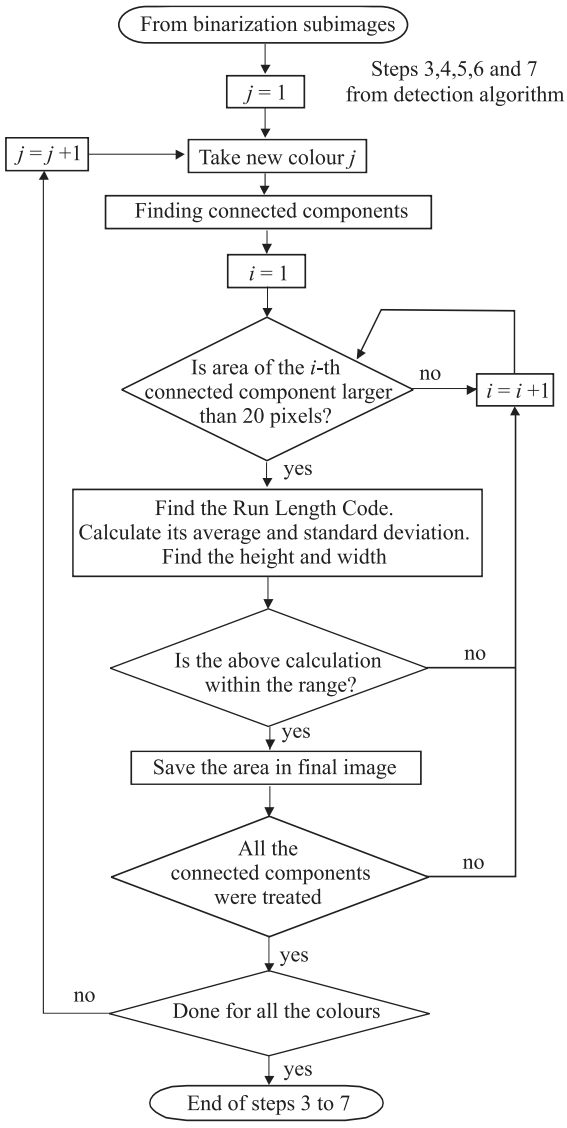

Fig. 5. Steps 3, 4, 5, 6 and 7 of the detection algorithm.

So, taking one color at a time will solve the problem of merging the text with the background. Figure 2 shows the flowchart.

2.1.3 Finding the connected components

There are two methods of connected components:
(A) 4-connected method (Fig. 3), where Pixels are connected if their edges touch. This means that pair of adjoining pixels is part of the same object only if they are both on and are connected along the horizontal or vertical direction.
(B) 8-connected component (Fig. 4), where Pixels are connected if their edges or corners touch. This means that if two adjoining pixels are on, they are part of the same object, regardless of whether they are connected along the horizontal, vertical, or diagonal direction.

2.1.4 Deleting the noise

Each component in the image which has area less than 20 pixels is considered as noise. This assumption is justified because this algorithm is designed to capture the text which has font size between 17–29 points.

2.1.5 Calculating the basic features of the connected components

For each connected component, Run Length Algorithm will be applied, which means that for every row or column we should count how many times it changes from “on” to “off” or “off” to “on” . As an example for 0011101010101 it gets 9 and for 0000001111 it gets 1. Then calculate the average and standard deviation of them for horizontal projection and again for the vertical projection. Finding the height, the width and the ratio length of connected component are also counted.

2.1.6 Classifying the connected component

Based on the calculations which have been done before we are able to decide if the connected component is a character or not. The thresholds values are based on the same calculations done for three different styles of fonts which are: Lucida Console, Verdana and Arial and sizes of 17–29 points. Take the letters from (a–z) and from (A–Z) for each font type and size and calculate RLC for each one of them, this will give the pass range for each letter. The maximum and minimum of this range are considered as threshold points.

2.1.7 Leaving the image with the text area only

If the features of the connected component agree with the thresholds data it will stay in the image as text, otherwise it will be deleted. The above presented approach is summarized in Fig. 5.

2.2 Recognition algorithm

In this part, the characters will be identified as letters and the image will be converted to text. After the image is cleaned up and becomes a binary image which contains only the text, the binary image is then saved and the memory is cleaned up. This step is very important to increase the speed of the system. After that the following steps should be done.

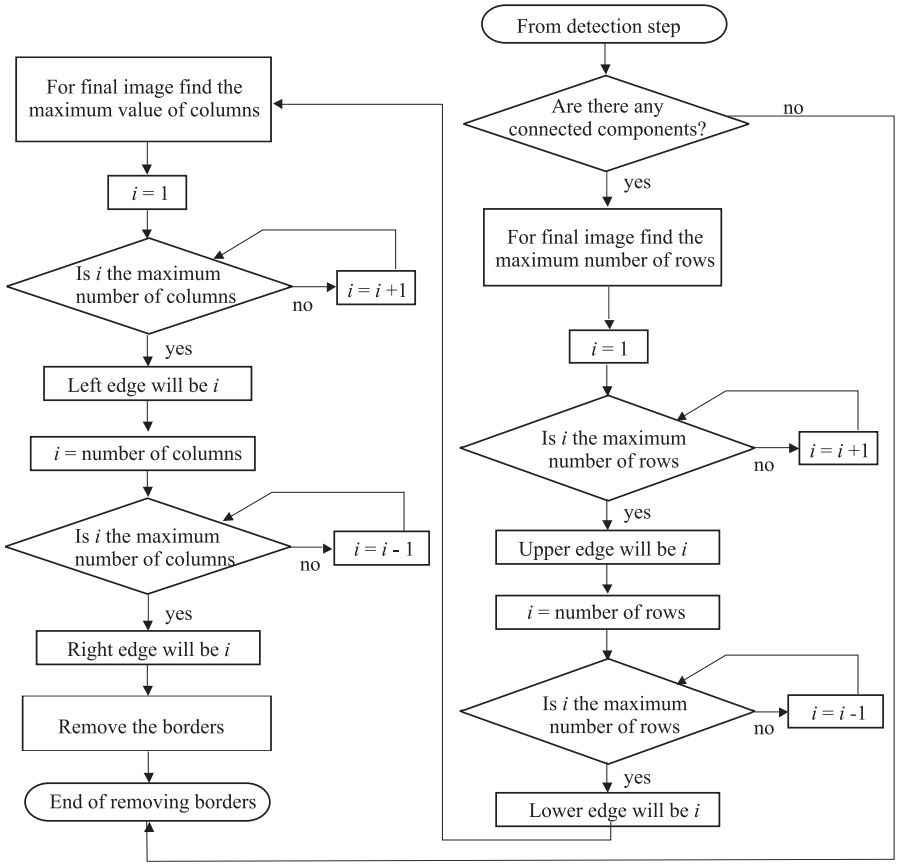

Fig. 6. Remove the borders flowchart.

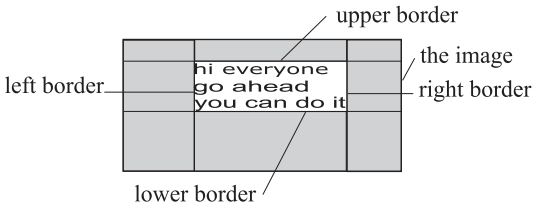

Fig. 7. Removing borders example.

2.2.1 Remove the borders

The borders should be removed; this will reduce the image size. Only the rectangular part of the image which contains the text will remain. Other parts will be removed; this step is presented in Figs. 6 and 7.

As shown in Fig. 7 the gray region will be removed, this will make the image size smaller and consequently make the program faster.

2.2.2 Divide the text into rows

After moving the borders, the area will now be divided into rows. Again the main rule is: there are at least three characters with the same color. Each row is saved in an array to be used in the next step (Figs. 8 and 9).

2.2.3 Divide the rows (lines) into words

The single line is then divided into words. Before that, the empty area before and after the text are removed. There are no restrictions imposed on the word as many other algorithms did. The word may be a single character or more than that, the size may be different in the same word and it is not necessary that the word has a meaning. Figures 10 and 11 present how rows can be divided into words.

2.2.4 Divide the word into letters

Each word is then divided into characters and saved in an array. Again the main assumption is that no merge between characters and no break points in the single character, this step is shown in Figs. 12 and 13.

Then for each character we calculate the Euler number which is equal to the number of the connected component in the image minus the number of holes. This will divide the characters into 3 groups:

1. Euler number equal one and this contains: s, S, f, F, G, h, H, j, J, k, K, l, L, z, Z, x, X, c, C, v, V, n, N, m, M, w, W, E, r, t, T, y, Y, u, U, i, I.
2. Euler number equal zero and this contains: q, Q, R, o, O, p, P, a, A, d, D, g, b.
3. Euler number equal minus one and this contains: B.

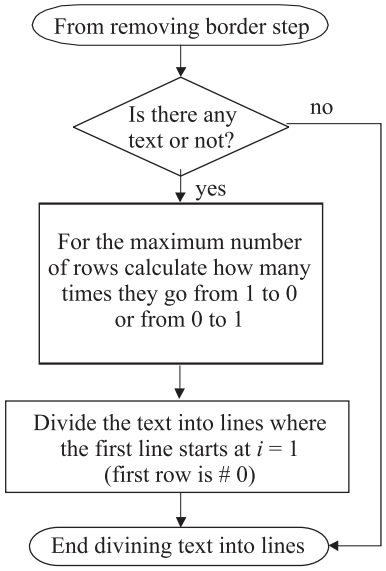

Fig. 8. Divided text into rows flowchart.

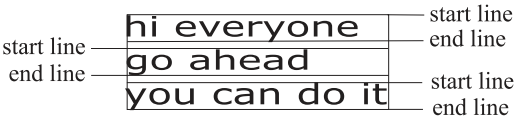

Fig. 9. Example of dividing text into rows.

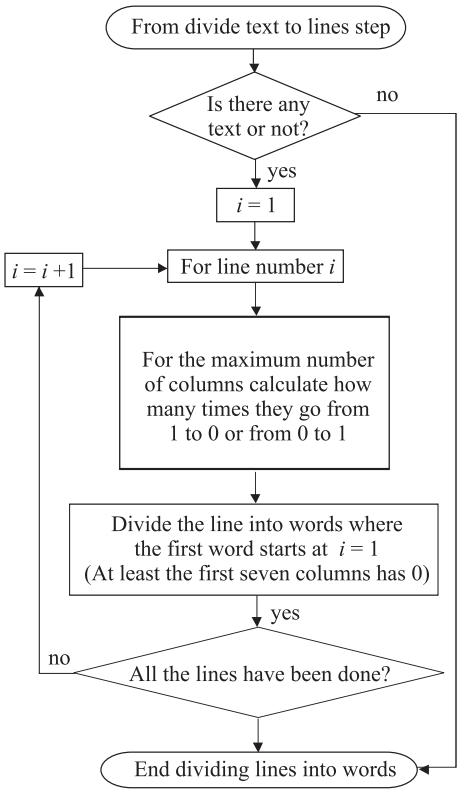

Fig. 10. Divided rows into words.

As we can see the third group contains only one letter B and this makes the identification accuracy very high, because there is no other letter can look like it. The recognition system here is to match the character behavior in the image, and answer the following questions: where is the location of the centroid? Which corner in the image is on? *etc.* but due to complexity of group one and two, they are sent to a fuzzy system to identify the character.

2.2.5 Send the information of the character image to a fuzzy system to identify the character

For the first group as we can see there are a lot of characters which may have the same shape, so in this part we increase the number of inputs to increase the accuracy. The inputs (inputs to the developed fuzzy system) are:

3. The four corners' pixels of the image (find if they are on or off).
2. The center of the image (find if it's on or off).
3. The center of each corner (find if it's on or off).

For inputs 1, 2 & 3, take the sequence: upper-left, upper-middle, upper-right, middle-right, lower-right, lower-middle, lower-left, middle-left then middle-middle. These inputs are crisp inputs not fuzzy inputs.

4. Ratio of its length (width/height).
5. The number of lines that will be cut by a line that goes through the center vertically
6. The number of lines that will be cut by a line that goes through 25 % horizontally.
7. The number of lines that will be cut by a line that goes through 58 % horizontally.

Figure 15 shows the inputs for the first group.

For the second group the inputs are:

1. The corners pixels of the image (find if they are on or off), the sequence is: upper-left, upper-right, lower-right then lower left.
2. The position of the hole in x-direction.
3. The position of the hole in y-direction.
4. The number of lines that will be cut by a line that goes through the center vertically.

Figure 16 shows the inputs for the second group.

The inputs are processed in the fuzzy system; therefore, the character will be identified. For example, (Fig. 16) the inputs to the fuzzy system are [0 0 0 1 35 30 6], where the first four numbers are the corner pixel, the x-component of the center of the hole, the y-component of the center of the hole and the number of lines that will be cut by a vertical line goes through the center, respectively.

For the first fuzzy system we can see that the first ten inputs are crisp inputs, also for the second one the first four inputs are crisp. The classical system is a special case of a fuzzy system and the crisp input is a special case from fuzzy input, so that we can treat the crisp inputs like fuzzy inputs. Results show that the fuzzy system is able to handle the whole system perfectly. Fuzzy system for the first group is shown as an example in Appendix A.

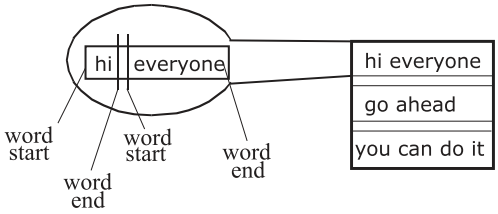

Fig. 11. Example of dividing a row into words.

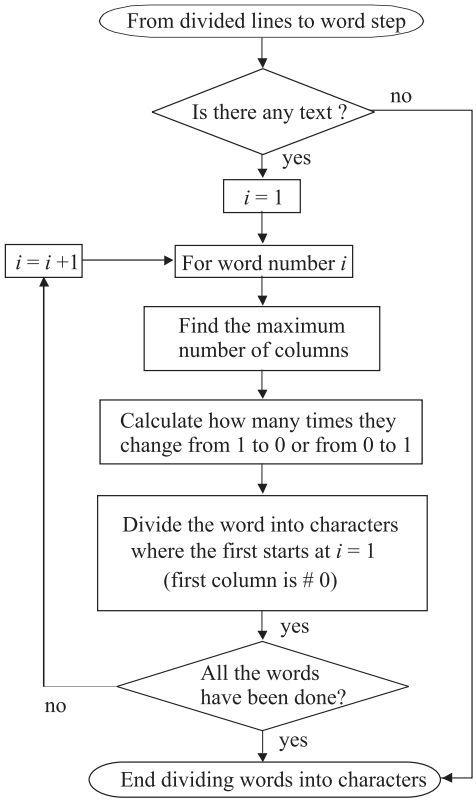

Fig. 12. Divide a word into characters.

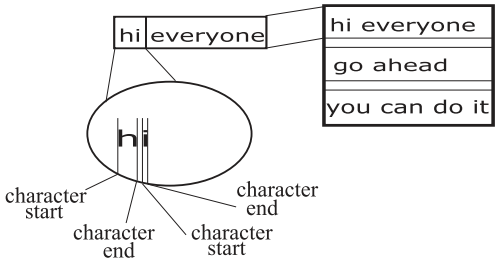

Fig. 13. Example of dividing a word into characters.

2.2.6 Print the character out

After the characters have been identified, each character will be printed in a sheet which is the final step in our approach. After that, the letters can be saved as a word document or notepad.

3 DISCUSSION OF RESULTS

3.1 Applying the proposed approach

Comparing this method with other methods, we note that the main difference is in the binarizing technique. Previous work depends mainly on histogram shape. They calculate the histogram, and then smooth it to find the threshold points. These methods are not working well when the text and background colors are very similar, and also if it may create an area of negative text. The negative text may be treated as a graphic region which may affect the system efficiency.

The presented approach will take each color alone and treat it as ON and the others as OFF. This will make the merge between the text and the background unlikely to happen. Also there will be no negative text in the whole image because the negative text to a color will be normal text for another. Also we can see that most of the previous work goes under Page Segmentation methodology which means a huge image with white background and known layout form. These works are commonly used for only 1-3 layout forms. If more layers are needed, Neural Network will be used. But these methods failed when the image is unpredictable. The proposed approach in this paper is efficient with high accuracy as shown in the following examples. The results are shown for font size between 17–29 points for the three types of font: Verdana, Arial and Lucida Console. These types were chosen because the characters have low variance and there is less redundancy in the single character. Also, they have no breakpoints in the single character or merge in group of characters as noticed in Times New Roman type.

The result of the proposed algorithm will be printed in a separated file so anyone can edit it later. The recognition part as we can see depends on a very basic fuzzy system which makes the system fast and more flexible.

In Fig. 17 we can see that a picture of a woman has been taken then text has been added to it. The text contain two words (meja) with green color and (meja) with Ceylon color. The space between letter e and letter j has been taken to insure that there is no merge between them. As you can see in Fig. 17(a) that image colors are within the blue region. In all previous work this image couldn't be treated because of similarity in colors between the text and the background. Also you can read the word meja with Ceylon color hardly, and all previous work imposed restrictions on this situation.

The first step in our approach is to decrease the number of colors in the image by ignoring colors which have number of pixel below 60 pixels, and by decreasing the level of colors from 256 to 150; the result is shown in (b). Each color will be treated as a binary image by considering the color ON and the others to be OFF. The results are shown in (c), (e), (f), (g), (h), (i) and (j). In each binary image the connected components will be detected and segmented to classify it as a character or noise. The restrictions are:

Table 1. Accuracy comparison

image	# of characters	The proposed approach	ABBYY Finereader 7	soft writing	Matti algorithm	Kwang algorithm
1	70	100	47.14285714	18.57142857	37.71428571	61.42857
2	21	100	9.523809524	0	100	23.80952
3	38	100	100	0	100	28.94737
4	17	82.35294118	29.41176471	29.41176471	0	88.23529
5	5	100	100	100	0	100
6	59	100	57.62711864	97.61016949	0	100
7	9	100	0	0	10	100
8	11	100	100	81.81818182	9.090909091	100
9	12	50	50	0	0	50
10	8	100	87.5	50	0	50

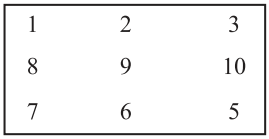

Fig. 14. The sequence of inputs 1, 2 & 3.

- There are at least three objects in a single line having the same color.
- The object properties match the threshold points.
- The object area is not below 20 pixels.
- The height and width are not above 45, 40 pixel, respectively.

You can see that by using this algorithm the negative text problem has been solved. In (i) you can see the negative text which has been treated as a normal text in (e) and (h). The final result is shown in (k) where all the background has been removed leaving the text only. Then the final image will be divided into lines (rows). For each line the text in it will be divided into characters which are then classified into three groups as discussed in the previous discussion. Each character will then be processed to find some values which are the inputs to the fuzzy system. The character is then identified and the result will be printed out.

3.2 Comparison between different approaches

Here we are going to make comparisons between the proposed algorithm and some other popular algorithms; Finereader 7.0 professional and softwriting, Matti's algorithm and Kwang's algorithm [17]. Matti's algorithm [2] depends on the number of edges in single block and also it deletes the weak edges from the image. Kwang's algorithm depends on SVM and Neural Networks. The programs were named ABBYY Finereader 7 and soft writing.

The proposed approach is compared with other approaches. The accuracy comparison is shown in Tab. 1 and the false position comparison is shown in Tab. 2. The accuracy is calculated as follows:

$$\text{Accuracy} = \frac{\#\text{ of letters detected}}{\text{total }\#\text{ of letters}} \times 100\,\%$$

The images from B.1–B.10 are shown in appendix B. The second column in Tab. 1 shows the number of characters in each image. As we can see, the accuracy of the proposed approach is better than that in other approaches. In image 9 we can see that the accuracy is low and this is because there are letters in this image that do not belong to our font styles which make the error goes high.

The second comparison is how many false positions have been detected. This comparison is shown in Tab. 2.

Table 2. False position comparison.

Image	The proposed approach	ABBYY Fine-reader 7	soft writing	Matti algo-rithm	Kwang algo-rithm
1	0	0	0	10	0
2	0	3	0	0	9
3	0	0	0	0	1
4	0	0	0	> 50	2
5	0	2	3	> 50	0
6	0	3	2	> 50	3
7	0	0	0	> 50	2
8	0	0	0	> 50	0
9	0	0	0	> 50	1
10	0	1	0	> 50	1

As we can see, there is no false position in our algorithm compared to others. Because of these results we can see that all algorithms have high accuracy if the images have a single color background with a single color text, but when this restriction is eliminated the accuracy goes down. Also, if the image contains a lot of colors, over 10 colors, and the text color almost matches the background color, the results other algorithms become so poor, while the proposed approach maintains its accuracy.

4 CONCLUSIONS

In this work, a connected component method was used to establish the detection algorithm and the recognition

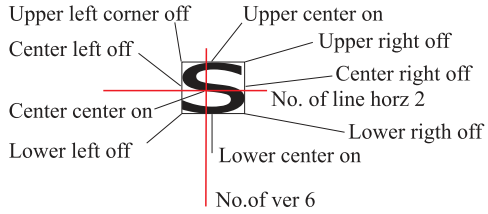

Fig. 15. Inputs for the first group.

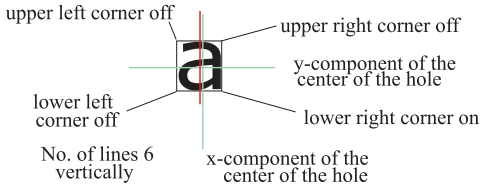

Fig. 16. Inputs of the second group.

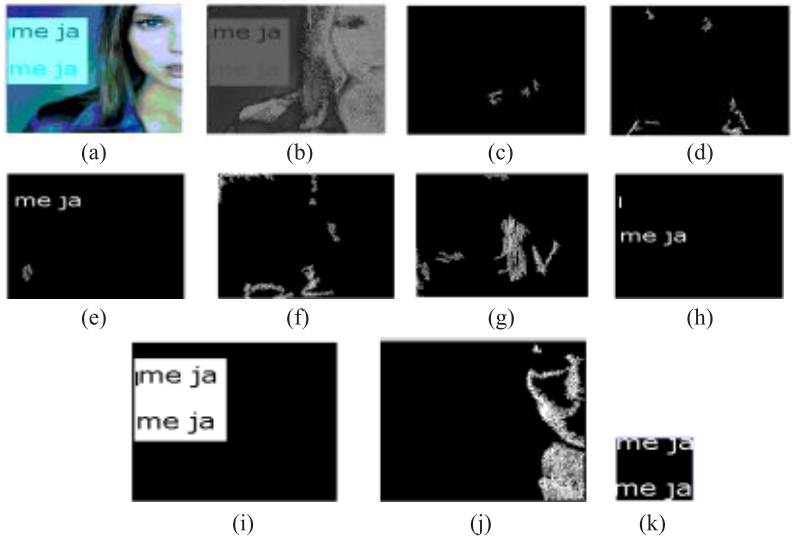

Fig. 17. Example (a) the original image (b) result after reducing the order of colors (c), (d), (e), (f), (g), (h), (i) and (j) extraction of connected component (k) the final result.

algorithm. A computer program was developed to implement the algorithm for three types of fonts and sizes within 17–29 points. The proposed algorithm assumed that there are at least three characters of same color in a single line, the character is on its best view which means no broken point in a single character and no merge between group of characters and at last, a single character has only one color. The basic algorithm used the 8-connected component to binirize the image, then to find the characters in the image and recognize them. The shown and discussed results showed the efficiency of the algorithm and how it is different compared with other algorithms.

APPENDIX A

Fuzzy system

Both fuzzy systems have been developed using Matlab Fuzzy toolbox. The systems are based on Mamdani fuzzy approach. The first task is to define the inputs and the outputs of the fuzzy system. This stage depends on expert decision. For the first fuzzy system the inputs are in Fig. A.1.

The output memberships of the first system are in Fig. A.2

In Fuzzy system design the expert will identify the commands (rules) that will model the relationship between the inputs and the output. Some rules are listed here for the first system.

First lets use the following summarized notation

Upper-left → UL	Lower-left → LL
Lower-middle → LM	Lower-right → LR
Upper-middle → UM	Upper-right → UR
Middle-left → ML	middle-middle → MM
Middle-right → MR	Ratio length → R
position hole x → X	position hole y → Y
No. of line intersection vertically → V	
No. of line intersection horizontally 25 % → H25	
No. of line intersection horizontally 75 % → H75	

The following are some of the rules that are identified by the expert.

— If (UL is on) and (UM is on) and (LR is on) and (LM is on) and (LL is on) and (ML is on) and (MM is on) and (V is high) and (H75 is low) then (output is e).

— If (UM is on) and (UR is on) and (ML is off) and (LR is on) and (LM is on) and (LL is on) and (ML is off) and (MM is off) and (V is high) and (H75 is low) then (output is z).

— If (UL is off) and (UM is on) and (LR is off) and (LM is on) and (ML is off) and (MM is off) and (V is high) and (H75 is low) then (output is s).

— If (UL is on) and (UM is off) and (UR is on) and (ML is off) and (LR is off) and (ML is off) and (MM is off) and (V is low) and (H75 is low) then (output is y).

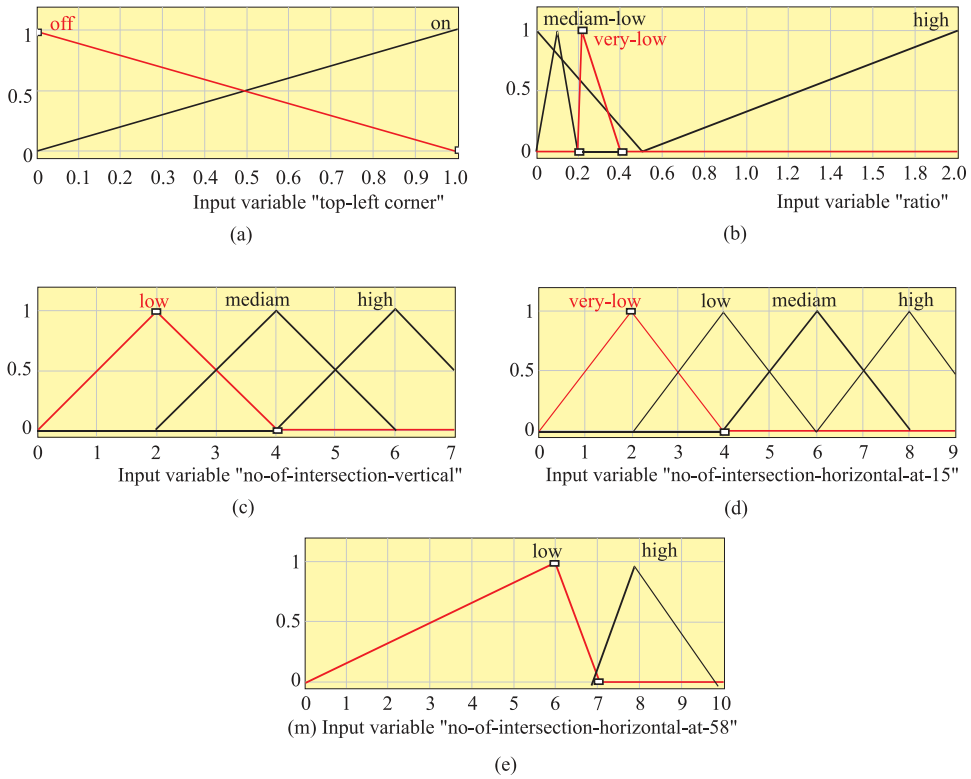

Fig. A.1. The inputs of the second system, (a) input1(upper-left corner), (b) input2(upper-middle corner), (c) input3(upper-right corner), (d) input4(middle-right corner), (e) input5(lower-right corner), (f) input6(lower-middle corner), (g) input7(lower-left corner), (h) input8(middle-left corner), (i) input9(middle-middle corner), (j) input10(ratio width/height), (k) input11(no. of line intersection vertically), (L) input12(no. of line intersection horizontally 25 %) and (m) input13(no. of line intersection horizontally 75 %).

— If (UL is on) and (UM is off) and (UR is on) and (ML is off) and (LR is off) and (ML is off) and (MM is off) and (V is mediam) and (H75 is low) then (output is y).

— If (UL is off) and (UM is on) and (UR is on) and (ML is off) and (LR is off) and (LL is off) and (ML is off) and (MM is off) and (V is low) and (H75 is low) then (output is f), *etc.*

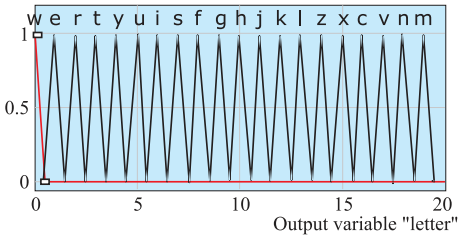

Fig. A.2. The output of the first system.

APPENDIX B

The pictures used in the comparison test

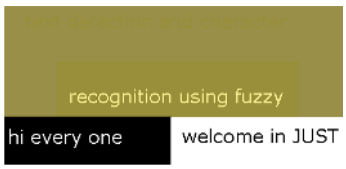

Fig. B.1. image 1

Fig. B.2. image 2

Fig. B.3. image 3

Fig. B.4. image 4

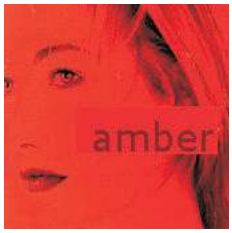

Fig. B.5. image 5

Fig. B.6. image 6

Fig. B.7. image 7

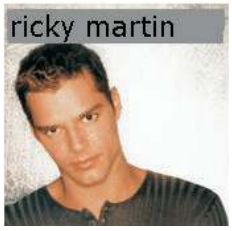

Fig. B.8. image 8

Fig. B.9. image 9

Fig. B.10. image 10

References

[1] YUAN, Q.—TAN, C.: Text Extraction from Gray Scale Document Images Using Edge Information, Proc. Sixth Int. Conf. on Document Analysis and Recognition, 302–306.

[2] PIETIKÄINEN, M.—OKUN, O.: Text Extraction from Grey Scale Page Images by Simple Edge Detectors, Proc. of the 12th Scandinavian Conference on Image Analysis (SCIA'2001), Bergen: Norway, 2001, 628–635.

[3] PERANTONIS, S.—GATOS, B.—MARAGOS, V. A.: Novel Web Image Processing Algorithm for Text Area Identification That Helps Commercial OCR Engines to Improve Their Web Image Recognition Efficiency, Proceedings of the Second International Workshop on Web Document Analysis (WDA2003), UK: Edinburgh, 2003.

[4] CHEN, W.—CHEN, S.: Adaptive Page Segmentation for Color Technical Journals' Cover Images, Image and Vision Computing **16** (1998), 855–877.

[5] STROUTHOPOULOS, C.—PAPAMARKOS, N.—ATSALAKIS, A.—CHAMZAS, C.: Locating Text in Color Documents, ICIP, 2001.

[6] NEGI, A.—SHANKER, K. N.—CHEREDDI, C. K.: Localization, Extraction and Recognition of Text in Telugu Document Image, ICDAR03.2003, 1193–1197.

[7] KARATZAS, D.—ANTONACOPOULOS, A.: Two Approaches for Text Segmentation in Web Images, Seventh International Conference on Document Analysis and Recognition, IEEE, Scotland, Edinburgh, 2003, 1–131.

[8] HARA, S. OCR for Japanese Classical Documents Segmentation of Cursive Characters: National Institute of Japanese Literature, PNC2004 Annual Conference on Conjunction with PRDLA, 2004.

[9] WOLF, C.—JOLION, J.—CHASSAING, F.: Text Localization, Enhancement and Binarization in Multimedia Documents, In Proceedings of the International Conference on Pattern Recognition (ICPR). IEEE Computer Society, Canada: Quebec City. 2002, 4, 1037–1040.

[10] OKUN, O. Morphological filter for text extraction from textured background. In Vision Geometry X, Longin J. Latecki, David M. Mount, Angela Y. Wu, Robert A. Melter, Editors, Proc. of SPIE California: San Diego. 2001, 4476, 86–96 .

[11] CLARK, P.—MIRMEHDI, M.: Finding Text Regions Using Localized Measures, Proceedings of the 11th British Machine Vision Conference, 2000,.

[12] CHEN, D.—BOURLARD, H.—THIRAN, J. P.: Text Identification in Complex Background Using SVM, CVPR01, 2001, 2, 621–626.

[13] YE, Q.—GAO, W.—WANG, W.—ZENG, W.: A Robust Text Detection Algorithm in Images and Video Frames, the 4th International Conference on Information, Communications & Signal Processing — 4th IEEE Pacific-Rim Conference On Multimedia (ICICS-PCM2003), Singapore, 2003.

[14] KOIVUSAARI, M.—SAUVOLA, J.—PIETIKÄINEN, M.: Automated Document Content Characterization for a Multimedia Document Retrieval System, Proc. SPIE Multimedia Storage and Archiving Systems II. TX, Dallas, 1997, 3229, 148–159.

[15] CLARK, P.—MIRMEHDI, M.: Estimating the Orientation and Recovery of Text Planes in a Single Image, Proceedings of the 12th British Machine Vision Conference (BMVC2001), Tim Cootes and Chris Taylor, editors. 2001, 421–430.

[16] PENG, H.—LONG, F.—SIU, W.—CHI, Z.—FENG, D.: Document Image Matching Based on Component Blocks, Proceedings of 2000 International Conference, Canada, 2000; 2, 601–604.

[17] KANG, I.—KIM, K.—JUNG,—KIM, J.: Texture-Based Approach for Text Detection in Images using Support Vector Machines and Continuously Adaptive Mean Shift Algorithm, IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI) **. 25** No. 12 (2003), 1631–1639.

Received 11 August 2005

Mohanad Alata, (PhD), born in 1969, in Amman, Jordan, completed Bachelor at Jordan University of Science and Technology 1992, obtained a Master degree from Concordia University in Montreal, Canada 1996 and a PhD degree in Intelligent Control from Concordia University in 2001. His research interests are Fuzzy sets and Fuzzy systems, intelligent control, Automation and Robotics. Since 2001 he is an assistant professor at Jordan University of Science And Technology in the Mechatronics division at the Mechanical Engineering Department.

Mohammad Al-Shabi, born in 1980, in Kuwait, completed Bachelor of Mechatronics at Jordan University of Science and Technology 2002, obtained a Master degree in Mechanical Engineering from Jordan University of Science and Technology 2005. His research interests are Fuzzy data analysis, Image processing and OCR. Since 2005 he is a Lab. Supervisor at the Mechanical Engineering Department, Philadelphia University.