Arabic CWR Based on Correlation of Normalized Signatures of Words Images

Hala S. Zaghloul, Taymoor Nazmy
Faculty of computer science, Cairo, Egypt

ABSTRACT

The traditional methods for Arabic OCR (AOCR) based on segmentation of each word into a set of characters. The Arabic language is of cursive nature, and the character's shape depends on its position in the word. There are about 100 shape of the characters have to be classified, and some of them may be overlapped.

Our approach use a normalized signature of the time signal of the pulse coupled neural network PCNN, supported with some shape primitives to represent the number of the word complementary and their positions within the image of the word. A lookup dictionary of words with its signatures was constructed, and structured in groups using a decision tree.

The tested signature was routed through the tree to the nearest group, and then the signature and its related word with higher correlation within the selected group will be the classified. This method overcome many difficulties arise in cursive word recognition CWR for printed script with different font type and size; also it shows higher accuracy for the classification process**,** 96%.

KEYWORDS: *Arabic CWR, PCNN, dots classification, decision tree.*

1. INTRODUCTION

Recent advances in printed document digitization and processing led to large scale digitization efforts of legacy printed documents producing document images. To enable subsequent processing and retrieval, the document images are often transformed to character-coded text using Optical Character Recognition (OCR). Although OCR is fast, OCR output typically contains errors. The errors are even more pronounced in OCR'ed Arabic text due to Arabic's orthographic and morphological properties. The introduced errors adversely affect linguistic processing and retrieval of OCR'ed documents.

Most cursive script recognizers, segment the words into characters [1-10], either prior to recognition or during recognition. Whole of word recognition removes the needs for segmentation of the word into characters, eliminating problems associated with poor placement or missing segmentation points. The clear problem with this is that instead of a finite alphanumeric vocabulary, an unbounded word vocabulary is needed for the unrestrained case. However, in many cases, the application context means that there will be a strictly finite number of words in the application vocabulary. Therefore word recognition becomes feasible. Some examples are signature recognition and the words indicating the dollar amount on a cheque.

According to a survey on off-line cursive word recognition [7], features useful in recognition of off-line segment-free recognition of cursive word recognition can

be classified into three categories based on representation level, ie:

1. Low level
2. Medium level
3. High level

Low level features include, smoothed traces of the word contour, pieces of strokes between anchor points, edges of the polygonal approximation etc. Medium level category is an aggregation of low level features to serve as primitives. Medium level features are continuous in nature in contrast to low level features. High level features are holistic or global features such as ascenders, descenders, loops, i dots, t strokes etc.

Arabic is the official language of twenty two countries representing more than 280 million inhabitants. It is ranked amongst the top ten languages in the world in terms of number of speakers. In our previous studies achieved on high and medium quality documents

Recently, [11-21] attempted to avoid segmentation at all. using morphological operators he tried to recognize at least a part of a word and then the entire word by searching a large data-base of references.

The present research was aimed at the recognition of printed Arabic, widely used in books and periodicals. Such texts are printed almost entirely in so called Naskhi font.

In this paper, a recognition system of Arabic printed text is presented, using a structural method based on segments the word into complementary rather than characters, and implement a system that recognizes machine-printed Arabic words. This method is based mainly on using the pulse coupled neural networks. At recognition time, the complementary and the signatures from PCNN are generated for a word image. The system then matches the detected features with lookup dictionary for the available words. The advantage of using this whole word approach versus a segmentation approach is that the result of recognition is font size independent, which is one important issue in recognition of printed words. Results of preliminary experiments using a lexicon of 10,000 words show a recognition rate of 96%. The next section in this paper gives some important features for Arabic script, followed by a description of the PCNN structure. Section 4 dedicated for the proposed system, and finally the conclusion.

2. ARABIC SCRIPT

Arabic's cursive script in which most characters are connected and their shape vary with position in the word. The optional use of word elongations and ligatures, which are special forms of certain letter sequences. The presence of dots in 15 of the 28 letters Arabic words are built from a closed set of about 10,000 root forms that typically contain 3 characters, although 4-character roots are not uncommon, and some 5-character roots do exist. Arabic stems are derived from these root forms by fitting the root letters into a small set of regular patterns, which sometimes includes addition of "infix" characters between two letters of the root.

Arabic writing can be, in general, classified into typewritten (Naskh), handwritten (Ruq'a) and artistic (Kufi, Diwani, Royal and Thuluth) styles. Arabic is written from right to left and is cursive in general i.e. Arabic letters are normally connected on the writing line to be called midline.

The characteristic of Arabic Script includes:

-The text is written right to left

-Arabic has 28 basic characters, of which 16 have from one to three dots; these dots differentiate between the otherwise similar characters.

-The dot and Hamza are called secondary and they are located above the character, below the character or in the middle of the character. Ex: (ج) - (ب) - (أ)

-Arabic text is cursive, some characters connect to the preceding character or the following character and some others don't connect.

-Consecutive letters within a word are typically joined together by a baseline stroke.

-Characters may assume one of four forms: *beginning*, *middle*, *end*, and *isolated*.

-Six common letters in the alphabet are exceptions to this convention and lack the medial and final forms.

-Arabic text contains a large number of special forms, called ligatures, which replace particular character pairs or even triples. For example, when the LAM character is followed by the ALEF character (لا) they will almost always be combined into a single ligature character called the LAM–ALEF.

-Characters in the word may overlap vertically although non touching ex :أحمر بخير نجيب .

-Arabic characters doesn't have fixed size, size varies across different characters and across different shapes of the same character.

From all those features for Arabic script one can deduce how it is difficult to segment word into characters and find the correct positions for segmentation.

3. THE PULSE COUPLED NEURAL NETWORK

The pulse coupled neural network PCNN can be used in various image processing disciplines. The implementation of the PCNN, [22,23] , with an input image is shown in Fig. (1). The PCNN neuron has a feeding and linking input which are then combined in a second order fashion and then compared to a dynamic threshold .

The equations for a single iteration of the PCNN are:

$$F_{ij}[n] = e^{-\alpha_F} F_{ij}[n-1] + S_{ij} + V_F \Sigma\ M_{ijkl} Y_{kl}[n-1] \qquad (1)$$

$$L_{ij}[n] = e^{-\alpha_L} L_{ij}[n-1] + V_L \Sigma\ W_{ijkl} Y_{kl}[n-1] \qquad (2)$$

$$U_{ij}[n] = F_{ij}[n](1 + \beta L_{ij}[n]) \qquad (3)$$

$$\Theta_{ij}[n] = e^{-\alpha}\Theta[n-1] + V_\Theta Y_{ij}[n] \qquad (4)$$

$$Y_{ij}[n] = 1 \text{ if } U_{ij}[n] > \Theta_{ij}[n-1] \qquad (5)$$

$$0 \text{ Otherwise}$$

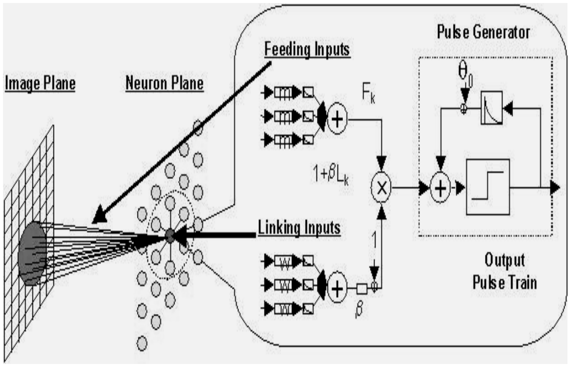


Fig.1 The structural model of PCNN

Where **S** is the stimulus, **F** is the feed, **L** is the link, **U** is the internal activity, **Y** is the pulse output and Θ is the dynamic threshold. The local connections **M** and **W** are fixed (usually Gaussian). This system requires no training. Through these local connections the activation of a neuron adds to the internal activity of the neighboring neurons. Groups of neurons receiving similar stimulus that are spatially close to each other tend to synchronize pulses. The out put of this network is a series of binary images. From those images, a time series signal is created, using the following equation:

$$G(n) = \Sigma_{i,j} Y_{i,j}(n) \qquad (6)$$

This time signal can be used to characterize the input image. If the PCNN iterates *N* times and outputs *N* pulse images *Y*, the signal will be a vector *G* with *N* elements. Each element is the number of white pixels in the corresponding pulse image.

4. THE PROPOSED CWR

The process of recognizing Arabic text can be broadly broken down into five stages: (1) pre- processing, (2) segmentation, (3) feature extraction, (4) classification, and (5) post-processing. The preprocessing stage is a collection of operations that apply successive transformations on an image. It takes in a raw image and enhances it by reducing noise and distortion, and hence simplifies segmentation, feature extraction, and consequently recognition.

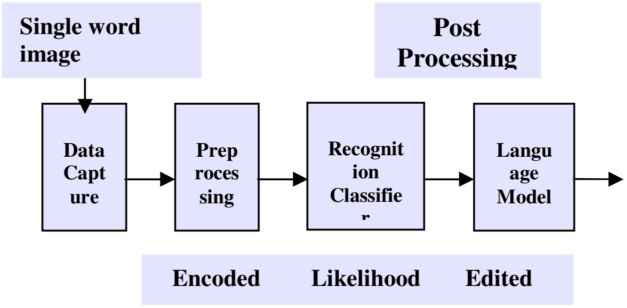
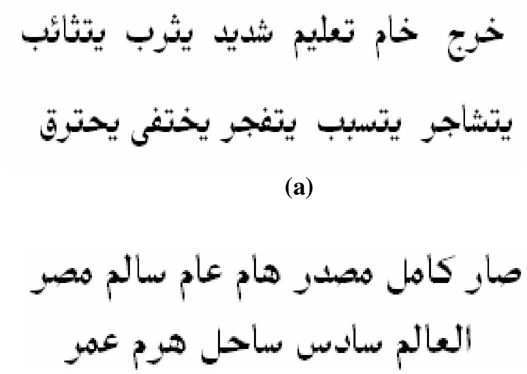


Fig.2 A model for Arabic cursive word recognition system.

In most of the reported AOCR research the segmentation is considered the main source words and text lines. Fig. 2 shows the common process for OCR systems.

The basic idea of our approach is to use the complementary, mainly dots, and isolated characters to speed up the classification process. It can be easily noticed that the Arabic words that contain one or more dot is the dominant over that without dots, see a sample in fig.3. Fig.4 shows the block diagram for the proposed system that starts with capturing the text with scanner, and removing any possible noise in the background. The page was the segmented into lines and words, the complimentary in any word will play a role in the classification step; therefore it will be identified in number and position. The PCNN signature will be generated for the selected word. A decision tree will be used to put the words in a lookup dictionary in groups. By comparing the signature and the complementary of the tested word with those in the nearest group with a matching process the correct word can be found.


خرج خام تعليم شديد يشرب يتثائب
يتشاجر يتسبب يتفجر يختفي يحترق
صار كامل مصدر هام عام سالم مصر
العالم سادس ساحل هرم عمر

(a)

(b)

Fig.3 Two sets of Arabic words (a) Set for words with dots, and (b) Set for words without dots.

4.1 PREPROCESSING

In the proposed system pre-processing includes line and word separation, and word separation, dot extraction. This was applied to on an image after removing noise and apply an adaptive global thresholding algorithm as binarization method. The slope correction applied to the segmented words, the slope is defined as the angle of the baseline of a word.

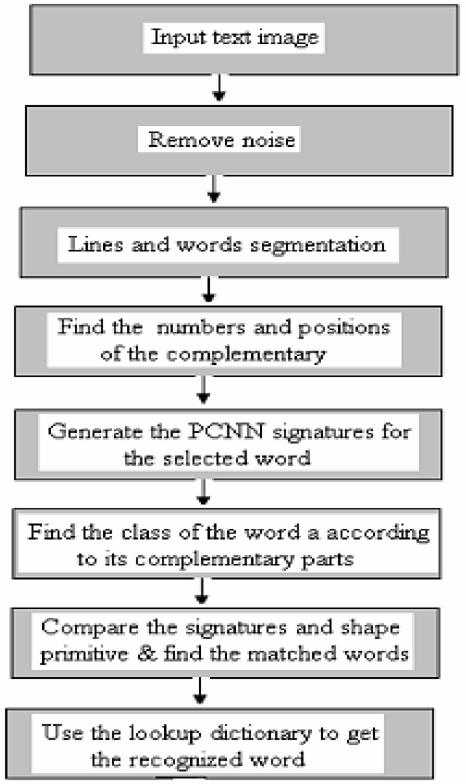

Input text image
Remove noise
Lines and words segmentation
Find the numbers and positions of the complementary
Generate the PCNN signatures for the selected word
Find the class of the word a according to its complementary parts
Compare the signatures and shape primitive & find the matched words
Use the lookup dictionary to get the recognized word

Fig.4 Flowchart for the proposed CWR

After the baseline is estimates the slope may be corrected using the following equation:

$$y = a*x + b \qquad (6)$$

$$M = \tan^{-1}(\text{angle}) \qquad (7)$$

The slope correction algorithm is based on getting (b) for every black pixel given x, y and an angle in the range [-45, 45] then get the most redundant M, B. Now we can get baseline as we have its M, B, using two different points on x-axis to get two points in the y-axis then draw line. If the baseline slope is not horizontal i.e. its degree is not equal = 0, then the image is rotated with a degree equal to the inclination of the baseline.

There will be no need for slant correction since we use printed scripts, where the slant is the deviation of strokes from the vertical axis.

4.2 SEGMENTAION

Segmentation stage is decomposed into three main processes: page segmentation: this involves segmenting the input page into a set of lines, connected component analysis (this involves segmenting the page into set of sub-words), and word segmentation (this involves segmenting each sub-word into set of segments where each segments corresponds to a character of the word).

Page segmentation is a sub-field of document analysis. In works on Arabic that researchers have examined, page segmentation was limited to separating the different lines of a text block. By far most common methods of line separation is to use the horizontal projection histogram.

The line-breaks in the text correspond to gaps in the histogram.

In all printed Arabic characters, the width at a connection point is much less than the width of the beginning character. This property is essential in applying the baseline segmentation technique.

The baseline is a medium line in the Arabic word in which all the connections between the successive characters take place. If a vertical projection of bi-level pixels is performed on the word [equation (8)],

$$v(j) = \Sigma_i\, w(i, j) \qquad (8)$$

Where $w\ (i, j)$ is either zero or one and i, j index the rows and columns, respectively, the connectivity point will have a sum less than the average value (AV) [equation (9)]

$$AV = (1/N_c)\ {}_{j=1}\Sigma^{Nc}\ X_j \qquad (9)$$

where N_c is the number of columns and X_j is the number of black pixels of the jth column. Hence, each part with a sum value much less than AV should be a boundary between different characters. However if the histogram produced from the vertical projection does not follow the condition of equation (10), the character remains un-segmented. By examining Arabic characters, it is found that the distance between successive peaks does not exceed one third the width of the Arabic character. That is

$$|d_k| < d_l/3 \qquad (10)$$

where d_k is the distance between kth peak and peak k+1, and d_l is the total width of the character. Moreover, at the end of a word or a sub-word, equation (11) is also to hold.

$$L_{\mathrm{k}+1} > 1.5L_{\mathrm{k}} \qquad (11)$$

where L_{k} is the kth peak in the histogram. This rule is brought to bear because of the inter-connectivity of Arabic characters and their shapes at the end of a word. This approach depends heavily on a predefined threshold value related to the character width. The main steps for page segmentation algorithm can be described as follows;

1- Based on the horizontal histogram (smoothing Required)
2- Identification of local maxima on the histogram
3- Ignoring those maxima that are far from the mean
4- Identification of every line with three positioning lines at 80% - 100% - 80% levels around the maxima.

Fig.5 shws the horizontal histograms for a set of lines.

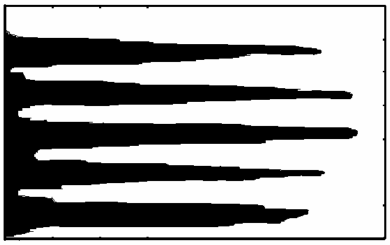


Fig.5 The horizontal histogram for a set of lines of printed text

The same algorithm can be applied in a vertical way to detect the begging and end of each word in a line.

4.3 Complementary parts detection

The complementary of a word includes dots, hamza, and any isolated character within. Those parts can be detected by using the baseline of the word, and then by scanning above and below the baseline to half the maximum width between two successive lines, if a set of black pixels was detected followed by silent space then this is one dot. Also, the isolated characters within one word can be detected by using a vertical projection of a line text on a horizontal axis. The obtained histogram will have some zero value columns. These columns are used to delimit the connected parts; it consists of determining the beginning and the end of the connected parts. Each text line obtained in the horizontal segmentation is sub-divided into connected parts. The vertical sweeping is done from top to bottom. Its principle is as follows:

–Proceeding by a vertical sweeping, we determine the beginning of the connected part corresponding to the first Column of the binary matrix, which contains at least one black pixel.

–Next, we determine the end of the connected part corresponding to the first column, which contains no black pixel., fig.6.

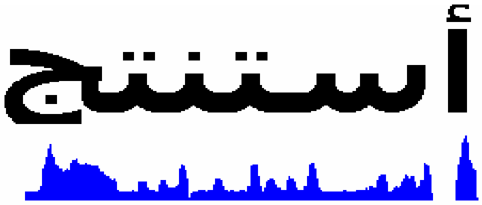


Fig.6 Vertical projection profile of a word

4.4 Building the lookup dictionary

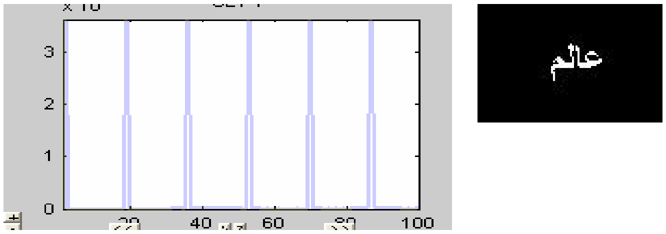
This dictionary contains all the available words that can be used as a reference to classify the words. Each word has it own PCNN signature and the supported shape primitives. The tested word features (PCNN signature and primitives) are then routed in a decision tree to select the proper group of words. Since the used decision tree is relatively hug, which make the number of word in each group is relatively small. We classified the group in this tree as following;

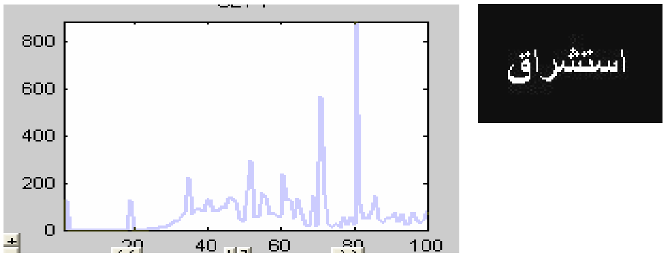
- Words with out complementary
- Words with complementary.
- Words with dots,
- Words with complementary other than dots.
- Words with dots can be grouped according to the number and the position of dots as follows:
- Only one dot (above or below the baseline)
- Two dots (above, below ,one above, and one below)
- Three dots with all the possible distribution above and below the base line
- 4, 5, 6, up to 8 dots with all possibilities.
- Other complementary such as hamza (above below), isolated characters (one, two, or three)

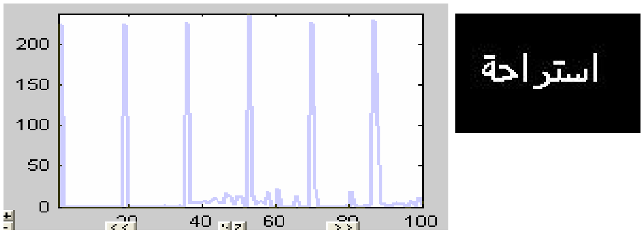
This tree will have about 130 nodes, each of which may be represent a set of words.

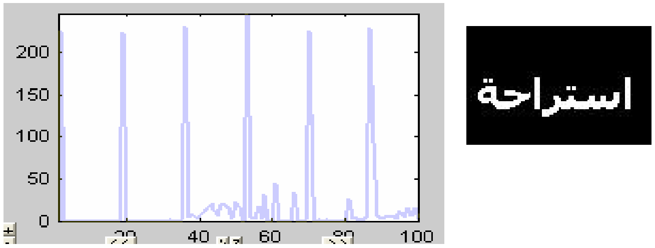
4.5 Tested results

For the testing ten densely printed pages were scanned. After the preprocessing step and segmentation of the page in to lines and words, the complementary parts were specified. The signature of the PCNN was generated for every word. Fig.7 shows a sample of those signatures, it can be seen that those signatures haves a unique distribution for each word, as well as a similar trend for similar words and with different font size or style. Those signature were tested with those in the lookup dictionary and the Euclidian distance was use to find the best matching among the signatures. According the number of word used in the lookup dictionary (10,000 word), and the tested word were 3000 word. The accuracy of classification was 96%.









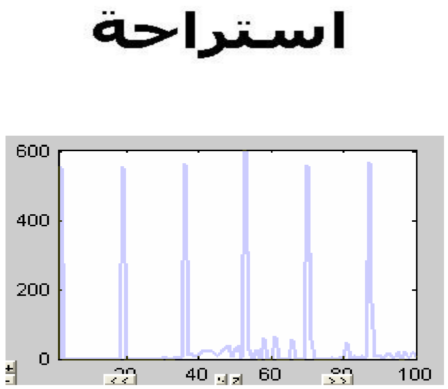


Fig.7 Results of PCNN signatures for a set of words.

5. CONCLUSIONS

In this paper, we have presented a recognition system for printed Arabic writing that involves multi-style and multi-font characters. An algorithm for segmentation of Arabic text is used, based on the use of histograms and of criteria relating to the morphology of Arabic characters. The problem of over-segmentation of some characters was solved by this system. The recognition stage uses a structural method, which does not need skeleton partitioning (which is time consuming, and does not always keep the character's form).

A new approach for offline whole-word recognition have been addressed in this paper. The objective is to eliminate problems associated with poor placement or missing segmentation points in cursive scripts. The signatures generated by the PCNN gave a unique fingerprint for each word, beside using the complementary parts such as dots, make the search in lookup dictionary faster. This method can be tested for larger dictionary (more than 50,000 word), also it can be applied to some extend to handwritten scripts.

6. REFERENCES

[1] Darwish, K. and D. Oard. Term Selection for Searching Printed Arabic. In SIGIR-2002 (2002).pp. 413-423.

[2] Darwish, K., H. Hassan, and O. Emam. Examining the Effect of Improved Context Sensitive Morphology on Arabic Information Retrieval. In ACL Workshop on Computation Approaches to Semitic Languages, Ann Arbor, (2005), pp.233-250.

[3] Baeza-Yates, R. and G. Navarro. A Faster Algorithm for Approximate String Matching. In Combinatorial Pattern Matching (CPM'96), Springer-Verlag LNCS (1996).

[4] Brill, E. and R. Moore. An improved error model for noisy channel spelling correction. In the proceedings of the 38th Annual Meeting on Association for Computational Linguistics, (2000),pp.286 – 293.

[5] De Roeck, A. and W. Al-Fares. A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots. In the 38th Annual Meeting of the ACL, Hong Kong, (2000), pp.330-355.

[6] Harding, S., W. Croft, and C. Weir. Probabilistic Retrieval of OCR-degraded Text Using N-Grams. In European Conference on Digital Libraries (1997), pp. 134-154.

[7] T Steinherz, E Rivlin, N Intrator, Off-line cursive word recognition . A survey, International Journal on Document Analysis and Recognition, Volume 2, Issue 2-3, 1999, pp. 90-110.

[8] Larkey, L., L. Ballesteros, and M. Connell. Improving stemming for Arabic information retrieval: light stemming and cooccurrence analysis. In proceedings of the 25th annual international ACM SIGIR conference, (2002)., pp. 275- 282.

[9] Lee, Y., K. Papineni, S. Roukos, O. Emam, and H. Hassan. Language Model Based Arabic Word Segmentation. In the Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, (2003),pp.399 – 406.

[10] Lu, Z., I. Bazzi, A. Kornai, J. Makhoul, P. Natarajan, and R. Schwartz. A Robust, Language-Independent OCR System. In the 27th AIPR Workshop: Advances in Computer Assisted Recognition, SPIE (1999).

[11] Moussa B., M. Maamouri, H. Jin, A. Bies, X. Ma. Arabic Treebank: Part 1 - 10Kword English Translation. Linguistic Data Consortium (2003).

[12] Oflazer, K. Error-Tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction. Computational Linguistics 22(1), (1996), pp. 73-90.

[13] Taghva, K., J. Borsack, and A. Condit. An Expert System for Automatically Correcting OCR Output. In SPIE - Document Recognition (1994).

[14] Tillenius, M., Efficient generation and ranking of spelling error corrections. NADA (1996), pp.34-56.

[15] Tseng, Y. and D. Oard. Document Image Retrieval Techniques for Chinese. In Symposium on Document Image Understanding Technology, Columbia, MD (2001).

[16] F. Hussain and J. Cowell, “Character Recognition of Arabic and Latin Scripts”, *Proceedings, IEEE International Conference on Information Visualisation*, 2000, pp. 51–56.

[17] D Guillevic , C Y Suen, .Cursive script recognition applied to the processing of bank cheques., Proceedings, International Conference on Document Analysis and Recognition, 1995, pp. 11-14.

[18] B. Albadr and S.A. Mahmoud, "*Survey and bibliography of Arabic optical text recognition*," Signal Processing, vol. 41, no. 1, 1995, pp. 49-77.

[19] L. Hamami-Mitiche, “Segmentation of an Arab Text Paragraph Printed in Characters”, *Proceedings of the 8th International Conference on Computer Theory and Applications, ICCTA’98, Alexandria, Egypt*, 15–17 September 1998, pp. IV.6–IV.8.

[20] A. Amin. Recognition of printed arabic text based on global features and decision tree learning techniques. Pattern Recognition, 33(8):,August 2000, pp. 1309–1323.

[21] A. Amin, “Recognition of Arabic hand printed mathematical formulae”, ***Arabian J. Engrg. Sci.,*** Vol. 16, No. 4, 1991, pp. 531-545.

[22] J. Kinser, *A Simplified Pulse-Coupled Neural Network*, Proceedings, SPIE, Vol. 2760, No. 3, 1996.

[23] J. Atmer, *Image Signatures from PCNN using Computers,* Diploma work, Royal Institute of Technology (KTH), Stockholm, 2003.