http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (7): 971-981, 2009
ISSN 1812-5638

Off-Line Jawi Handwriting Recognition using Hamming Classification

[1]Z. Razak, [1]N.A. Ghani, [1]E.M. Tamil, [1]M.Y.I. Idris, [1]N.M. Noor, [1]R. Salleh,
[1]M. Yaacob, [2]M. Yakub and [2]Z.B.M. Yusoff
[1]Faculty of Computer Science and Information Technology, University of Malaya,
Kuala Lumpur, Malaysia
[2]Department of Al-Quran and Al-Hadith, Academy of Islamic Studies, University of Malaya,
Kuala Lumpur, Malaysia

Abstract: This study is focusing on off-line character recognition. The algorithm of pre-processing such as line and character segmentation is studied and determined so that the design can give a good result and can be implemented in hardware. A process of transformation towards the characters is done using discrete wavelet transform since, it will show the details of the pixels. After that, a process to generate a sequence of binary that using a value of threshold (threshold value is determine by experiment) is done so that it can be use for recognition process. This sequence of binary will be classified using Hamming distance which can trace bit changes in the two sequence of binary and the bit value distinction will be used to recognize the character.

Key words: Off-line Jawi handwriting character recognition, Hamming distance, classification, Jawi

INTRODUCTION

Character recognition is a process that identifies non-digital characters from printed scripts and interprets the collections of the character shapes in a digital format. Jawi script contains 36 characters that can be classified into two categories: grouped form and individual form. Every Jawi character also has three different shape based on their position in a connected Jawi word which are at the beginning, middle or at the end of the word. The character recognition research will be put more emphasis on the writing styles, since there are some of the Jawi characters have similar shapes at the middle and end of the word.

The widely usage of Jawi scripts in inscription, religion text about Islam, petition, newspaper and magazine has encourage this research to be conducted. This research will create a system that efficient, cheap, flexible and user friendly which can promote usage to retrieve, process and store Jawi manuscripts on computers that will facilitate the use of Jawi scripts in office automation and administration by shy Jawi literate IT users. The digitized Jawi scripts can also assist historians in studying old Jawi manuscripts and preserve their content for future research. Without the aid of information technology and full-fledge Jawi digitized system, the content of these old and yet invaluable manuscripts may be lost forever for the current IT literate generation. This study will present our approach by using Hamming distance classifier for off-line handwritten Jawi character recognition.

LINE AND CHARACTER SEGMENTATION

Segmentation process is to simplify or to change the representation of an image so that it can be easily analyze. It is typically used to locate objects and boundaries such as lines and curves in images. Line segmentation is important in analyzing the arrangement and separates the upper and lower lines, while character segmentation is the task of separating the words into its component characters. This two process should be done accurately to prevent any dissemination of rectification to others process such as in feature extraction.

Line segmentation: In present research (Zaidi *et al*., 2007), old manuscripts (with many overlapped characters over the lines) is considered as the domain problem and it require planning and designing of a very accurate method in the first stage of segmentation i.e., line segmentation. There is no artificial intelligence is used in analysis since it can hardly be implemented in hardware or morphological tracing as has been used by Ymin and Aoki (1996), neither using the calculation of pixel averages as done by Cheung *et al*. (1997) and Amin (1991). Instead, this study uses histogram projection without

Corresponding Author: Zaidi Razak, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

taking into consideration of the character orientation and line skew. The false local minimum SP is omitted when normalization of the histogram is performed and this algorithm can improve accuracy and speed, while maintaining the quality of the segmented text lines. The full result of this process is shown in Fig. 1-5.

Louloudis *et al.* (2006) presented a text line detection technique for off-line unconstrained handwriting based on a three step strategy. The first step includes for image enhancing pre-processing, connected component extraction and average character height estimation. In the second step, a block-based Hough transform is applied for potential text lines detection while a third step is applied to correct possible false detections. The performance of the proposed strategy is based on an evaluation technique that compares the text line detection result and the corresponding ground truth annotation.

Fig. 1: Old Jawi manuscript with size 1277×774 pixels

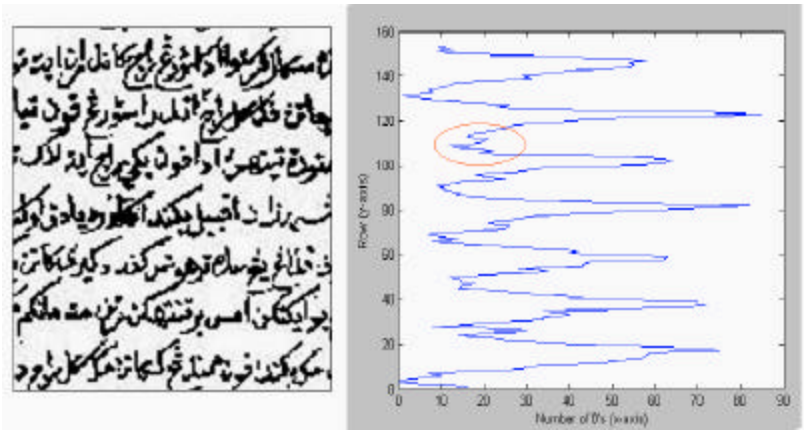

Fig. 2: (a) Binary image of ROI of Old manuscript and (b) Graph row versus no of 0's

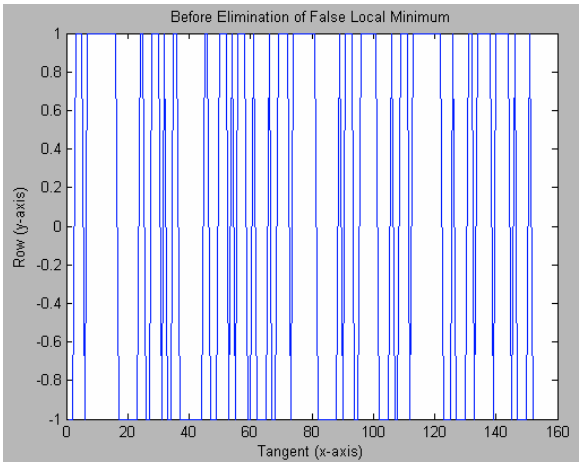

Fig. 3: New representation of tangent versus row (before elimination of false local minimum)

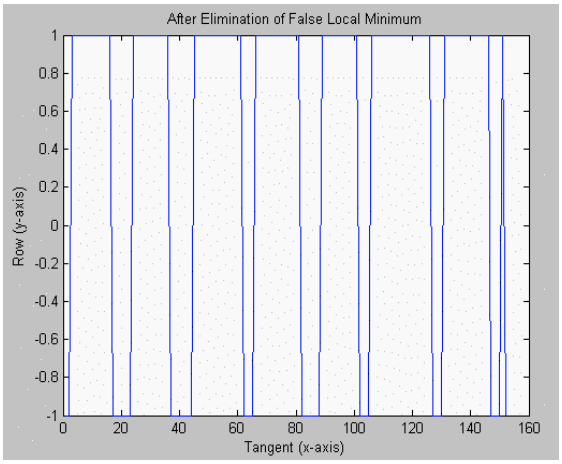

Fig. 4: Tangent versus row (after elimination of false local minimum)

Fig. 5: Result of line segmentation

The method proposed by Likforman-Sulem *et al.* (1995) detected text lines on handwritten documents which may include either lines oriented in various directions, erasures, or annotations between main lines. The method consists of a hypothesis-validation strategy which is iteratively activated until the segmentation is completed. At each stage of the process, the best text-line hypothesis is generated in the Hough domain, considering the fluctuations of the text-line components. Then, the validity of the line is checked in the image domain using proximity criteria which analysis the context which is perceived as the alignment hypothesis. Ambiguous components belonging to several text lines are also marked.

Character segmentation: For character segmentation, it will be much more challenging, since handwritten Jawi text has cursive nature and various writing styles. For this

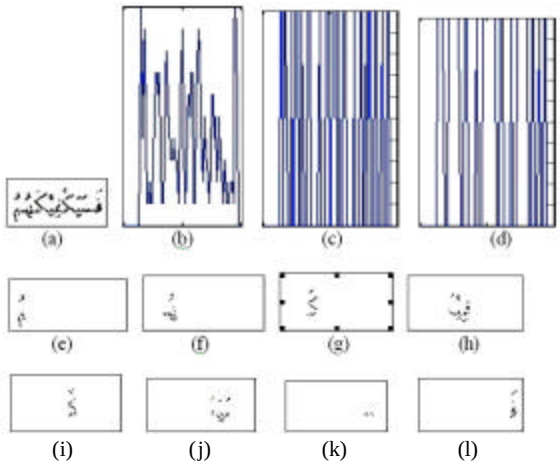

Fig. 6: Character segmentation results

study, histogram normalization and sliding windows is re used for character segmentation process. This method has been used in many several of character recognition system to segment the words and character by horizontal or vertical (Casey and Lecolinent, 1996). The vertical projection histogram will segment the document vertically. It can detect the space between each character and also can specify the location of the vertical strokes in printed documents or any region of various line of handwriting.

Gouda and Rashwan (2004) segmented the word to many basic characters based on the baseline using vertical histogram. In Al-Youseffi and Udpa (1992), the character is segmented by horizontal and vertical projection histogram into primary and secondary parts, while Syiam *et al.* (2006) implemented a clustering technique (k-means algorithm) on the vertical histogram. This improved the performance of histogram technique with recognition of handwritten characters. The character is clustered to identify the similarities of the characters. By using this algorithm, the design will be simple and enables it to be implemented in hardware without requiring a large amount of resources. Below is the sample of character segmentation result. A pre-processing such as edge detection or contour tracing is not performed since 100% accuracy is not crucial in our approach.

The text line height is assumed as the sliding window width. In the histogram, the pixel counts where the negative sign gradients meet the positive sign gradients zare set as the character segmentation points (Fig. 6). If the length between neighboring segmentation points is less than the sliding window width then the particular segmentation points are used for segmentation. Otherwise, if the length between neighboring

segmentation points is more than the sliding window width then the sliding window length is used for segmentation.

FEATURES EXTRACTION

The process of transformation using Discrete Wavelet Transform (DWT) is a process that using the pyramid algorithm which develops by Mallat (1989) for decomposition process of various efficiency resolutions. This decomposition process will be operating on signal (in this research it will be value of pixel) and it can also be applied to image processing.

A system for recognition of handwritten Farsi/Arabic characters and numerals was developed by Mowlaei *et al.* (2002). The discrete wavelet transform is utilized to produce wavelet coefficients, which are used for classification. Haar wavelet is used for feature extraction in Farsi/Arabic handwritten postal addresses containing the names and postal codes of cities from a database of 579 postal addresses in Iran.

Discrete Wavelet Transform (DWT) is chosen because it can provide the information that can describe the position of pixel and also the density of pixel in character representation which has been segmented (Zaidi *et al*., 2005). It also can synthesize pixels to DWT coefficient in short time. By using the Mallat algorithm (Mallat, 1999), DWT can decomposed sub band which signal (value of pixel) is divided into component with high frequency but low resolution. The locality of this frequency and structure of various resolutions is one of the DWT main traits which make it suitable for image compression and to get the best representation to represent the sowing of character pixel that will be process.

Unique code extraction process: The unique code is obtained by scanning rows and columns of DWT coefficient which represent a Jawi character. This scanning is done by using an Eq. 1 as shown below:

$$R_{b,l} = \begin{cases} 1 & k>t \\ 0 & k<t \end{cases} \quad (1)$$

where, R is the unique code, b is row, *l* is column, k is DWT coefficient and t is the value of threshold.

If the threshold value is exceeded, then the value is set to one. Otherwise, it will be set to zero. Twenty two bit strings of values one or zero for both rows and columns will be obtained. Then, both of the 22 bit string will be combined to get 44 bit string of value which is now known as unique code of the character. From here, the Hamming distance will be calculated to get the reference value for the letter Alif.

There are two important matter that need to be consider in determine the value of threshold which is duplicate and class. Duplicate in this term meaning that the unique code which has been produced by using the threshold is similar to other character's unique code while class means every character will be divided into their classes according to its shape. For an example, character ba (ب), ta (ت) and tha (ث) are in the same class since they have a similar shape and can only be differentiate by the position of the dots. So, if the value of threshold causing a duplication of unique code and they are not from the same class then the value of threshold will be neglected. This process will continue until one value of threshold did make any duplication and if there are any duplication the character should be in the same class (Fig. 7).

The list of unique code for each character after the threshold is shown in Table 1-4.

Table 1: List of unique code for isolated character

Isolated	
Characters	Unique code
Alif	01010101110101010000001010101010101010101010
Ba	01110101111111110000001010101010111010101010
Ta	01010111101111100000010101010101010101010101
Tha	11111111111111110000001010101010101010111010
Jim	11111111111111010000001010101010101010101010
Ha	01111111111111110000001010101010101010111010
Kha	01111111111111110000001010101010101010101010
Dal	01111111101110100000010101010101010101010010
Dzal	01010111011111010000001010101010101010101010
Ra	11111101111101010000001010101010101010101010
Zai	01011101111111010000001010101010101010101010
Sin	11110111101111100000001010101010101010101010
Shin	01111101111111110000001010101010101010101010
Sod	11111101111111110000001010101010111010101010
Dhod	11011111111111010000001010101010101011101010
Tho	01111111011111010000001010101010101011101010
Dzo	01110111011111010000001010101010101011101110
Ain	01011111011111010000001010101010101010101110
Ghain	11110101111101110000001010101010101010101010
Fa	01111111111111110000001010101010101011101010
Kaf	01111111010111010000001010101110101010101010
Qaf	01010101110111010000001010101110101010101010
Lam	01010101010111010000001010101010101010101110
Mim	01010111110101010000001010101010101010101010
Nun	01010101110111110000001010101010101010101110
Wau	01010111111101010000001010101010101010101010
He	01011111111111010000001010101010101010101010
Ya	01111111111111010000001010101010101010111010
Cha	11111111111111110000001010101010101010101010
Ga	11011101111111010000001010101010101010101010
Nga	01111111110101010000001010101010101010101010
Nya	01010111111111110000001010101010101010101110
Pa	01011111011111010000001010101010101010101010
Va	01010111111111010000001010101010101010101010

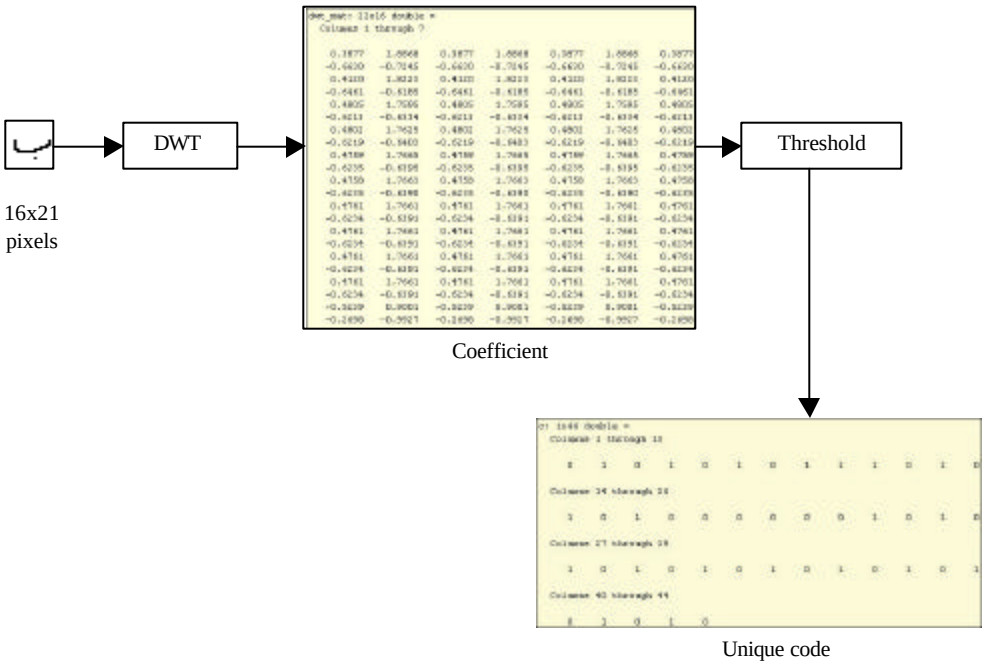

Fig. 7: The summary of feature extraction and threshold process

Table 2: List of unique code for character at the beginning of word

Beginning

Character	Unique code
Alif	0101010111010101000000101010101010101010101010
Ba	0111111111111101000000101010101010101010101010
Ta	1111111111110101000000101010101010101010101010
Tha	0101010111110101000000101010101010101010101010
Jim	1101111111111111000000101010101010101010101010
Ha	1101111111010101000000101010101010101010101010
Kha	0111011111110101000000101010101110101010101010
Dal	None
Dzal	None
Ra	None
Zai	None
Sin	0111111111111101000000101010101010111010101010
Shin	0111111111110111000000101010101010101010101010
Sod	0111111111111111000000101010101010111010101010
Dhod	0101111101111111000000101010101011101010101010
Tho	0111011111011111000000101010101010101010101010
Dzo	0111010111011111000000101010101010101010101010
Ain	0111011111110101000000101010101010101010101010
Ghain	1111111111111101000000101010111010101010101010
Fa	0101010111111101000000101010101010101010101010
Kaf	0101111111111111000000101010101010101010101010
Qaf	1111111111111101000000101010101010111010101010
Lam	0111111101110101000000101010101010101010101010
Mim	1111011101011111000000101010101010101010101010
Nun	1101011111111101000000101010101010101010101010
Wau	None
He	1101111111111101000000101010101010101010101010
Ya	1101011111111111000000101010101010101010101010
Cha	1111111111111101000000101010101010101010101010
Ga	0101111111111101000000101010101010101010101010
Nga	0111111111111101000000101010101010101010101010
Nya	None
Pa	0101111010101010000001010101110101010101010
Va	None

Table 3: List of unique code for character at the middle of word

Middle

Character	Unique code
Alif	None
Ba	1111111111010101000000101010101010101010101010
Ta	0101011101010111000000101010101010111010101010
Tha	0111011101010111000000101010101010111010101010
Jim	1111111111111101000000101010101010101010101010
Ha	1111111111011111000000101010101011111010101010
Kha	1111111111111111000000101010101011101010101010
Dal	None
Dzal	None
Ra	None
Zai	None
Sin	0111111111010101000000101010101010101101010
Shin	0111111111010101000000101010101010101101010
Sod	1101010111111111000000101010101011101010101010
Dhod	1111011111110111000000101010101010101010101010
Tho	0111111101111111000000101010101010101010101010
Dzo	0101110111010101000000101010101010101010101010
Ain	0101111101111111000000101010101010111010101010
Ghain	1111111111010101000000101010101111101010101010
Fa	1111111111111101000000101010101010101010101010
Kaf	1101010111111111000000101010101010101010101010
Qaf	0101111111010111000000101010101010101010101010
Lam	0101010111111111000000101010101010101010101110
Mim	1111110111111111000000101010101110101010101010
Nun	1111111101010111000000101010101010111010101010
Wau	None
He	0101011111010101000000101010101010101010101010
Ya	1111111111111111000000101010101010101011111010
Cha	1111011111111101000000101010101010101010101010
Ga	1111111111011111000000101010101010101010111010
Nga	1111111101110101000000101010101010101010101010
Nya	1111110111111111000000101010101010101010101010
Pa	0101011111111111000000101010101010111010101010
Va	None

Table 4: List of unique code for character at the end of word

Character	Unique code
Alif	01010101111111110000001010101010101010101010
Ba	11111111011111110000001010101011101010101010
Ta	11111111110111110000001010101010111101010101
Tha	11011110111111110000001010101010101010101010
Jim	11011111101110100000001010101010101010101010
Ha	11111111111111110000001010101010101010101010
Kha	11111111111111010000001011101010101010101010
Dal	11111111111111010000001010101110101011101010
Dzal	01010111111011100000001010101010101010101010
Ra	01011110101111100000001010101010101010101010
Zai	01010111111111110000001010101010101010101010
Sin	11010111111011100000001010101010101010101010
Shin	01010101111111110000001010101010111010101010
Sod	11011101010101100000001010101010101010101010
Dhod	11110111111101110000001010101010101010101010
Tho	01111111011101010000001010101010101010101010
Dzo	01111101011111010000001010101010101010101010
Ain	01111111110101110000001010101010101010101010
Ghain	01111111111101010000001010101010101010101010
Fa	11111111111101110000001010101010101010111010
Kaf	01011101010111010000001010101010101010101010
Qaf	01110101111111110000001010101010101010101010
Lam	01010101010111010000001010101010101010101010
Mim	11011101111111110000001010101010101010101010
Nun	01110111011111010000001010101010101010101010
Wau	01010111111101010000001010101010101010101110
He	01010111111101010000001010101010101010101010
Ya	None
Cha	11111111101110100000001010101010101010101010
Ga	01011111011101010000001010101010101010101110
Nga	11010111010101110000001010101010101010101010
Nya	01110111111111110000001010101010101010101010
Pa	11111101110101110000001010101010101010101010
Va	01011111011101010000001010101010101010101010

CLASSIFICATION

The Hamming distance gives a measure of how many bits are different between two bit patterns. Using the Hamming distance of two bit patterns, a decision can be made as to whether the two patterns were generated from different Jawi characters or from the same one. For binary strings a and b the Hamming distance is equivalent to the number of ones in a xor b.

In comparing the bit patterns X and Y, the Hamming distance, HD, is defined as the sum of disagreeing bits (sum of the exclusive-OR between X and Y) over N, the total number of bits in the bit pattern.

$$HD = \frac{1}{N}\sum_{j=1}^{N} X_j \ (XOR)\ Y_j \qquad (2)$$

The most favored distance measure for binary features is the Hamming distance. To further improve the performance, there are two approaches. First, weights can be applied to features (Yoon *et al*., 2005) and optimized using techniques such as genetic algorithms (Guoxing *et al*., 1998; Pouliquen *et al*., 1997). Another approach is to use a similarity measure that gives full credit to features present in both patterns, less credit to those not present in either pattern and no credit to those present in only one of the patterns to be matched (Guoxing *et al*., 1998). Both approaches have been reported to perform better than the simple Hamming distance approach. Yoon *et al*. (2005) produced a new measure that combines these two approaches where experimental results demonstrate its superiority over the other measures.

A compact smart current mode Hamming neural network for classifying complex patterns such as totally unconstrained handwritten digits. It is based on multi-threshold template matching, multi-stage matching and k-WTA (k-Winner-Takes-All), some different from general Hamming neural network. The neural classifier consists of two kinds of templates: one is binary template and another is multi-value programmable templates, each of them has its own threshold and realized in MOS current mirrors, the current mode k-WTA which is reconfigurable is put forward. The second stage matching templates are programmable from outside of the chip. This mixed analog-digital Hamming neural classifier can be fabricated in a standard digital CMOS technology.

Pouliquen *et al*. (1997) used the basic building blocks to design an associative processor for bit-pattern classification; a high-density memory based neuromorphic processor. Operating in parallel, the single chip system determines the closest match, based on the Hamming distance, between an input bit pattern and multiple stored bit templates; ties are broken arbitrarily.

Hamming distance algorithm applied for matching Jawi characters:

- For each pair of identity matrices A and B
- For each matrix position

If the matrix value in identity matrix A does not match the matrix value in identity matrix B increment the distance between identity matrices A and B by 1.

SYSTEM ARCHITECTURE

From the Fig. 8, we can see that there is one main controller that controls all the activities. The activity or process begins with data entry which the image will be scanned line by line and stored into a temporary memory. Next, line segmentation process will begin as shown above. The entity which is line segmenter will calculated the histogram projection whether it fulfill the conditions of line segmentation or not. After line segmentation process is succeeded, it will activate the word

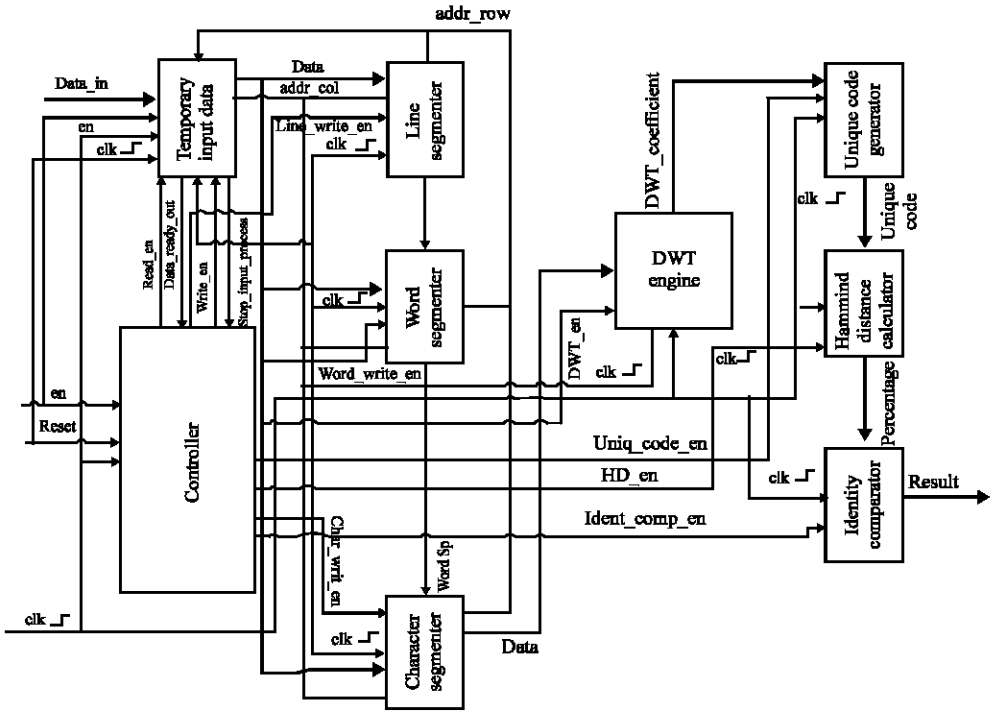

Fig. 8: Combination all entities for jawi character recognition tool

segmentation process. This entity is responsible for calculating the dots to segment the word. The first word which has been segmented will activated the character segmentation process. The character segmenter as shown in Fig. 8 will make a projection histogram for a character and this process will operate together with line and word segmentation process.

After the character has been segmented, the DWT engine will transform the character into wavelet coefficients and then the unique code generator will using the coefficients to produce the unique code by changing or normalized the coefficient so that comparison can be done correctly. Lastly, the unique code will be matched using Hamming distance algorithm and it will generate the Unicode for that character.

EXPERIMENTAL RESULT

Table 5 is the result of Hamming distance for all the isolated Jawi character. From Table 5, the lowest percentage is for character mim, (م) which is 2.2727% and the value of HD is one.

Table 6 is the result of Hamming Distance and its difference percentage for character at begin of word. The lowest percentage is for character tha (ﺙ) with Hamming distance equal to one.

Table 5: Difference percentage of isolated Jawi character against character alif

		Isolated		
Image	Unicode	Char	HD	Percentage
ا	627	alif	0	0.0000
ب ـب	628	ba	5	11.3636
ت ـت	062A	ta	3	6.8182
ث ـث	062B	tha	8	18.1818
ج ـج	062C	jim	6	13.6364
ح ـح	062D	ha	7	15.9091
خ ـخ	062E	kha	6	13.6364
د ـد	062F	dal	4	9.0909
ذ ـذ	630	dzal	4	9.0909
ر ـر	631	ra	4	9.0909
ز ـز	632	zai	3	6.8182
س ـس	633	sin	5	11.3636
ش ـش	634	shin	5	11.3636
ص ـص	635	sod	7	15.9091
ض ـض	636	dhod	5	11.3636
ط ـط	637	tho	7	15.9091
ظ ـظ	629	dzo	7	15.9091
ع ـع	630	ain	7	15.9091
غ ـغ	062A	ghain	4	9.0909
ف ـف	062B	fa	7	15.9091
ق ـق	062C	kaf	5	11.3636
ك ـك	062D	qaf	2	4.5455
ل ـل	062E	lam	3	6.8182
م ـم	062F	mim	1	2.2727
ن ـن	638	nun	3	6.8182
و ـو	639	wau	2	4.5455
ه ـه	640	he	4	9.0909
ي ـي	629	ya	6	13.6364
چ ـچ	630	cha	7	15.9091
ڬ ـڬ	062A	ga	4	9.0909
ڠ ـڠ	062B	nga	3	6.8182
ڽ ـڽ	062C	nya	5	11.3636
ڤ ـڤ	062D	pa	5	11.3636
ۏ ـۏ	062E	va	3	6.8182

Table 6: Difference percentage of Jawi character at begin of word

		Begin		
Image	Unicode	Char	HD	Percentage
ا	627	alif	0	0.00000
ب	628	ba	6	13.6364
ت	062A	ta	5	11.3636
ٹ	062B	tha	1	2.2727
ج	062C	jim	6	13.6364
ح	062D	ha	3	6.8182
خ	062E	kha	4	9.0909
د	062F	dal	None	None
ذ	630	dzal	None	None
ر	631	ra	None	None
ز	632	zai	None	None
س	633	sin	6	13.6364
ش	634	shin	5	11.3636
ص	635	sod	7	15.9091
ض	636	dhod	7	15.9091
ط	637	tho	4	9.0909
ظ	629	dzo	3	6.8182
ع	630	ain	3	6.8182
غ	062A	ghain	7	15.9091
ف	062B	fa	2	4.5455
ک	062C	kaf	5	11.3636
ق	062D	qaf	7	15.9091
ل	062E	lam	5	11.3636
م	062F	mim	6	13.6364
ن	638	nun	4	9.0909
و	639	wau	None	11.3636
ه	640	he	5	11.3636
ي	629	ya	7	15.9091
چ	630	cha	6	13.6364
ڬ	062A	ga	5	11.3636
ڠ	062B	nga	5	11.3636
None	062C	nya	None	None
ڤ	062D	pa	4	9.0909
None	062E	va	None	None

Table 7: Difference percentage of Jawi character at middle of word

		Middle		
Image	Unicode	Char	HD	Percentage
None	627	alif	none	0
ب	628	ba	4	9.0909
ت	062A	ta	4	9.0909
ٹ	062B	tha	5	11.3636
ج	062C	jim	6	13.6364
ح	062D	ha	8	18.1818
خ	062E	kha	8	18.1818
None	062F	dal	None	None
None	630	dzal	None	None
None	631	ra	None	None
None	632	zai	None	None
س	633	sin	4	9.0909
ش	634	shin	4	9.0909
ص	635	sod	5	11.3636
ض	636	dhod	5	11.3636
ط	637	tho	7	15.9091
ظ	629	dzo	1	2.2727
ع	630	ain	7	15.9091
غ	062A	ghain	5	11.3636
None	062B	fa	5	11.3636
ک	062C	kaf	4	9.0909
None	062D	qaf	3	6.8182
ل	062E	lam	4	9.0909
م	062F	mim	7	15.9091
ن	638	nun	7	15.9091

Table 7: Continued

		Middle		
Image	Unicode	Char	HD	Percentage
None	639	wau	None	None
ه	640	he	1	2.2727
ي	629	ya	9	20.4545
چ	630	cha	5	11.3636
ڬ	062A	ga	7	15.9091
ڠ	062B	nga	7	15.9091
ڽ	062C	nya	6	13.6364
ڤ	062D	pa	5	11.3636
None	062E	va	None	None

Table 8: Difference percentage of Jawi character at end of word

		End		
Image	Unicode	Char	HD	Percentage
ا	627	alif	3	6.8182
ب	628	ba	9	20.4545
ت	062A	ta	8	18.1818
ٹ	062B	tha	7	15.9091
ج	062C	jim	4	9.0909
ح	062D	ha	7	15.9091
خ	062E	kha	6	13.6364
د	062F	dal	8	18.1818
ذ	630	dzal	3	6.8182
ر	631	ra	5	11.3636
ز	632	zai	4	9.0909
س	633	sin	4	9.0909
ش	634	shin	4	9.0909
ص	635	sod	4	9.0909
ض	636	dhod	5	11.3636
ط	637	tho	5	11.3636
ظ	629	dzo	5	11.3636
ع	630	ain	4	9.0909
غ	062A	ghain	4	9.0909
ف	062B	fa	7	15.9091
ق	062C	kaf	3	6.8182
ک	062D	qaf	4	9.0909
ل	062E	lam	2	4.5455
م	062F	mim	5	11.3636
ن	638	nun	5	11.3636
و	639	wau	3	6.8182
ه	640	he	4	9.0909
ي	629	ya	None	None
چ	630	cha	5	11.3636
ڬ	062A	ga	5	11.3636
ڠ	062B	nga	4	9.0909
ڽ	062C	nya	5	11.3636
ڤ	062D	pa	4	9.0909
ڤ	062E	va	4	9.0909

Table 7 shows the result for character at middle of word. The lowest percentage for Table 8 is dzo (ظ) and he (ه) with both has the same percentage which is 2.27%.

Lastly, Table 8 shows the result for Jawi character at end of words. Three characters which is alif (ا), dzal (ذ) and kaf (ق) has the lowest percentage which is 6.82% with three Hamming Distanc.

Table 9 shows the example result of classification using Hamming Distance after the character segmentation process as shown in Fig. 9a-f.

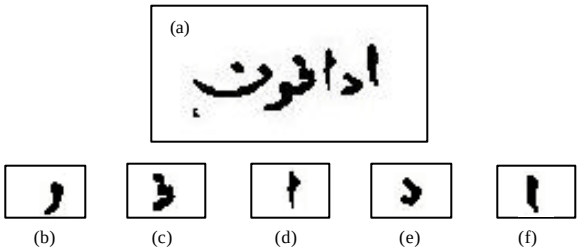

Fig. 9: (a) Image before segmentation process and (b-f) image after character segmentation process

Table 9: Result of hamming distance and its percentage of error for image in Fig. 9

Character	Image	HD	Error (%)
Alif		1	2.2727
Dal		4	9.0909
Alif		0	0
Fa		1	2.2727
Wau		2	4.5455

Table 10: Benchmarking result for Jawi character recognition

Authors	Method	Experiment data	Accuracy	Errors
Al-Yousefi and Udpa (1992)	Off-line statistical method for feature extraction and Bayesian Classifier for recognition system	Isolated, handwritten and printed Arabic text	Classification rate with 85.5% achieved using linear discriminant analysis. By using quadratic discriminant, the classification rate is 99.5%	Classification rate for handwritten character is not as good as printed characters with mixed of fonts and sizes
Abuhaiba *et al.* (1994)	Statistics from moments of horizontal and vertical projections for feature extraction. Clustering for classification process	Arabic handwriting	Classification rate 73.6-100%. Proved to be flexible	The system is believed accurate enough. The only limitation is speed
Mohamed Fakir *et al.* (2000)	Feature extraction using Hough Transform technique.Using dynamic programming matching technique for classification process	Applied to a set of 300 words handwritten Arabic text	The classification rate is 95%	Substitution error are the most common error occurred in this system and it happen during the thinning process
Sarfraz *et al.* (2003)	Using moment invariant technique for features extraction and Artificial Neural Network for classification	Printed Arabic text which using Naskh font	Recognition rate is 73%	The extracted features are deviated from the respective result in training data, because of the resolution differences. It affecting the geometric moments of image
Ahmed *et al.* (2001)	Strokes, loops and feature points. Template matching technique for classification process	Printed Arabic documents	Classification rate are 95.2% for the first set while the second set gave 94.1%	The main error is the character that touching each other in irregular positions due to bad printing and scanning
El-Hajj *et al.* (2005)	Analytical approach by extracting pixels densities, density transitions and concavity configurations along frame with respect to baseline and using character HMM for recognition process	Offline Arabic handwritten	The accuracy is quite satisfactory. The recognition rate range from 85.45 to 87.2%	The diacritical marks are often not in the exact position and generation of a letter can be extended nnder one or more letters of the same words
Present approach	Hough transform for baseline detection, Discrete Wavelet Transform (DWT) for feature extraction and Hamming distance algorithm for classification	Off-line Jawi handwriting	97%	Different size of character's width gives different unique code

BENCHMARKING RESULT

Benchmarking result for Jawi character recognition are compared with earlier data as shown in Table 10.

CONCLUSION AND FUTURE WORK

A system using histogram projection for line segmentation, histogram normalization for character segmentation, Discrete Wavelet Transform (DWT) for feature extraction and Hamming distance algorithm for classification of Jawi characters has been presented in this study. Different width of character could effecting the result and process of fitting the character to same width need to be done to overcome this problem.

REFERENCES

Abuhaiba, L.S.I., S.A. Mahmood and R.J. Green, 1994. Recognition of handwritten cursive arabic characters. IEEE Trans. Patt. Anal. Mach. Intell., 16: 664-672.

Ahmed, M. and A. Elgammal Mohamed Ismail, 2001. A graph-based segmentation and feature extraction framework for arabic text. Proceedings of the 6th International Conference on Document Analysis and Recognition, September 10-13, IEEE Computer Society, Washington, DC., USA., pp: 622-626.

Al-Youseffi, H. and S.S. Udpa, 1992. Recognition of arabic characters. IEEE Trans. Patt. Anal. Mach. Intell., 14: 853-857.

Amin, A., 1991. Recognition of handprinted mathematical formulae. The Arabian J. Sci. Eng., 16: 532-542.

Casey, R.G. and E. Lecolinent, 1996. A survey of methods and strategies in character segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 18: 690-706.

Cheung, A., R.A. Ammar and M. Abdalla, 1997. A recognition-based arabic optical character recognition system. Proceeding of the 2nd IEEE Symposium o n Computer and Communication, Jul. 1-3, IEEE Computer Society, USA., pp: 286-291.

El-Hajj, R., L. Likforman-Sulem and C. Mokbel, 2005. Arabic handwriting recognition using baseline dependant features and hidden markov modeling. Proceeding of the International Conference Document Analysis and Recognition, August 31-September 1, IEEE Computer Society, Washington, DC., USA., pp: 893-897.

Gouda, A.M. and M.A. Rashwan, 2004. Segmentation of connected arabic character using hidden markov models. The International Conference of Computational Intelligent for Measurement Systemsand Applications, July 14-16, USA., pp: 115-119.

Guoxing, L., S. Bingxue and L. Wei, 1998. A modified current mode hamming neural network for totally unconstrained handwritten numeral recognition. The 1998 International Joint Conference on Neural networks, May 4-9, USA., pp: 1857-1860.

Likforman-Sulem, L., A. Hanimyan and C. Faure, 1995. A hough based algorithm for extracting text lines in handwritten documents. Proceedings of the 3rd International Conference on Document Analysis and Recognition, August 14-15, USA., pp: 774-774.

Louloudis, G., K. Halatsis, B. Gatos and I. Pratikakis, 2006. A block-based hough transform mapping for text line detection in handwritten documents. 10th International Workshop on Frontiers in Handwriting Recognition, October, La Baule, France, pp: 515-520.

Mallat, S., 1999. A Wavelet Tour of Signal Processing. 2nd Edn., Academic Press, London, ISBN-10: 012466606X.

Mallat, S.G., 1989. A theory for multiresolution signal decomposition: The wavelet representation. Trans. Pattern Anal. Mach. Intell., 11: 674-693.

Mohamed Fakir, M., M. Hassani and S. Chuichi, 2000. On the recognition of arabic characters using hough transform technique. Malaysian J. Comput. Sci., 13: 39-47.

Mowlaei, A., K. Faez and A.T. Haghighat, 2002. Feature extraction with wavelet transform for recognition of isolated handwritten farsi/arabic characters and numerals. Proceedings of 14th International Conference on Digital Signal Processing, July, USA., pp: 923-926.

Pouliquen, P.O., A.G. Andreou and K. Strohbehn, 1997. Winner-takes-all associative memory: A hamming distance vector quantizer. J. Analog Integrat. Circ. Signal Process., 13: 211-222.

Sarfraz, M., S. Nazim and A. Al-Khuraidly, 2003. Offline Arabic text recognition system. Proceeding of International Conference on Geometric Modeling and Graphics, Jul. 16-18, IEEE Computer Society, pp: 30-35.

Syiam, M., T.M. Nazmy, A.E. Fahmy, H. Fathi and K. Ali, 2006. Histogram clustering and hybrid classifier for handwritten Arabic characters recognition. Proceedings of the 24th IASTED International Conference on Signal Processing, Pattern Recognition and Applications, February 15-17, Innsbruck, Austria, pp: 44-49.

Ymin, A. and Y. Aoki, 1996. On the segmentation of multifont printed uygur scripts. Proc. Int. Conf. Pattern Recognit., 3: 215-219.

Yoon, S., S. Cha and C.C. Tappert, 2005. On binary similarity measures for handwritten character recognition. 8th International Conference on Document Analysis and Recognition, Aug. 29 -Sept. 1, Seoul, Korea, pp: 4-8.

Zaidi, R., S. Rosli and Y. Mashkuri, 2005. Hardware design of on-line jawi character recognition chip using discrete wavelet transform. 8th International Conference on Document Analysis and Recognition, August 29- September 1, IEEE Computer Society, Seoul, Korea, pp: 91-95.

Zaidi, R., Z. Khanza, S. Rosli, Y. Mashkuri and M. Emran Tamil, 2007. A real-time line segmentation algorithm for an offline overlapped handwritten jawi character recognition chip. Malaysian J. Comput. Sci., 20: 69-80.