Printed Arabic Sub-Word Recognition Using Moments

Ibrahim A. El rube', Mohamed T. El Sonni, and Soha S. Saleh

Abstract**—**the cursive nature of the Arabic writing makes it difficult to accurately segment characters or even deal with the whole word efficiently. Therefore, in this paper, a printed Arabic sub-word recognition system is proposed. The suggested algorithm utilizes geometrical moments as descriptors for the separated sub-words. Three types of moments are investigated and applied to the printed sub-word images after dividing each image into multiple parts using windowing. Since moments are global descriptors, the windowing mechanism allows the moments to be applied to local regions of the sub-word. The local-global mixture of the proposed scheme increases the discrimination power of the moments while keeping the simplicity and ease of use of moments.

Keywords**—**Arabic sub-word recognition, windowing, aspect ratio, moments.

I. Introduction

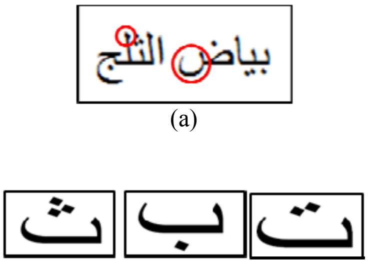
ARABIC word recognition is gaining a lot of interests these days trying to find a promising improvement. The Arabic language is widely used; there are more than 200 million people, who speak the Arabic language. Sometimes we call the written Arabic "Modern Standard Arabic" (MSA), which is a standardized version used for official Arabic communication across the Arab world. The Arabic characters are similar to the character of the Farsi (Persian), Crude and Urdu languages [1]. Keeping the documents on papers is posing a lot of difficulty in storage, retrieval, searching and updating, while the electronic documents makes all that easy for us [2]. The Arabic words recognition can be defined as the transformation of text representation in the spatial form of graphical marks into its symbolic representation [3].

In Arabic word recognition systems, a printed typed document is scanned and used as an input to the system. Then the document is prepared for processing; this stage is called preprocessing stage which includes noise removal, binarization and baseline estimation. The document is then segmented into lines and the lines are segmented into words, sub-words, or characters by using horizontal and vertical projections. In this paper, windowing applied to each extracted sub-word image, which segments the image into multiple smaller parts. For each part, three types of moments (central, Hu, and Zernike moments) are tested and compared. Similarity measure using correlation is used for obtaining similar sub-words in a given dataset.

This paper is organized as follows; First, Arabic writing characteristics are introduced. Second, the proposed Arabic sub-word recognition system is explained. Third, the dataset preparation and the experimental results are discussed. Finally, the conclusion and future work are drawn.

II. Arabic writing Characteristics

Arabic writing is unlike English, it is written from right to left, and it consists of 28 characters [4]. Characters do not contain upper and lower case and their shape depend on its position in the word (isolated, beginning, median and end) [5]. The Arabic letters may be connected from one side only or both sides, which depend on the word itself, also each word, may be composed of one unit (connected characters) or more. Some letters contains “ascenders” and “descenders”, as shown Fig. 1(a) [1]. Arabic characters may have the same body shape but differ in the number and position of the dots, Fig. 1(b) [6].



(a)

(b)

Fig. 1 a) 'descenders' and 'ascenders' in Arabic writing, b) examples of Arabic letters having the same body but with different number of dots.

There are only six letters that have two shapes of writing, either isolated or final, which are “د” “D”, “ذ” “Z”, ”و” “W”, “ا” “A”, “ر” “R”, “ز” “Z”.

III. Arabic-Sub-Word Recognition System

After the printed document containing text is scanned, some preprocessing tasks are performed, such as binarization, noise removal and baseline estimation. The next stage segments the document into lines then into sub-words using horizontal and vertical projections. Windowing is then applied to the sub-word images, which will result in dividing each sub-word image into multiple smaller parts. A feature vector is constructed from concatenation of the moments applied on

Authors are with Arab Academy for Science, Technology, and Maritime Transport (AASTMT), Alexandria, Egypt
e-mail: ielrube@yahoo.com, sooha_salah@hotmail.com

each window of the image. Fig. 2 shows the flow of feature extraction of sub-word images.

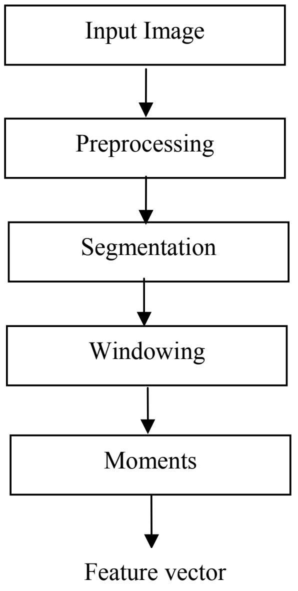


Fig. 2 Sub-word feature extraction block diagram.

A. Preprocessing

The preprocessing stage includes binarization, noise removal, and baseline estimation using horizontal projection. Fig. 3 shows an example of baselines obtained for two different sub-words that compose a complete word in Arabic text.

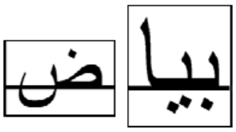


Fig. 3 Example baseline estimation of the word “بياض”.

B. Word Segmentation:

Most of the researchers segmented the Arabic word into characters, such as [5], [8] and [9]. Others used the word without segmentation, such as [10] and [11]. However Mona et al [7] introduced sub-words segmentation algorithm for multi-font texts based on conditional labeling of up and down contours using horizontal and vertical projection. Also, Jawad et al [12] segmented the words into sub-words using vertical histogram and connected component analysis. The recognition rates obtained when no segmentation was used, was low (approximately 80%) [11]. On the other hand, segmenting the words into characters performed a higher recognition rates, but this segmentation suffers from segmentation problems such as over segmentation, under segmentation or misplaced segmentation which affects the recognition performance in a negative way. Therefore, in our work we use the separation between connected characters of each word to obtain naturally segmented sub-words (connected parts). Sub-words may consist of one character or multiple characters.

After the preprocessing stage, the document is segmented into lines text using horizontal projection, and then these lines text are segmented into sub-words by using the vertical projection as shown in Fig. 4.

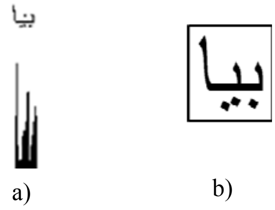


Fig. 4 a) Vertical projection of a sub-word, b) segmented sub-word image.

The Bounding Box is then applied to each sub-word to remove the unnecessary pixels which may have negative effect on the results. However a two pixel width has been added to the bottom of the sub-word images in case of baseline extraction of the sub-word images as shown in Fig. 5.

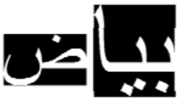
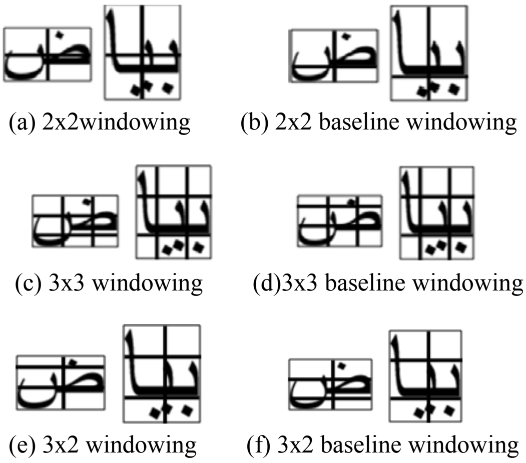


Fig. 5 Sub-word images cropping using Bounding Box with two lines added at the bottom of the image.

C. Windowing

Windowing is applied on each sub-word image, resulting in dividing the sub-word image into multiple segments. We examined 2x2, 3x2, and 3x3 windows; the windows either divided equally or divided with respect to the baseline. Fig. 6 shows the used windowing layouts with and without baseline detection.



(a) 2x2windowing (b) 2x2 baseline windowing

(c) 3x3 windowing (d)3x3 baseline windowing

(e) 3x2 windowing (f) 3x2 baseline windowing

Fig. 6 Different windowing layouts are investigated.

D. *Alef "ا" Windwing*

The character Alef "ا", has a special layout, which has to be treated in a different way, other than the rest of the character images and sub-word images. This letter has very small width, therefore there is no need to divide it vertically if the width is smaller than certain number of pixels.Therefore, only two equally windows should be used (as shown in Fig. 7). This arrangement increases the recognition rate of the character "ا" to 100% compared to a recognition rate 50% before defining a special window for this character image.

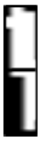


Fig. 7 Alef Windowing.

E. *Moments*

Three different types of moments are tested and compared in this work.

Central moments: The first tested moments are Central moments which are invariant to scale and translation only. The two-dimensional central moments μ_{pq} of order *p* and *q* for a (*N* x *M*) discretized image, *f* (*x* , *y*), is

$$\mu_{pq} = \sum_{x=0}^{M}\sum_{y=0}^{N}(x-\bar{x})^p(y-\bar{y})^q f(x,y) \tag{1}$$

Hu moments: The Hu moments are invariant under general linear transformations. However, the Hu moments are not orthogonal, so there is redundancy in the information they capture. Hu defined seven values, computed from central moments through order three,

$$\begin{aligned}
M_1 &= (\eta_{20} + \eta_{02}),\\
M_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2,\\
M_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2,\\
M_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2,\\
M_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]\\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2],\\
M_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}),\\
M_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]\\
&\quad - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned} \tag{2}$$

where

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}$$

Zernike moments: Zernike moments introduce a set of complex polynomials that formed a complete orthogonal set over the interior of a unit circle. They can be calculated to whatever order is desired. The order of the Zernike is determined by the value of ***n*** in the below equation [13]. The Zernike polynomials, V_{nl} (*x* , *y*), of order *n* , are defined by:

$$V_{nl}(x,y) = R_{nl}(r)\ e^{il\theta} \tag{3}$$

where

$$0 \ \le\ l\ \le\ n \quad n\ -\ l\ =\ even$$

The real-valued radial polynomial is defined by:

$$R_{nl}(r) = \sum_{m-0}^{(n-l)/2}(-1)^m \frac{(n-m)}{m\,![(n-2m+l)/2]\,!\,[(n-2m-l)/2]\,!} r^{n-2m} \tag{4}$$

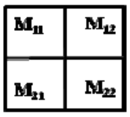
The Zernike moment Z_{nl} is defined as

$$Z_{nl} = \frac{(n+1)}{\pi}\int_0^{2\pi}\int_0^{\infty} V_{nl}(r,\theta)^*\ f(r,\theta) r dr d\theta \tag{5}$$

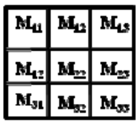
Where * indicates the complex conjugate

The main advantage of the Zernike moments is that they are invariant to rotation and they can be made scale invariant by normalization. In addition, Zernike moments are robust to noise and minor variations in shape.

Feature vectors are obtained by applying Zernike moments to each segment of the sub-word image resulted from the windowing stage (as shown in Fig. 8).



2x2 Feature Vector = [M_{11}; M_{12}; M_{21}; M_{22}]



3x3 Feature Vector = [M_{11}; M_{12}; M_{13}; M_{21}; M_{22}; M_{23}; M_{31}; M_{32}; M_{33}]

Fig. 8 Feature vectors are obtained by applying moments to each segment of the sub-word image.

IV. MATCHING ALGORITHM

The matching and recognition processing is carried out after filtering the dataset with respect to the aspect ratio of the query image. Only features of the sub-words that passed the filtration stage are considered for matching with the query image. Fig. 9 shows the flow of the matching phase until sub-words similar to the query image are obtained from the dataset.

A. *Aspect Ratio Filtration*

Usually similar sub-words are close in aspect ratio even if the scale (font size) is not the same. Therefore, the aspect ratio of the query sub-word image is calculated and only the sub-word images close to query aspect ratio are included (as shown in Fig. 10).

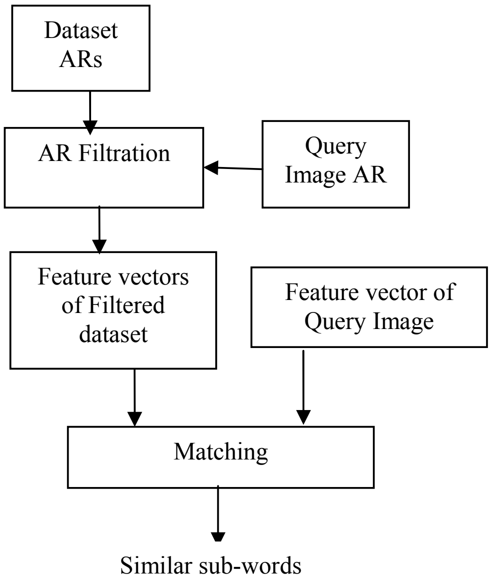

Dataset ARs
AR Filtration
Query Image AR
Feature vectors of Filtered dataset
Feature vector of Query Image
Matching
Similar sub-words

Fig. 9 Sub-word recognition block diagram. AR is the Aspect Ratio.

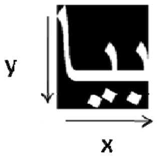


Fig. 10 Aspect ratio calculation.

The aspect ratio is defined as:

$$\text{Aspect Ratio} = x / y \tag{6}$$

where

x= number of columns and
y= number of rows.

The aspect ratio is expected to save much time without affecting the performance of recognition rates. In this paper, the filtration is done by allowing sub-words that have a difference of 0.25 or less than the aspect ratio of the query image which filtered around 66% of the whole dataset.

B. Matching

In the matching stage the extracted features of each segment of the sub-word query image are compared with the feature vector of the equivalent segments of the sub-word image that needs to be matched. The correlation coefficient equation is defined as

$$r_k = \frac{\sum_{t=1}^{N} (x_t - \bar{x})(y_{t+k} - \bar{y})}{\sqrt{\sum_{t=1}^{N} (x_t - \bar{x})^2 \sum_{t=1}^{N} (y_t - \bar{y})^2}} \tag{7}$$

where x_t is the feature vector of the query sub-word and is the y_t feature vector for another sub-word, and $\bar{x}$ are the $\bar{y}$ means of and x_t y_t respectively.

The images are then ranked in the descending order giving us the closest recognized sub-word images. The recognition rate is calculated by counting the number of similar sub-word images that are correctly recognized and divide them by total number of relevant images in the whole dataset, then multiplied by 100. Thus recognition rate equation is given by:

$$R = \left(\frac{\#\text{similar retrieved images}}{\#\text{similar images in Dataset}} \right) \times 100 \tag{8}$$

V. Experimental Results

A. Dataset

The dataset consists of 200 different sub-word images written in the same font "Times New Roman". Each sub-word is repeated four times in (10, 11, 12, and 14) font sizes. Fig. 11 shows samples of the sub-words that compose the dataset used in the experiment. The documents are printed, and then scanned and input to the system. Each group is binarized and then segmented using the horizontal and vertical projections. After that, the bounding box is applied to each sub-word.

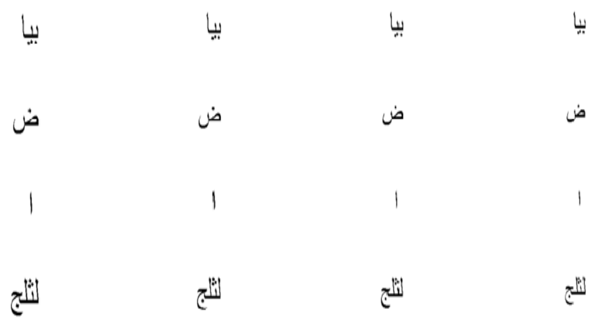


Fig. 11 Sample of sub-word images in the dataset.

B. Evaluation

Several experiments are applied to the dataset using different moments with different window layouts.

In the first experiment, three types of moments are applied separately on the dataset sub-words without applying any windowing. It is known that low order moments have low recognition rates, while higher order moments have better recognition rates with the expense of higher computation time. Therefore, moments of orders 3 or 4 could be more suitable for both complexity and recognition rates than low or high order moments.

In the second experiment, different window layouts are applied to the sub-word images of the dataset. All windows are 10% overlapped with the adjacent segments. Two types of windows are tested; the first window divides the sub-word images into equal parts and the other window type divides the sub-word images with respect to the baseline. Furthermore, different window layouts are applied; 2x2, 2x3, and 3x3 windows. Table I indicates the recognition rates of the three tested moments; central, Hu, and Zernike moments, with and without windowing. It is clear form the results of Table I that windowing increases the discrimination power of the moments.

TABLE I
RECOGNITION RATES OF DIFFERENT TYPES OF MOMENTS WITH DIFFERENT WINDOWS APPLIED

Windowing Type	Recognition Rate		
	Central Moments	Hu Moments	Zernike Moments
No Windowing	78.8%	72.6%	61.9%
2x2	92.1%	87.1%	89.4%
2x2 Baseline	84.8%	87.3%	84.7%
3x3	90.5%	91.8%	95.2%
3x3 Baseline	87.6%	92.3%	90.3%
3x2	88.9%	86.8%	95.1%
3x2 Baseline	86.8%	87.8%	93.6%

Fig. 12 shows an example of ten retrieved images arranged from most similar (1) to the lease one (10) for a query images "لثلج".

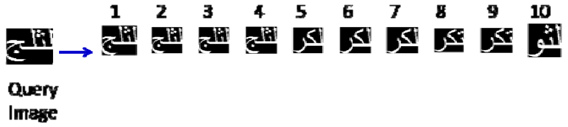


Fig. 12 Example of retrieved sub-word images for a given query image.

Since Zernike moments are scale invariant, the performance of the Zernike moments is improved by resizing the sub-word images before calculating the moments. Table II shows the recognition rates for the three types of moments after resizing the sub-word images (100x100 pixels) and when using different windowing layouts.

TABLE II
RECOGNITION RATES OF DIFFERENT TYPES OF MOMENTS WITH DIFFERENT WINDOWS APPLIED AFTER RESIZING SUB-WORD IMAGES

Windowing Type	Recognition Rate		
	Central Moments	Hu Moments	Zernike Moments
2x2	95.9%	91.4%	99.1%
2x2 Baseline	91.9%	87.2%	95.3%
3x3	89.9%	91.2%	**99.8%**
3x3 Baseline	90.2%	91.8%	92.5%
3x2	93.6%	92.4%	**99.8%**
3x2 Baseline	90.1%	87.1%	96.3%

Image resizing significantly improved the results of the Zernike moments using uniform windowing (shown in Table II). The maximum recognition rate obtained was 99.8% for Zernike moments with 3x3 and 3x2 windowing and after resizing the sub-word images to a fixed size. For Central and Hu the change of the recognition rates is not that significant since they are scale invariant.

VI. CONCLUSION AND FUTURE WORK

A printed Arabic sub-word recognition system is introduced in this paper. The proposed algorithm uses a new idea of segmenting the sub-word images into multiple parts using windowing, then applying the moments. The recognition rates obtained indicate that the moments and windowing layout are important aspects for obtaining high recognition rates. In the experiments, Zernike moments performed better than other moments after resizing the input image to a fixed size. This algorithm can be applied to handwritten images with some modifications to segmentation and windowing layout to fit the characteristics of the handwritten sub-words. Also, the system could be made more robust to noise and outliers when using the obtained feature vectors with neural network for sub-words recognition.

REFERENCES

[1] Lorigo, L.M. Govindaraju, "Offline Arabic handwriting recognition: a survey", IEEE Computer Society; pp. 712-724, vol. 28 no. 5, 2006.

[2] Jamshid Shanbehzadeh, Hamed Pezashki, Abdolhossein Sarrafzadeh, "Feature Extraction from Farsi Handwritten letters", Proceedings of Image and Vision Computing New Zealand 2007, pp. 35–40, Hamilton, New Zealand, December 2007.

[3] Aburas, Abdurazzag Ali Gumah, Mohamed E., "Arabic Handwriting Recognition: Challenges and Solutions", International Symposium on Information Technology, ITSim, 26-28 Aug. 2008, vol 2, pages 1-6, 2008.

[4] Al-Hajj Mohamad, R.; Likforman-Sulem, L.; Mokbel, C.; "Combining Slanted-Frame Classifiers for Improved HMM-based Arabic Handwriting Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31, No. 7, pp. 1165-1177, 2009.

[5] Abdulaziz, E.Alsaif, K.I., "Radon Transformation for Arabic Character Recognition", International conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, 13-15 May 2008, pages 433-438, 2008.

[6] Aburas, A.A.; Rehiel, S.A., "Off Line Tool for Arabic Handwritten Character Recognition Based On JPEG2000 Image Compression", ICTTA, Kuala Lumpur, Malaysia, 7-11 April 2008, pp. 1 – 5, 2008.

[7] Mona Omidyeganeh, Reza Azmi, Kambiz Nayebi and Abbas Javadtalab, "A New Method to improve Multi Font Farsi/Arabic Character Segmentation Results: Using Extra Classes of Some Character Combinations", Multimedia Modeling Conference, Nanyang Technological University, Singapore MMM2007, pages 670-679, 2007.

[8] Zidouri, "A PCA-based Arabic Character feature extraction", 9th International Symposium on Signal Processing and Its Applications, Sharjah, UAE, 12-15 Feb. 2007, pp. 1 – 4, 2007.

[9] Mahmoud, S.A. Mahmoud, A.S., "Arabic Character Recognition using Modified Fourier Spectrum (MFS)", Cybernetics and Systems, Vol. 40, No. 3, April 2009, pp. 189 – 210.

[10] Gregory R Ball, Sargur N Srihari, Harish Srinivasan, "Segmentation-Based And Segmentation-Free Methods for Spotting Handwritten Arabic Words", Tenth International Workshop on Frontiers in Handwriting Recognition, CEDAR 2006.

[11] Sargur N. Srihari, Gregory R. Ball and Harish Srinivasan. "Versatile Search of scanned Arabic Handwriting", Arabic and Chinese Handwriting Recognition, LNCS 4768, Springer 2008, pp. 57-69, 2008.

[12] Jawad H AlKhateeb, Jianmin Jiang, Jinchang Ren, and Stan S Ipson, "Component-based Segmentation of Words from Handwritten Arabic Text", World Academy of Science, Engineering and Technology, vol. 41, pp.344-348, 2008.

[13] Gheith Abandah, Nasser Anssari, "Novel moment features extraction for recognizing handwritten Arabic letters", Journal of Computer Science, vol. 5, No. 3, March, 2009, pp. 226-232.