Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 11 (2013) 334 – 341

Procedia Technology

The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)

Arabic Character Recognition System Development

Iping Supriana*, Albadr Nasution

Informatics Engineering, Institut Teknologi Bandung, Jalan Ganeca 10 Bandung 40132, Indonesia

Abstract

We develop Arabic Optical Character Recognition (AOCR) system that has five stages: preprocessing, segmentation, thinning, feature extraction, and classification. In preprocessing stage, we compare two skew estimation algorithms i.e. skew estimation by image moment and by skew triangle. We also implemented binarization and median filter. In thinning stage, we use Hilditch thinning algorithm incorporated by two templates, one to prevent superfluous tail and the other one to remove unnecessary interest point. In segmentation stage, line segmentation is done by horizontal projection cross verification by standard deviation, sub-word segmentation is done by connected pixel components, and letter segmentation is done by Zidouri algorithm. In the feature extraction stage, 24 features are extracted. The features can be grouped into three groups: main body features, perimeter-skeleton features, and secondary object features. In the classification stage, we use decision tree that generated by C4.5 algorithm. Functionality test showed that skew estimation using moment is more accurate than using skew triangle, median filter tends to erode the letter shape, and template addition into Hilditch algorithm gives a good result. Performance test yield these result. Line segmentation had 99.9% accuracy. Standard deviation is shown can reduce over-segmentation and quasi-line. Letter segmentation had 74% accuracy, tested on six different fonts. Classification components had 82% accuracy, tested by cross validation. Unfortunately, overall performance of the system only reached 48.3%.

Keywords: Arabic optical character recognition; AOCR

* Corresponding author.
E-mail address: iping@informatika.org

1. Introduction

Optical character recognition (OCR) is very useful for human work, for example automatic license plate reading, mail sorting, or document editing. Research in this field has produced practical applications especially for Latin and East Asia characters. Arabic unfortunately still not much developed.

Arabic writing system has major differences compared to Latin and East Asia. Arabic has 28 letters. Some letters is very similar in form and have secondary object to differentiate it. Arabic is written right to left (RTL) cursively. Each letter changes shape depending on its position in a word. Some letters can only be connected from right. Therefore, unlike English, language component that separated by space is not a word but sub-word.

Arabic always written cursively both typed and printed. This means, intuitively, Arabic recognition system needs letter segmentation. This is one of the reasons Arabic OCR is underdeveloped compared to Latin and East Asia's.

Research in Arabic OCR (AOCR) is still wide-open and AOCR Application is not many. These motivate us to try to develop a prototype of AOCR system in this study.

2. Related Research

Sarfraz et al. [1] developed a license plate reader with 95% accuracy using template matching. But, template matching assumes there is only one typeface used. Therefore, this method cannot be used in wider context.

Izakian et al. [2] developed a multi-font AOCR application using chain code feature. They achieved very high accuracy for 3 fonts: 97.4%. Unfortunately, they did not involve segmentation, only a manually segmented letter.

Comprehensive research about Arabic letter feature is done by Abandah and Kheder [3]. There is four group of letter feature. First, main body feature e.g. aspect ratio, pixel distribution, and loop. Second, boundary feature e.g. chain code and perimeter length. Third, skeleton feature e.g. interest point. And the last, secondary object feature. Letter is grouped by similarity of its main body. After main body is determined, secondary object feature is used to classify the letter. Unfortunately, this research also did not involve segmentation.

Al-Taani and Al-Haj [4] develop online AOCR application using decision tree. They use segment number (letter stroke that written without lifting the pen) and density ratio of the letter. Density ratio is computed by dividing the letter image by quadrants. The feature is number of pixel of each quadrant per total area of letter.

Zidouri [5] proposed a general method for Arabic letter segmentation. This method uses thinned image of sub-word to find band of horizontal pixel on the baseline. If this band reaches a threshold, it is considered a candidate to segment the sub-word. Actual separation line is chosen from candidates that fulfill one of the four rules proposed based on local features of each band candidates.

3. System Architecture

We developed a modular AOCR system. The system developed has five stages. The stages, process implemented in each stage, and the data flow can be seen in system architecture in Fig.1.

3.1. Preprocessing Stage

Processing stage input is document image. This stage implements several processes such as binarization, skew detection, and median filter. Binarization converts input image into black and white image. Three RGB channel is checked. If two of them have value less than 127, pixel is converted to black. The purpose of this two value scheme is to reduce noise. Skew detection corrects skewed image. In this study, we implement two methods of skew detection. First, finding skew angle of document by image moment as proposed by Kapogiannopoulos and Kalouptsidis [6] and second, by skew triangle.

Skew triangle is a white triangle formed by skewed document. Arabic is written RTL. Because of that, we can assume that skew triangle can be found on the right side of the image, see Fig. 2. The triangle can be estimated by its two tip points i.e. the most far point in the top and right area. The problem is when the document image is not skewed at all there is no triangle. To find out whether the document is really skewed or not, number of pixel inside the triangle is counted. If there are many, we can be determined that the skew triangle found is false and the document is not skewed. The reason is the triangle formed by two tip points found will be crossing some writing of document images.

Median filter aims to reduce noise by replacing the value of each pixel to the median of its and its neighbor's value.

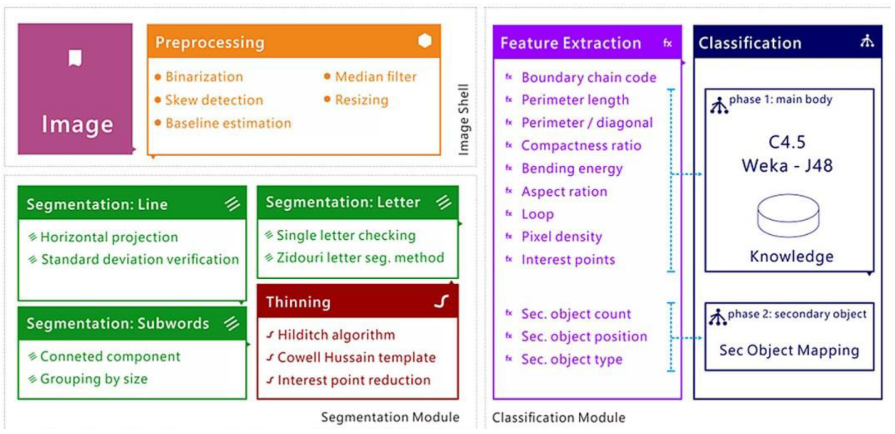


Fig.1. System Architecture

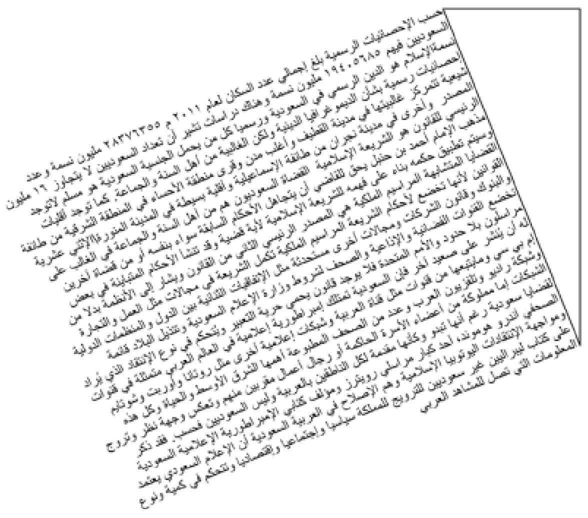


Fig. 2. A skewed document and its skew triangle

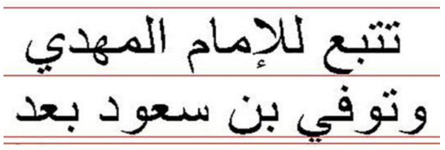


Fig. 3. Line segmentation and quasi-line

3.2. Segmentation Stage

There is three kind of segmentation i.e. line segmentation, sub-word segmentation and letter segmentation.

Line segmentation is done using horizontal projection. We search array of pixel rows that have projection value less than a threshold. Twenty pixels are used as threshold. We did not use zero for the purpose to separate the lines even when there is very small space between them. Zero space between writing lines is frequent in small document image.

After obtaining the separation line between lines, the height population of line segments produced is verified by standard deviation. The purpose is to eliminate line segment that height is very small compared to others. This kind of segments usually produced by noise or space between writing line and secondary object is too far (we called it quasi-line, see Fig. 3) or even over-segmentation. Quasi-line contains only noise or secondary object. Quasi-line is not an over-segmentation because the main body is still intact.Verification is done by checking the mean and standard deviation of height population. If the mean per standard deviation is less than 10%, the segment population is considered normal. If not, some segments that have height less than mean minus standard deviation is considered abnormal. These segments will be joined by its predecessor or successor depending on horizontal projection in its two separation line. Separation line with highest horizontal projection will be deleted and the line segment will be joined.

Sub-word segmentation is done using connected pixel component, using Amin implementation [7]. Arabic sub-words' pixel always connected thus it can be segmented by finding the connected pixel component. After obtaining the connected pixel components, each component is grouped into one of three groups: main body, secondary object, and noise. Grouping is done by size. Main body, the biggest one, is considered has more than 400 pixels or crossed by baseline. Secondary object has size between 40 and 400 pixels. The rest is considered as noise. This absolute threshold can be done after resizing the line segments by 64 pixel height before sub-word segmentation is performed.

Letter segmentation is done using Zidouri algorithm [5]. There is no major modification in the implementation.

3.3. Thinning Stage

This stage is performed before letter segmentation. The thinned image resulted from this stage is used in Zidouri algorithm. Thinning stage input is only main body image of a sub-word. The secondary object of the sub-words is deleted before thinning stage is performed.Thinning is performed by Hilditch thinning algorithm [8] by incorporating some templates. This addition is done because Hilditch is not fine-tuned for Arabic writing.

There are two additional templates. First one is template to prevent superfluous tail by Cowell and Hussain [9]. Arabic writing tends to have thickness variation. In the thick stroke, thinning algorithm will produce unnecessary tail. This template is presented in Fig.4 (a). Second one is template to remove unnecessary interest point. Hilditch algorithm produces thinned image that has more than one pixel thickness. Consequently, interest point produced will be abundantly unnecessary. This template has purpose to remove it. This template is presented in Fig.4(b).

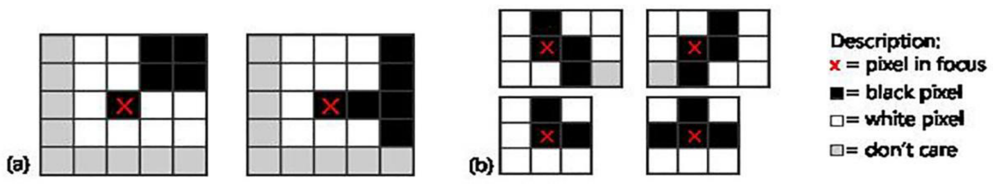


Fig.4. Thinning template: (a) Superfluous tail prevention [9] (b) Unnecessary interest point removal

3.4. FeatureExtraction Stage

This stage input is letter image both normal one and thinned one. There are three groups of feature:

1. Main body feature.
   This group is related to the main body of a letter. This group includes aspect ratio, distribution of pixel, and loop feature. Pixel distribution is computed by dividing the main body image by four quadrants [2][3]. This pixel distribution yields six values, four values from each quadrant, top bottom ratio, and left right ratio.
2. Boundary and skeleton feature.

This group is related to the boundary and the skeleton of the main body of a letter. This group includes interest points, perimeter length, perimeter-to-diagonal ratio, compactness ratio, and bending energy. The last four is boundary features implemented by Abandah and Khedher [3]. Interest point is a pixel that have one neighbor (end point), three neighbor (branch point), and four neighbor (cross point). Each is counted in each quadrant.

3. Secondary object feature.
   This feature is related to the secondary objects of a letter. This group includes position, number, and type of each secondary object. There is two type of secondary object, dot or hamza. Type is estimated from thinning result of the secondary object and compactness ratio (to see roundness) of the secondary object.

3.5. Classification Stage

This stage input is features that obtained from feature extraction stage. We use decision tree in the classification stage. This tree is generated by C4.5 algorithm. We use main body image of each letter from each font as training set.

We use 24 features used to classify main body on this classification stage: six values of pixel distribution, twelve values of interest points, four values of boundary features, one value of aspect ratio, and one of loop number.

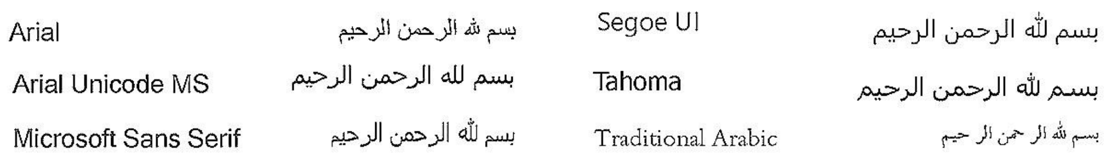
After main body is classified, second part of classification classify letter by checking secondary object features. For each main body, there is a mapping to one letter depends on its secondary object features. In this second part of classification, some known classification error from first part is corrected on the mapping. For example, if the main body is classified as nine (٩) but has two dots, it will be reclassified by qāf (ق), not the number nine (٩).

4. Evaluation and Analysis

4.1. Test Data And Procedure

We use some documents image as test data. The writing is taken from article titled “Kingdom of Saudi Arabia” from Wikipedia Arabic. This article is printed using multiple typefaces as seen in Fig.5. Total test data is 37 pages.

For training the classification stage part one (to classify main body of letter), we use main body image of single-letter from each typeface used. These images is manually segmented and grouped by the similarity of its shape.



Arial	بسم الله الرحمن الرحيم	Segoe UI	بسم الله الرحمن الرحيم
Arial Unicode MS	بسم الله الرحمن الرحيم	Tahoma	بسم الله الرحمن الرحيم
Microsoft Sans Serif	بسم الله الرحمن الرحيم	Traditional Arabic	بسم الله الرحمن الرحيم

Fig.5.Basmallah on every typeface used in test data

4.2. Evaluation Result

Median filter functionality is correct. Noise is reduced. If performed repeatedly, noise can be minimized. But, it is shown that letter shape is eroded as a result. Some lost its continuity and broken. Secondary object is getting smaller and even diminished. Based on this result, we exclude median filter from overall process.

Skew estimation functionality is tested by randomly skewed by -30 to 30 degree of each 37 document images. Skew estimation by image moment can correctly estimate all of document images. Average error is 1.5 degrees. Average execution time is 10.3 seconds per image.Skew estimation by skew triangle can only correctly estimate 84% of document images (the rest 16% has skew angle more than five degrees after correction). Average execution time is 5.8 seconds per image. Skew correction error is due to noise in the area of the skew and some error is due to correcting already not-skewed image. Average error of this method is 2.9 degrees. This is surprisingly small and can compete to 1.5 degrees error of image moment method, considering one of the images is falsely "corrected" by 85 degrees error. By only counting error from the 84% correctly corrected image, this method only has 0.05 degrees error.

Thinning stage can yields decent one pixel thick skeleton. Template addition is shown a good thing. It can delete superfluous tail as shown in Fig.6 (a) and remove unnecessary thickness interest point as shown in Fig.6(b).

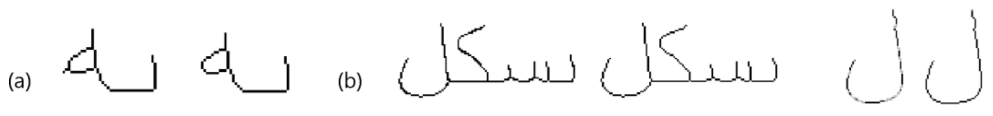


Fig.6. Skeleton:(a) without (left) and with (right) superfluous tail template, (b) without (left) and with (right) interest point template.

Line segmentation has accuracy 98% with 2% over-segmentation. Line segmentation performance for each typeface is presented in Table 1. Column labeled "Quasi" is quasi-line i.e. line that only contain secondary object or noise.

With standard deviation verification, the number of quasi-line and over-segmentation is greatly reduced.However, it should be noted that standard deviation verification is done by removing the separation line between writing-lines. Thus, this method is prone to under-segmentation. In both test, under-segmentation is not found but if we repeat the verification (in hope to eliminate quasi-lines and over-segmentation left), under-segmentation will increase. By repeating the verification process only twice, under-segmentation is increased by 2%.

Line segmentation performance with incorporating standard deviation verification is 99.9%.

Table 1.Performance of line segmentation

Typeface	Without Standard Deviation Verification			With Standard Deviation Verification		
	Correct	Over	Quasi	Correct	Over	Quasi
Arial	99%	1%	21%	100.0%	0.0%	0.0%
Arial Unicode MS	99%	1%	9%	100.0%	0.0%	0.8%
Microsoft Sans Serif	98%	2%	2%	100.0%	0.0%	0.0%
Segoe UI	98%	2%	27%	100.0%	0.0%	4.0%
Tahoma	98%	2%	13%	100.0%	0.0%	0.8%
Traditional Arabic	97%	3%	33%	99.4%	0.6%	0.0%

Table 2.Performance of letter segmentation

Typeface	Correct	Over	Quasi
Arial	84%	11%	5%
Arial Unicode MS	79%	17%	4%
Microsoft Sans Serif	54%	43%	3%
Segoe UI	91%	2%	7%
Tahoma	92%	2%	6%
Traditional Arabic	46%	50%	4%

Sub-word segmentation can generally extract the connected pixel components and group them into main body or secondary object. The only problem found is number zero (٠) and letter combination of *lam-alif* (لا). Because of its size, zero usually grouped into secondary object. On the other hand, *lam-alif* (لا) has main body part that detached from the sub-word. Both cases will be ignored for they are not considered as legitimate main body.

Letter segmentation has accuracy 74% with 21% under-segmentation and 5% over-segmentation. Letter segmentation performance for each typeface is presented in Table 2.

In Table 2, we can see that Microsoft Sans Serif and Traditional Arabic has very low accuracy. The cause is their letter spacing as seen in Fig.5. Both typefaces have very narrow space between letters. Thus, Zidouri algorithm cannot find appropriate band candidate more than the threshold. Furthermore, Traditional Arabic has many ligatures. This can greatly increase under-segmentation.We also can see the high accuracy of Segoe UI and Tahoma. This is due to wide space of its letter spacing. But, the over-segmentation in these typefaces also increased compared to other typefaces. The wide spaces also widen some letter such as sīn (س) and syīn(ش) that prone to over-segmentation.

Classification stage has accuracy 82% by cross validation. This accuracy is achieved using 24 features for generating the decision tree and using normal and bold image of letter as the training set. The main body classes are 34 letter shapes and 10 number shapes.

Before this 82% accuracy obtained, we perform some experiments. Using only 11 features (aspect ratio, six pixel distribution, and four boundary features) and 56 target main body letter classes (as recommended by Abandah and Kheder [3]), we get 46% accuracy. This increased to 64% after incorporating interest point features and number of loop (total 24 features are used). This increased even more to 82% after compressing the amount of target class into 34 letter shapes. This main body-classes compressing is done by joining similar connected letter form from a letter into one class. For example, ـبـ and ـتـ has two classes before compressing but joined into one class after compressing.

Overall performance of the system is quite disappointing, only 48.3 accuracy. This is less than expected accuracy based on performance of letter segmentation and classification, i.e. 74% × 82% = 61%. The cause is probably incompatibility between training set used to generate the decision tree and the real letter segments.

5. Conclusion

Skew estimation by image moment is more accurate but slower than by skew triangle. But, skew triangle method has somewhat greater accuracy on noise free skew image.Median filter is not compatible for Arabic recognition system. It erodes letter and minifies secondary object. It needs to be redesigned specifically for AOCR.On the other hand, incorporating superfluous tail prevention template of Cowell and Hussein and unnecessary interest point removal template into Hilditch thinning algorithm yields a good skeleton.

Line segmentation using horizontal projection gives a good result on a printed document. The accuracy is 99.9%. This relies on the skew estimation algorithm though. But, horizontal projection is sensitive to a short writing-line. Almost all of over-segmentation on the experiment comes from this kind of line.

Line segmentation using horizontal projection can produce a quasi-line i.e. line-segment that only contains noises or secondary object. Quasi-line and over-segmentation can be greatly prevented by using standard deviation verification, though this method is prone to under-segmentation if not used with care.

Zidouri algorithm for letter segmentation is quite good. The accuracy of this method is 74%, evaluated on six different typefaces. This method gives great accuracy if the typeface has a wide letter-spacing nature. If letter-spacing is narrow and the typeface has many ligatures, under-segmentation is very high. If letter-spacing is wide, the number of correctly segmented letter is high but over-segmentation is also increased relatively.Using Arial as parameter reference, Zidouri algorithm can gives a good result on typefaces Arial Unicode MS, Segoe UI, and Tahoma.

Zidouri algorithm is using skeleton image as input for searching the separation line. Therefore, it depends on thinning algorithm and can be affected by noise. For example, if the main body is jagged, the thinned skeleton will also be jagged. Consequently, the algorithm cannot find a band of horizontal pixel on the baseline as the thinned baseline is not straight. Finally, under-segmentation will be increased.

Classification using decision tree generated by C4.5 algorithm evaluated by 10-folds cross validation gives 82% accuracy. The accuracy increase with more features used to generate the decision tree, with more training data, and less class target. Two step classification i.e. classify main body and then classify letter is a good approach to handle the nature of Arabic similar main body and the existence of secondary object.

Overall performance is only 48.3%. The cause is probably incompatibility between training set used to generate the decision tree and the real letter segments. Further research is needed to enhance this performance.

References

[1] Sarfraz, Muhammad, Ahmed, Mohammed Jameel, dan Ghazi, Syed A. Saudi Arabian License Plate Recognition System. In Proceedings of the 2003 International Conference on Geometric Modeling and Graphics (GMAG'03) (London), IEEE Computer Society Press; 2003. pp. 36 - 41.

[2] Izakian, H., Monadjemi, S. A., Ladani, B. Tork, dan Zamanifar, KMulti-Font Farsi/Arabic Isolated Character Recognition Using Chain Codes. World Academy of Science, Engineering and Technology 43; 2008. pp. 67-70.

[3] Abandah, Gheith A. dan Khedher, Mohammed Z. Analysis of Handwritten Arabic Letters Using Selected Feature Extraction Techniques. International Journal of Computer Processing Of Languages, 22, 1; 2009. pp. 1-25.

[4] Al-Taani, Ahmad T. dan Al-Haj, Saeed. Recognition of On-line Arabic Handwritten Characters Using Structural Features. Journal of Pattern Recognition Research, 1; 2010. pp. 23-37.

[5] Zidouri, Abdelmalek. On Multiple Typeface Arabic Script Recognition. Research Journal of Applied Sciences Engineering and Technology, 2, 5; 2010. pp. 428-435.

[6] Kapogiannopoulos, George dan Kalouptsidis, Nicholas. A Fast High Precision Algorithm for the Estimation of Skew Angle Using Moments. In In: IASTED, International Conference Signal Processing, Pattern Recognition and Application, SPPRA (Crete, Grece); 2002. pp. 275-279.

[7] Amin, Adnan. Recognition of printed arabic text based on global features and decision tree learning techniques. Pattern Recognition 33; 2000. pp. 1309-1323.

[8] Hilditch, C.J. Linear Skeletons from Square Cupboards. In Machine Intelligence IV (B. Meltzer and D. Mitchie eds) (Edinburgh), University Press; 1969. pp. 403-420.

[9] Cowell, John dan Hussain, Fiaz. Extracting Features from Arabic Characters. In Proceedings of the IASTED International Conference on COMPUTER GRAPHICS AND IMAGING (Honolulu, Hawai, USA); 2001. pp. 201-206.