Proceeding of the International Conference on Advanced Science, Engineering and Information Technology 2011

Hotel Equatorial Bangi-Putrajaya, Malaysia, 14 - 15 January 2011

ISBN 978-983-42366-4-9

Generating an Arabic Calligraphy Text Blocks for Global Texture Analysis

Bilal Bataineh[#1], Siti Norul Huda Sheikh Abdullah[*2], Khairudin Omar[#3]

[#]*Center for Artificial Intelligence Technology,*
Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia.
[1]*bilal_bataineh82@yahoo.com,* [2]*mimi,* [3]*ko@ftsm.ukm.my*

Abstract— **This paper objective is to improve the current method for generating an Arabic Calligraphy text blocks. We test on seven types of Arabic Calligraphy text. We apply projection profiles and a proposed filter to discriminate each line of the Arabic Calligraphy scripts. After performing text detection, skew correction, text and line normalization subsequently, we generate Arabic Calligraphy text blocks for global texture analysis purposes. We compare our proposed filter with current method and median filter. The results show that the proposed filter is outperformed. The proposed method can be further improved to boost the overall performance.**

Keywords— **Arabic calligraphy, global analysis, Optical font recognition, image processing, text preprocessing.**

I. Introduction

Optical font recognition (OFR) is an area where document images are being analyzed and extracted in order to obtain important features or information. OFR researches have been conducted worldwide but their aims are varies such as document characterization and classification [1], document layout analysis [2], improvement the Multi-font OCR [3] and reprinting documents [4]. OFR or feature extraction technique consist of two categories: local analysis [4][5] and global analysis [1][6]. Local analysis is a feature extraction approach that concerns on disconnected parts in document images such as lines, words, sub words, characters and shapes. On the other hand, global analysis feature extraction approach stresses on a region of text block (two lines or more) in text geometric form.

Arabic language is one of international languages. It has been spoken as a mother tongue or first language in 22 Arabic countries around the world. Other countries such as Nigeria, Senegal and Turkey have been using Arabic as their second language. Besides that, other languages such as Jawi, Farsi, Kurdish, Pashto, Urdu, and Hausa, also adopt Arabic scripts. At the same time, the calligraphy of Arabic scripts is becoming more popular among the Islamic Art communities. There are eight main Arabic calligraphy types: Old Kufi, Kufi, Thuluth, Naskh, Roqaa, Diwani, Persian and Maghrebi. Examples of this calligraphy are shown in Fig 1(a) until (i) consecutively Page Layout.

From the figure above we can observe that all the font or calligraphy type have regular patterns but they are differs in terms of their features and properties. Each calligraphy type uses special history, regions and counters or unique Arabic scripts
positioning. Previously, the OFR researches have focused on the Latin [7] or Asian language [1][6].

Fig.1 The sentence "al-khat lesan al-yad ((the calligraphy is tongue of the hand))" written in the main Arabic calligraphy types (a) Diwani, (b) Kufi, (c) Thuluth, (d) Persian, (e) Roqaa, (f) Naskh, (g) Maghrebi, (h) Old Kufi.

Despite of the importance of the Arabic scripts and Arabic calligraphy, the existing Arabic OFR systems are still ignoring the other benefits of Arabic calligraphy definition such as classify the documents structures and that's aims, documents library, classify the documents history and reprinting the documents as the input image format.

Objective of this paper is to improve the current method for generating an Arabic Calligraphy text blocks. This paper is organized as follows. Section II reviews the state of art in global analysis feature extraction. Section III explains the proposed method and Section IV presents and analyses the experimental results. Finally, conclusions are presented in Section V.

II. State of the Art

Usually, the Arabic calligraphy text images are in poor quality. The text images may contain spaces between words, lines and boundaries. The text images may also posses different size Fig. 1(e), skew Fig 2 (b), ornamental forms and various color scale (Fig.2 (a), (c) and (e)) and dark background Fig (e). For that reason we unable to process the text image directly in the features extraction stage. To apply a global feature extraction techniques in calligraphy text, the calligraphy text have to prepare to be fit to use like texture blocks. That requires generating texture text by normalizing the text block.

Fig .2 The sample of natural Arabic calligraphy text, a) Roqaa, b) Roqaa, c) Persian, d) Persian and e) Kufi

Text normalization for global analysis feature extraction is still becoming an active research [6][8]. They concentrated on Latin and Asian printing languages. They applied Projection profile for segmenting each line before performing normalization technique. *Projection profile* is a straight projection profile either in one or two dimensional view which represents any region in an image. Let G is a binary image of $N \times M$ size, where N is the number of lines and M is the number of columns. If the P_v is a vertical vector generate from the sum of all black pixels for each M line where S is the pixel at (i, j) define as:

$$P_v(\mathrm{i}) = \sum_{j=1}^{M} \mathrm{S(i, j)} \qquad (1)$$

The projection profile results of Latin and Asian printing texts are well separated and organized for each line as shown in Fig. 3(a) and 3(b). Unlike Arabic calligraphy text, this approach is less significant because the projection profiles Fig. 3(c) are connected and sometime it creates another peak that can consider as a new line. Therefore, an extension method of projection profile is highly initiated for identifying texture blocks in Arabic calligraphy text.

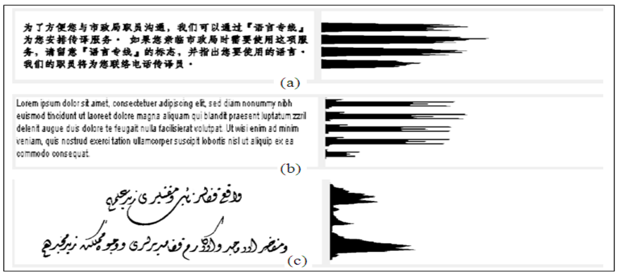

Fig.3 (a) Chinese text and its projection profile, b) Latin text and its projection profile and (c) Arabic calligraphy text and its projection profile.

In fundamental image processing area, median filter is useful to reduce the noise. It replaces the center pixel Fig. 4(a) into a median value from a fixed kernel like 3x3 which consists of a serial of ascending or descending pixel values Fig.4(b). For example Fig.4, the center pixel and its neighbour's pixel's values are (0, 1, 2, 3, **4**, 11, 19, 23, and 97). The value 4 is the median value. Median filter can also be used to discriminate the text lines into texture blocks Fig. 5(a). However, there are some cases where the median filter can also cause error such as Fig.5(b). As we can se In conclusion, a modification filter method should be considered.

#	#	#
#	4	#
#	#	#

(a)

11	1	0
2	97	4
23	3	19

(b)

Fig .4 (a) The result median filter for (b)an example region using 3×3 kernel.

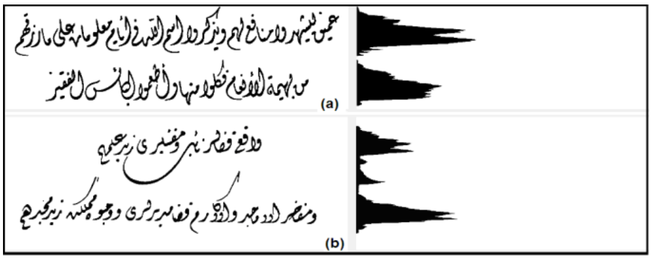

Fig .5 Examples of (a) good Diwani calligraphy projection profile and (b) bad Diwani calligraphy projection profiles after applying median filter 3×3 kernel.

III. The Proposed Method and Texture Block Proposed Framework

The first subsection describes the proposed method for generating Arabic Calligraphy texture blocks. Next, the second subsection explains on OFR proposed framework.

A. The proposed method

We extend the projection profile and median filter method by applying a proposed filter called proposed filter. The proposed filter is almost similar to median filter. However, we assume that the entire pixels in the kernel 3×3 (the region) will take the same value. We count the total number of white pixel in that 3×3 kernel. Then, we convert the binary image into black or white based on majority pixels in that kernel. Here we consider, ‘0’ is black and ‘1’ is white.

Let us assume G in a binary image. G_R is a region in image G using $N \times M$ kernel size. We define the proposed filter as below flowchart in Fig.6.

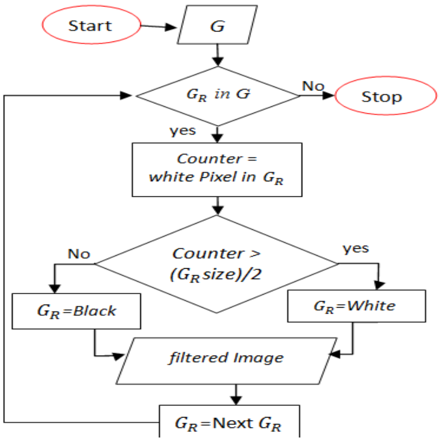

Fig.6 The proposed Filter Flowchart.

We show how the proposed filter works in Fig.7 by using a 3×3 kernel matrix. Fig.7a (i) is the source image pointing to a 3×3 region. Since there is more white pixels in that kernel, therefore Fig.7(a (ii)) converts all the values in the region into white. On the other hand, Fig.7(b(i)) converts into black Fig.7(b (ii)) when more black pixel exists in the region.

We discuss the experimental and image results after applying proposed filter in the next section.

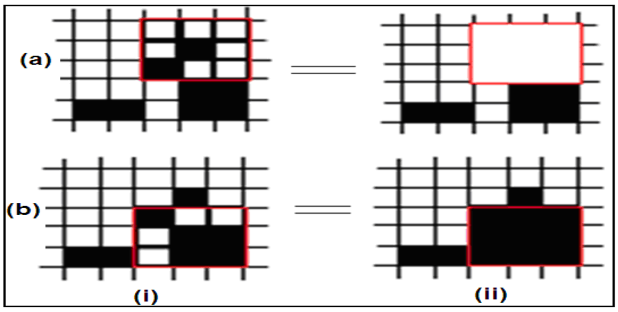

Fig .7 (i) A source binary image containing a 3×3 region, (ii) An image result after applying proposed filter.

B. *The texture block proposed framework*

Quite often that existing noise in an image will distract the document analysis. Since the Arabic Calligraphy text consists variety of notations or marks on the image, it is impossible to discriminate the important information from an image without applying any image enhancement methods. For that reason, we employ different methods to reduce, remove and prepare the image before undertaking the feature extraction process. In this research we perform global texture features extraction. The texture block proposed framework is shown as in Fig.8. This proposal consists of image pre-processing and text normalization major steps.

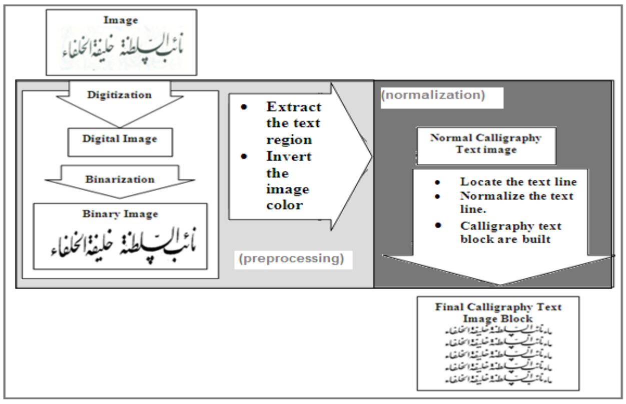

Fig.8 The proposed framework of Arabic calligraphy texture block.

C. *Image Pre-processing*

1) *Text image Binarization*

The source image can be either in color or grayscale. We apply the same Binarization method as [6] [8] to ease the text block detection. This Binarization process converts the color or grayscale image into binary image (0, 1) which we represents 0 as background and 1 as the foreground or the text. The advantage of this process is it decreases in the image size and processing time. Furthermore, the subsequent process can become less complex . Fig.9 shows the example of a color source image converted into binary format.

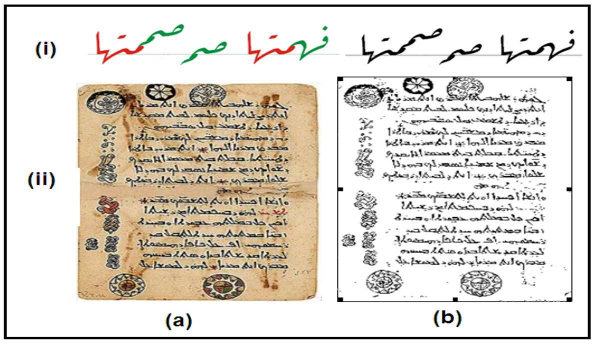

Fig.9 The source, and (b) binary image of "(i) Roqaa", (ii)" Kufi"

2) *Text region extraction*

Some documents contain pictures, graphs, samples or many other types of non-textual texture. Some works implement the text extraction process manually [9] and others apply automatically [11]. Therefore, at this stage we remove non-text foreground manually. Since this paper objective is to extract the calligraphy text region purposes, we exclude the meaning, grammar or ordering of the scripts. An example of text extraction is shown in Fig.10.

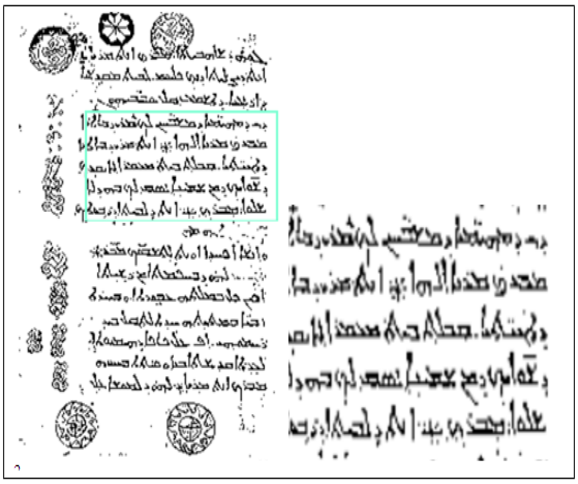

Fig.10 (Right) The selected of “Kufi” image text is in the bounding box, and (Left) the expansion of the text region.

3) Skew Correction

Unlike other scripts, the Arabic script is written in horizontal lines from right to left. Due to improper scanning process i.e wrong alignment, some script image may suffer skew problem. Apart from that, the Arabic Calligraphy can also be written and presented in artistic way which involves skewness. In order to realign the script, we use Hough Transform [10]. However, this approach is less prone to realign the script image which is extremely skewed. We flip the script image first before performing Hough transform process. An example of a skewed Arabic scripts and its correction image after applying Hough Transform technique are illustrated in Fig.11 (a) and (b) correspondingly.

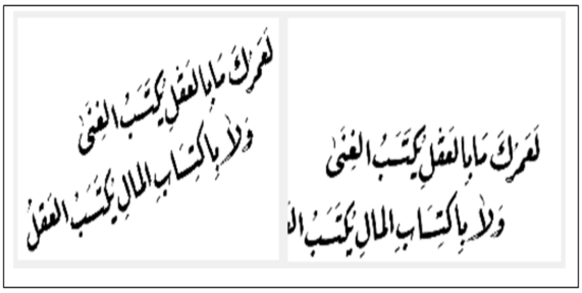

Fig.11 (Right) The source Roqaa calligraphy image, and (Left) the result image after applying skew correction using Hough transform technique.

4) Image inversion

In some cases in Arabic Calligraphy images, the background is darker than the foreground or the text region. Firstly, we invert only image that contains more dark pixels compared to the light pixels. This process is employed before proceeding to the normalization stage. Examples of the source and the inversion image are depicted in Fig.12.

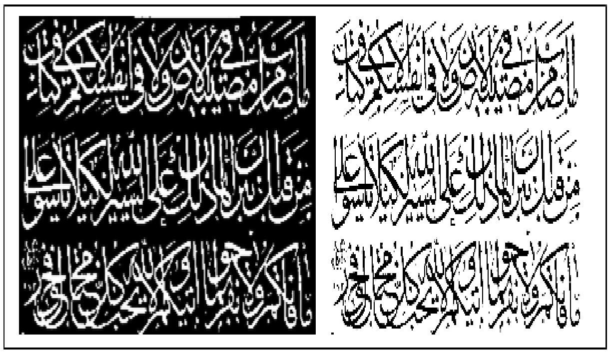

Fig.12 (Right) The source image of Thuluth calligraphy and (Left) the inversion image.

D. Text Normalization

Similar to Latin, each line of Arabic scripts is written horizontally but when reading, we read them from right to left. These lines consist of separated words and sub words that relate to each other. Some Arabic scripts have long, shot, connected and disconnected lines. Pertaining to those matters, text normalization step is a penalty subsequent step after employing skew correction and image inversion processes correspondingly.

1) Text Lines Location

We determine the lines by finding the lines boundary. Quite often other research used the projection profile technique to locate the text lines in the document images [6] [8]. This approach can be easily carried out on Chinese or Latin image documents because those scripts or letters have consistent height. As a result, they could achieve successful accuracy only by applying the projection profile technique. On the other hand, Arabic calligraphy scripts are normally consists of connected extension between two or more lines which are artistically written. Examples of those Arabic letters are ا ,ك ,و ,م , ل, ر and ز. Therefore, locating the actual text lines is a vital issue.

We introduce the proposed filter to solve the above issue. Since the proposed filter is a modification of median filter, we also compare their results in the next section. Normally, the extended letters expand in vertical form. We scan the foreground from right to left horizontally and recomputed all the pixels values in the region based on white or black majority values.

We show an example of a source image for Persian calligraphy with four lines in Fig.14(a) and the image results after performing proposed filter in Fig.14(c). While their projection profiles with and without proposed filter are shown in Fig. 14(b) and (d) subsequently. We can observe that the expanded areas is removed in Fig.14(a) and its the projection profile has shown that three or four potential lines exist in the image. We add new rules to justify and discriminating the fuzziness of extracting line 2 and 3. We employ Fig.13 process, thus the final four text line are separated and determined successfully as depicted in Fig.14(e). It removes the expanding area and keeps the text line body. In general proposed filter can produce a better solution for locating text line.

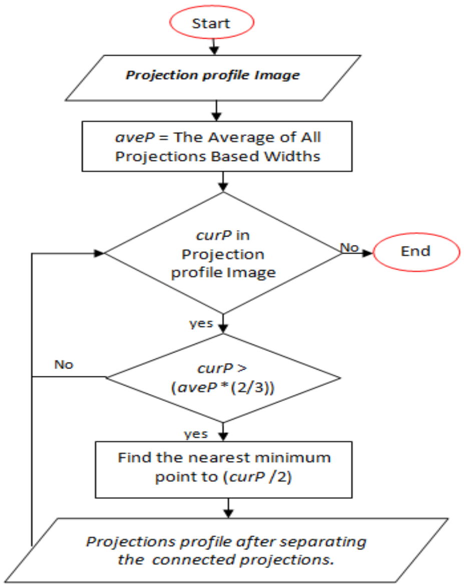

Fig.13 The proposed Filter Flowchart.

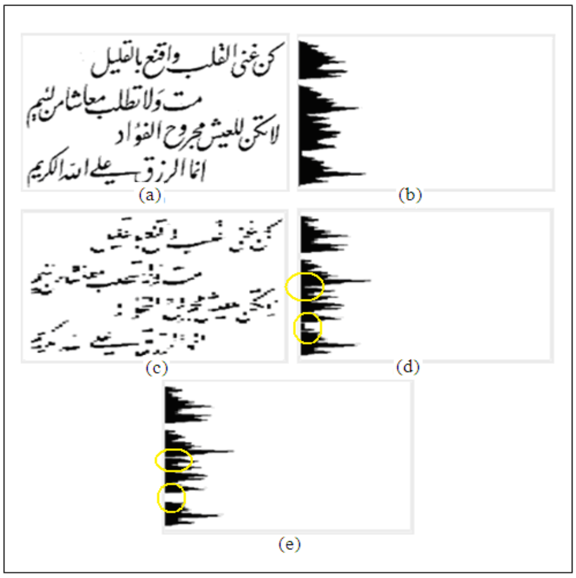

Fig.14 (a) The source image of "Persian calligraphy" image, (b) the projection profile before applying proposed filter, (c) the source image after applying proposed filter, (d) the projection profile after applying proposed filter, and (e) the final projection profile after applying Algo . 2.

2) Line Normalization

In order to produce optimum quality for the final Arabic Calligraphy art, we choose 512×512pixel image size [7]. We also estimate that five lines of each selected Arabic Calligraphy text to be printed on that image size. Upon several experiments, we identify that the Arabic Calligraphy's quality is well suited using above approaches for global texture analysis.

Consequently, after finish executing the text line location process, we can distinguish the text horizontal lines easily. Now, we must determine the individual word in vertical way. The gaps of some Arabic words are inconsistent. We apply only vertical projection profiles to calculate the gaps between words. Then we recalculate the new gaps based on the average gap's distance and normalize each word to fit on 100×512 pixel image size. The transformation of this stage is shown in Fig. 15.

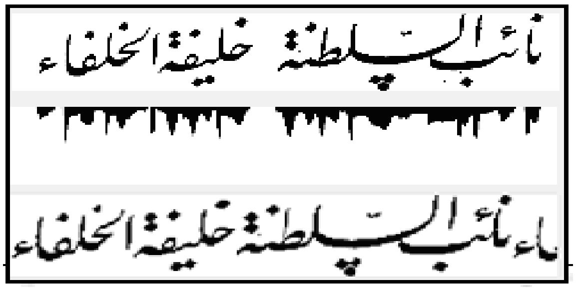

Fig.15 (Top) The vertical projection profile of Persian calligraphy image text and (Bottom) the image result after reorganizing the gaps to fit on 100×512 pixel image size .

3) Text Block Normalization

Next stage is the text block normalization. We normalize each selected text lines to fit on 512×512 pixel image size in bitmap format. We construct five lines for each text block. We build the text block by padding each other. We also insert two pixel spacing within each text block to avoid from interlacing or connecting between image patterns. As a result, the final Arabic Calligraphy product has consistent spacing vertically and horizontally as shown in Fig.16.

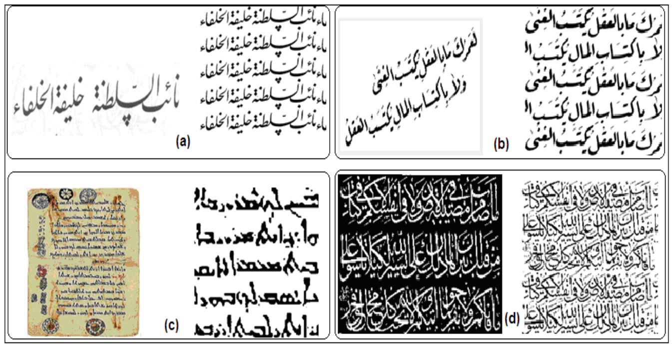

Fig.16 a) Diwani , b)t Roqaa , c) Kufi and d) Thuluth in the final result of text normalization

IV. Experimental Results

Our sample dataset is from a mixture of resources such as books, artistic works from calligraphy software product and internet. Their image size and format are various. We have collected 100 image samples which consist of 20 samples from each type: Kufi, Diwani, Persian, Roqaa, Thuluth and Naskh. The entire images contain text and non-textual information, different skews and color scales. We develop this framework proposal in VB. Net 2005. The final image results are in bitmap format in 512×512 pixel image size.

We conduct four experiments to achieve our main objective which is to generate Arabic Calligraphy text blocks for global texture analysis. We compare proposed filter with median filter with 3×3kernel size and without proposed filter as the control [6][8]. We also analyses the optimum proposed filter kernel size: $3 \times 3$and 5×2 kernel size. Table 1 and Fig. 17 show the experimental results.

From Table 1 and Fig. 15, we can summarize that by using 3×3 kernel size, the average accuracy rate for the proposed filter increases approximately 12% compared to the median filter and without proposed filter. On top of that, the proposed filter can also boost up the average accuracy rate by applying 5×2 kernel size. It produces the highest average accuracy rate about 95.7% while the rests only achieve at about 80.7% to 92.85%. The proposed filter performs better in 5×2 kernel size because mostly the Arabic Calligraphy letters are wider compared to other language letters. Even though, median filter is powerful for 'salt and pepper' image problem, but it is less prone to discriminate the text blocks in Arabic calligraphy.

We can also observe that the Diwani and Persian calligraphy dataset produce the least accuracy about 30-90/% compared to other type Arabic calligraphy scripts. Diwani and Persian are more suitable to use only projection profile approach with the proposed filter in 5×2 kernel size. This may be due to Diwani and Persian letters are normally more skew, thin and expand in horizontal and vertical. They may lead to unessential splitting lines. In summary, the projection profile can perform better results by employing proposed filter when discriminating the text lines in Arabic Calligraphy scripts.

Table1
The Percentage of Projection Profile Correction With and Without The Proposed Filter

Kernel size	**3x3**	**3x3**	**5x2**	**3x3**
Arabic Calligraphy Type	Without Proposed Rate	Proposed Rate	Proposed Rate	Median Rate
Kufi	100%	100%	100%	100%
Thuluth	80%	85%	95%	80%
Diwani	30%	85%	90%	55%
Naskh	90%	95%	95%	90%
Persian	65%	85%	90%	50%
Roqaa	100%	100%	100%	100%
Magh.	100%	100%	100%	90%
Average accuracy	80.7%	92.85%	95.7%	80.7%

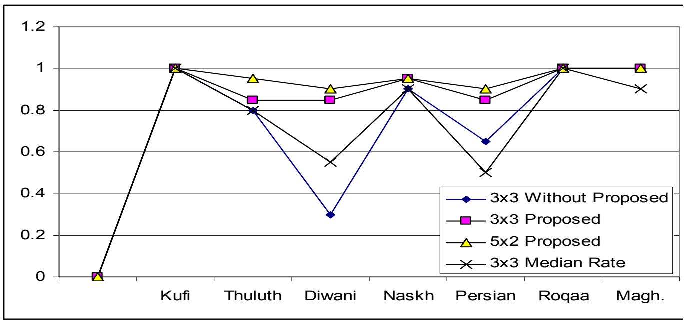

Fig.17. The graph of all Arabic Calligraphy types versus accuracy rate.

V. Conclusion

After image pre-processing, we perform text normalization. We propose combination of horizontal projection profile and a proposed filter to increase the accuracy of text line discrimination. Besides that the text normalization processes are also involves reorganizing the vertical spacing within texts and lines by using vertical projection profile. Finally, we can generate full calligraphy text blocks. These text blocks are meant for global texture analysis in OFR system later. The proposed method can also show improvement when applying combination of projection profile and proposed filter approaches in discriminating any continuous data signal processing.

Acknowledgment

We wish to thank the UKM University for the research grant UKM-OUP-ICT-36-186/2010 and UKM-GGPM-ICT-119-2010 and FRGS UKM-TT-03-FRG012-2010

References

[1] X. Ding, L. Chen, T. Wu, Character Independent Font Recognition on a Single Chinese Character, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 29, NO. 2. 2007.

[2] I. Abuhaiba , Arabic Font Recognition using Decision Trees Built from Common Words, Journal of Computing and Information Technology - CIT 13, 2005, 3, 211–223. 2005.

[3] S. Manna, A. Colla AND A. Sperduti, Optical Font Recognition for Multi-Font OCR and Document Processing, IEEE, 10th International Workshop on Database & Expert Systems Applications, Florence, Italy. 1999.

[4] A. Zramdini and R. Ingold, Optical Font Recognition Using Typographical Features, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 20, NO. 8. 1998.

[5] Hung-Ming Sun, Multi-Linguistic Optical Font Recognition Using Stroke Templates, icpr, vol. 2, pp.889-892, 18th International Conference on Pattern Recognition (ICPR'06) Volume 2. 2006.

[6] Y. Zhu, T. Tan, and Y. WANG, Font Recognition Based On Global Texture Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 10, pp. 1192-1200. 2001.

[7] V. Juan, A. Carlos, FONT RECOGNITION BY INVARIANT MOMENTS OF GLOBALTEXTURES, Very Low Bit-Rate Video-Coding 2005 (VLBV05) International Workshop. 15-16 September 2005. Sardinia, Italy. 2005.

[8] A. Busch, W. Boles, S. Sridharan, Texture for Script Identification, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 27, NO. 11. 2005.

[9] G. Joshi, S. Garg, J. Sivaswamy, A generalized framework for script identification, International Journal on Document Analysis and Recognition Volume 10 , Issue 2 ,Pages: 55 – 68, ISSN:1433-2833. 2007.

[10] Mackenb," How to deskew an image", http://www.codeproject.com/KB/graphics/Deskew_an_Image.aspx, October, 2009.

[11] C. Gopalan and Manjula, Text Region Segmentation From Heterogeneous Images, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.10, October 2008.