ISSN 2229-5518

Off-Line Arabic Handwritten Recognition Using Hybrids of Rotational Invariant Segment Features and Support Vector Machines

AMER AL-NASSIRI
College of Information Technology
Ajman University of Science & Technology, UAE,
Email: a.alnassiri@ajman.ac.ae

Abstract: This contribution presents a novel approach for off-line Arabic handwritten recognition.The technique combines the use of Rotational Invariant Segment (RISF) features, chain code frequency (RFC) features,and Support Vector Machines (SVM). The segmentation system assesses a large arrangement of bent portions or strokes of the Arabic word or sub-word image by utilizing a dynamic element extraction procedure. The geometrical feature (RFC) was extracted from character image using a newly programmed Matlab function named FC(d(fr)). The SVM classifier was trained and tested by 16851 Arabic handwritten characters using two class and multiclass experiments. This approach achieved an average rate of 92.96% of the five categories of the AHD/AMSH Arabic handwritten database and as high as 97.9% with fourth age category of the database.

Keywords: Handwritten Recognition System, Support Vector Machines, Arabic Character Recognition, Feature Extraction, Character Segmentation, Segmentation Techniques, OCR

1. Introduction

Any reputable handwritten character recognition (HCR) system consists of preprocessing, segmentation, feature extraction and selection, in addition to the classifier which is trained by selected features or words.The data generation of the handwritten character recognition can be either On-Line or Off-Line. In the former, data are acquired through a writing process using special pen or stylus while in the latter the data are captured by the scanner.

In Arabic handwriting the vowels are written as diacritics (dots) on the consonants. This is called a ‘composite character’, which is the most complex task in any Arabic handwritten character recognition system. The primary challenge in the off-line handwritten recognition system for the Arabic language is its cursive feature. Therefore, the system must be able to distinguish between variations in writing the same Arabic character when that character is written by different writers of different ages or even by the same writer at different times.This isdue to possible minor modification of the same character, like “د “and “ر”

Arabic handwritten character segmentation increased the complexity of off-line recognition system more than the typed characters due to itssegmentation algorithm bothin the explicit or the pure approach. This process keeps on trying to find out and remove the junctions between characters especially when these junctions are sometimes very short.

During the last three decades a substantial research efforts have been undertaken for Arabic handwritten recognition systems [1] but the outcomes were notpromising contrary to theLatin handwritten recognition systems which achieved a successful recognition rate exceeding 90% [2, 3, 4]. Many researchers

ISSN 2229-5518

reviewed the cursive problems in both Arabic and Latin scripts, and they agreed that the most common issue is the baseline detection [5]. For this reason, some researchers moved to other handwritten character recognition using implicit, holistic, and hybrid segmentation instead of explicit segmentation [2, 6, 7]. The most existing Arabic handwritten segmentation algorithms have three major issues: (a)Imprecise cutting words or characters into parts (b) Missing segmentation point and (c) Over- segmenting of a character more than one time, which leads to an error in character or word recognition. To overcome these issues, vertical histograms segmentation approach and implicit approach, are used by many researchers to detect or identify the junction point between connected characters. Another issue is the threshold in these algorithms. Researchers have proposed different techniques to identify the exact and stable threshold by using various types of distributions like Beta distribution [7, 8, 9]. The other widely used segmentation approaches are contour tracing and skeleton extraction, which have solved the overlapping segmentation problems in handwritten Arabic characters [6, 10, 11].

This paper presents an Arabic handwritten recognition system by using a combination of Arabic word segmentation, in which aRotation Invariant Segment Feature RISF [6], contour tracing feature named RFC, and SVM as a classifier were used. The segmentation phase of this research was initiated by previous studies [6, 12],which used a segment handwritten Arabic characters using RISF.Then the extracted features in addition to chain code repetition (RFC) are used to train and test the SVM classifier, by using AHD/AMSH database [12]. This achieved a high recognition rate of 97.9 %.
Fig. 1is an overview block diagram of the proposed system.It consists of scanning, preprocessing, RISF and RFC feature extraction, segmentation, and recognition phase using SVM.

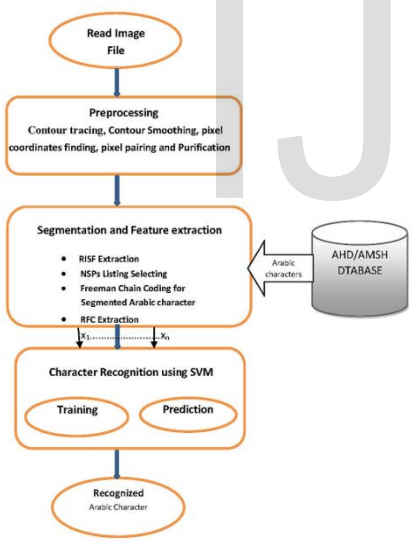


Fig..1.Arabic handwritten recognition System AHCR

Section 2 of this manuscript describes an overview of segmentation and features selection technique and extraction of RFC and RISF while Section 3 provides an outline of SVM. Experimental results and analyses are discussed in section 4,and finally the conclusion is drawn in section 5.

2. Segmentation and Feature ExtractionTechnique

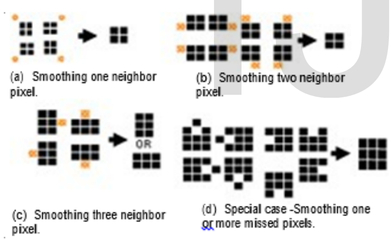
ISSN 2229-5518

The segmentation techniques of Arabic or Latin handwritten words (cursive scripts) may be classified into three categories namely Pure segmentation (Explicit segmentation), Recognition based segmentation (Implicit segmentation), and Segmentation free (Holistic) [2]. Alternatively, they can be categorized into Global transformation features such as Fourier and Gabor transformation, Statistical features such as projection features, or Topological and Geometrical features such as Freeman chain coding [13].

The advantages of using RISF segmentation algorithm are:
1. Solving the overlapping problem.
2. No V/H histogram, baseline or thinning algorithms are required.
3. Its simplicity and fast detection of Final Segmentation point from Nominated Segmentation point list.
4. Its Partial rotation invariant.
5. Solving the problem of segmenting the root valleys of the characters "ن" ,"ي", and “س” by thresholding the angle between the segmentation points.
A summarized RISF segmentation algorithm will be shown in section 2.2, and more details can be found in the segmentation section of reference [6].
We proposed an extraction of additional geometrical feature which represents the repetition of chain code, named RFC, by using newly designed Matlab function called FC (d(fr)) to improve system efficiency. More details will follow in section 2.3.

2.1 Preprocessing

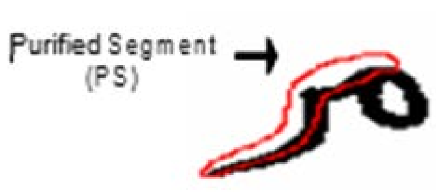
The primary objective of any preprocessing is to prepare a raw data to be used in segmentation phase. The Arabic handwritten script (word, sub-word, or character)is preprocessed into four steps: (1) Contour smoothing, (2) Finding the coordinates (X, Y) of each pixel located on the contour, (3) Adjacent pixel pairing PP, and (4) Purification. Fig.2 (a-d) shows an example on smoothing, Pixel pairing PP, and PPF Purification of Arabic handwritten sub-word “مر”.



		X_1	Y_1	X_2	Y_2
Pixels 1 & 2	PP_1	76	45	75	44
Pixels 3 & 4	PP_2	74	44	73	44
Pixels 5 & 6	PP_3	72	44	71	44
Pixels 7 & 8	PP_4	70	43	69	42
..	-	..	..	..	..
Pixels (n/2)-1 & n/2	PP_n	..	..	..	..

(a) Smoothing (b) Pixel Pairing PP

	X_1	Y_1	X_2	Y_2	PPF
PP_1	76	45	75	44	True
PP_2	74	44	73	44	True
PP_3	72	44	71	44	True
-	..	..	..	..	
-	..	..	..	..	
PP_{n-1}	14	6	14	5	False
PP_n	15	5	16	5	False



(b) PP generation (d) Purified Segment (PS)of Arabic sub-word “مر” .

Fig. 2. (a) Contour smoothing, (b-c) Pixel pairing PP and PPF generation, and (d) Purified Segment (PS)

2.2 Feature Extractionand Segmentation

This approach explores the Arabic handwritten word or sub-word segmentation. The primary objectives of this stage are to prepare a list of Nominated Segmentation Points (NSP) and choosing the Final segmentation points (FSP) from the previous list. To achieve these objectives each Arabic word or sub-word must pass through the following steps

Slope calculation (ΔX/ΔY) using PP and PPF

Preliminary segment creation using PP sign (+ and -)

Creating Decisive Segments (DS) and thresholding

Nomination of segmentation point (NSP)

Final Segmentation points selection (FSP)

Table 1shows the segmentation of the Arabic handwritten sub- word “مر”, and more details can be foundin the related material ofreference [6].

After performing several experiments using the RISF features and then testing the relevant findings, we created the following features and then explored what we thought to be the most efficient and more suitable ones for the proposed recognition of the Arabic Characters using SVM:

The number of Positive Sign (PS) “+” in the preliminary segment.

The number of Negative Sign (NS) “-” in the preliminary segment.

The Decisive Segment Sign Sequence (DSS) generation as shown in**Table 2.**

The Computed Threshold value (V).

Therefore, the new suggested RISF feature vector consists of PPF, PS, NS, DSS, and V.

Table 1. Final segmentation point selection FSP (61, 31) for Arabic sub-word “مر” .

X_1	Y_1	X_2	Y_2	Pixel pair flag	ΔX	ΔY	ΔY/ ΔX	preliminary segment	preliminary segment No	DS	DS No	Threshold Value (V)	NSP	FSP
76	45	75	44	True	-1	-1	1	+		+			Null	Null
74	44	73	44	True	-1	0	0	+		+			Null	Null
72	44	71	44	True	-1	0	0	+		+			Null	Null
70	43	69	42	True	-1	-1	1	+		+			Null	Null
68	42	67	41	True	-1	-1	1	+		+			Null	Null
67	40	66	39	True	-1	-1	1	+		+			Null	Null
66	38	65	37	True	-1	-1	1	+	1	+			Null	Null
65	36	66	35	True	1	-1	-1	-		+			Null	Null
66	34	66	33	True	0	-1	Invalid	-	2	+			Null	Null
66	32	65	31	True	-1	-1	1	+		+			Null	Null
64	31	63	31	True	-1	0	0	+		+			Null	Null
62	31	61	31	True	-1	0	0	+		+			2	2
60	31	59	31	True	-1	0	0	+		+			1,2	Null
58	32	57	32	True	-1	0	0	+		+			1,2	Null
56	32	55	32	True	-1	0	0	+		+			1,2	Null
54	32	53	31	True	-1	-1	1	+		+			1,2	Null
52	31	51	31	True	-1	0	0	+	3	+	1	2	1,2	Null

Table 2. The Decisive Segment Sign Sequence (DSS) generation.

Segment Sign(+ or -)						DSS	DSS value
+	+	-	-	+	....	11001..	25
+	+	-	-	+	....	11000...	24
...	...	...	...	..	...	....	.....

2.3 Freeman Chain Coding (FC) and (RFC)

In this stage, we proposed that all segmented or isolated Arabic characters, from segmentation stage and AHD/AMSH Arabic handwritten database will be subjected to 8-connectivity Freeman chain coding algorithm using MATLAB function [14]

C = fchcode (b, conn, dir)

The abbreviation ‘b’represents an array of image file, ‘conn’represents the connectivity (4 or 8), and ‘dir’ is the directory contained image file to be processed. This function computes the Freeman chain code of a (2xn) set of ordered boundary points stored in array 'b'. 'C' is the output structure with the following fields:

c.fcc = Freeman chain code (1 X n)

c.x0y0 = Coordinates where the code starts, other output parameters of the function where ignored.

In this research we also proposed a new MATLAB function named FC (d(fr)) to compute the repetition of FC where the output structure of this function is a feature set called RFC.

FC (d(fr)) = fcfreqcy (c.fcc)

The output FC has two parameters, the first represents the direction (d) while the second (fr) represents the direction's repetition. Mathematically, the Freeman chain code of Arabic character, word, or sub-word (FC) and the Smoothed Freeman chain FC(d(fr)) are defined by the following equations 1 and 2 respectively.

FC(X) = (x1, x2 , ….., xn), xi ϵ {0, 1, …, 7}, n = Chain length (1)

FC(d(fr)) = x1 (fr1), x2(fr2), …, xm(frm)), m≤ n (2)

Frj is the repetitionof xj,j=1,2,.., m

The following example shows part of the Arabic handwritten word "نعبد "

c = fchcode(b)

Where b represents an array extracted from scanned image of Arabic word "نعبد ", then

c.fcc = 2323323324543544334433433433211111……

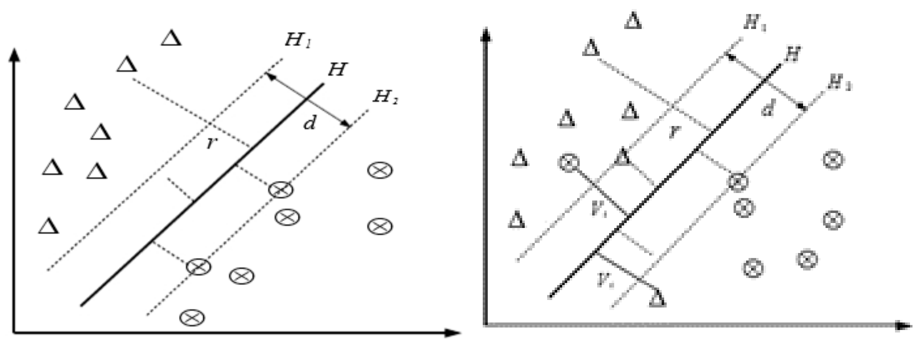
FC(d(fr) = fcfreqcy (2323323324543544334433433433211111) …….

FC(d(fr)) = 2(1) 3(4) 4(8) 3(3) 1(16) 3(10) ……

The next step of the work consists the use of these RFC and RSIF (PPF, PS, NS, DSS, and V) feature sets to train and test the SVM learner and classifier respectively.

3. Support Vector Machines (SVM)

In recent years an extensive researchhad been published about SVM at the application of image processing. Vapnik[15] originally proposed Vector Machines (also called vector support systems in machine learning SVM). They are directed learning models with related learning calculations that investigate information and perceive examples, utilized for grouping and setback investigation. Given an arrangement of preparing illustrations, every one samplehas a place with one of two classificationseither SVM preparing calculation constructs a model that allocates new cases into one classification or the one making it a non-probabilistic parallel direct classifier. An SVM model is a representation of the illustrations as focused in space and mapped so that the samples of the different classes are isolated by an acceptable wide hole. The SVM is a linear classifier where the learning of the support vectors is supervised. The data for the support vector machine is assumed to be generated identically and independently from a fixed (unknown) data distribution. The hyperplane, implicitly defined by support vectors (SV), divides the positive from negative data instances. Burges [16] indicated that dataset nearest to the hyperplane is in the decision margin. Therefore, any addition or removal of an SV changes the hyperplane boundary. After completing the training, then it is possible to reconstruct the hyperplanes and classify new data sets from the SV only. Fig. 3 shows linearly and non- linearly separable hyperplane, where there are two groups of data points represented by ' Δ ' and'⊗ '. The possibility of an infinite number of hyperplanes might exist but only one hyperplane represented by solid line optimally separates the sample points and is situated in between the maximal margins as shown in Figure 3. Additional, related mathematical details could be found in other publication [17].



(a) Linearly Separable (b) Non-linearly Separable

Fig.3.Linearly and Non-Linearly Separable Samples Indicated in a Hyperplane

The process of SVM classification is to divide training data set by a hyperplane separation. SVM is different from others learning machines because of its ability to find the optimal separation hyperplane by minimizing the gap between the hyperplane and the training data points. Because the data is linearly separable then the hyperplane is shown as a line (Fig. 3).If the data is not linearly separable then a kernel function should be used to remap these non-linear data points in a different dimension, which can be separated linearly by a line. The kernel is the essential parameter in SVM as it performs a fundamental task of finding a way to separate data points and thus to classify the unknown data.

The cycle of SVM contains learning and classification phases. In the learning phase, the SVM is provided with tagged training data for building training model that is used in the classification step. In most cases, this model is obtained using the kernel function. The classification of unknown classes depends on the trained model and if the trained model is comprehensive then the classification process will be successful. The method of construction of a training model includes converting the input data into a format comprising a set of characteristics to be read by the SVM. The effectiveness of features measured by the precision of SVM classifier and its ability to classify unknown data into benign or malignantis mostly based on the features' characteristics and uniqueness. The classification in the suggested recognition system of the Arabic characters with SVMis performed by the following steps: (1) Conversion of the feature sets to vectors used for similar features by SVM, (2) Scale SVM featured vector if necessary, (3) Perform the required cross-validation to select the feature vector data sets and find the exact value of RFC and RISF functions, and (4) Selection of the best values of RFC and RISF to train the SVM. During SVM training phase, a set of the stored Arabic handwritten character samples are processed by segmentation stageto extract RFC and RISF (PPF, PS, NS, DSS, and V) features. These vectors are used to train the SVM as shown in Figure 1, which also illustrates the architecture of the SVM proposed for the Arabic handwritten recognition System AHCR.

4. ExperimentalResultsand Analysis

In this section, we provide an experimental confirmation of our analysis on SVM using real data sets. All the experiments are done using MatLab R2015a.

4.1 Data Set

The experiments are conducted by using a large number of Arabic handwritten words, sub-words, and characters which are extracted from the Arabic handwritten database AHD/AMSH [12]. This contains 56170 isolated handwritten Arabic characters written by 82 writersand grouped into five categories according to the age as shown in **Table 3.**

Table3. AHD/AMSH database writers' age groups

categories	writers		Database				
	M	F	word	Sub-word	letter	Courtesy	Digits
5-15 Years	7	7	2100	4956	9590	490	280
16-25 Years	12	10	3300	7788	15070	770	440
26-35 Years	8	9	2550	6018	11645	595	340
36-45 Years	9	8	2550	6018	11645	595	340
Above 45	5	7	1800	4248	8220	420	240
Total			12300	29028	56170	2870	1640

Two different data sets are used one for training and one for testing. About 14323 Arabic handwritten characters are chosen randomly to train the SVM, while 2528 Arabic characters are used for testing SVM classifier as shown in **Table 5**.

4.2 Kernel function

Many kernels are used in SVM models; these are Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid as shown in equation 3. After carefully reviewing the different kernels, we selected the linear and RBF kernels for further evaluation. Then after many trials we discovered that both kernels have the same accuracy, but we chose the RBF for our further study because of their localized and limited response across the entire real x-axis range.

$$\varnothing = \begin{cases} X_i \; X_j & \text{Liner} \\ (\gamma X_i \; X_j + \text{Coefficient})^{\text{degree}} & \text{Polynomial} \\ \exp(-\gamma \, | X_i - X_j |^2) & \text{RBF} \\ \tanh(\gamma \chi_i \; X_j + \text{Coefficient}) & \text{sigmoid} \end{cases} \qquad (3)$$

4.3TwoClass Experiments

We performed two experiments to train the two class SVM using Matlab SVM toolbox in which the kernel parameter γ was set to 1.7. In this experiment we tried to confirm that SVM is not only able to classify dissimilar Arabic handwritten characters like “ح” and “ص” (category 2 in **Table 4**) but also to differentiate similar characters with minor differences like “ر” and “د” or “ق “and “ف “(category 1 in **Table 4**). The main reason for that is because these characters could be segmented but not completely isolated. **Table 4** summarizes recognition and error rates and also shows the similar Arabic handwritten character pairs. This belongsto category 1, a different character sets in which the classification error is higher than dissimilar character pairs due to the cursive nature of the Arabic handwriting characters, the writers, and the segmentation process.
In the first category of two- class experiments the differentiation of the Arabic character class pairs were shown to be similar and were frequently misrecognized by SVM. Generally, in all previous experiments,the SVM shows superior result than other handwritten classification techniques like ANN [18, 19].

Table 4. Two class experiments on AHD/AMSH, the first two character pairs are similar while the third one is dissimilar

Category #	Character pair	No. of training samples	Support vectors	Error rate	Classification rate
1	ف → ق	6650	188	2.46%	97.54%
1	ـدر →	7640	217	2. 08%	97.92%
2	ص → ح	3550	205	0.46%	99.54%

4.4 MulticlassExperiment

For a multiclass analysis, we thinkit is not essential to combine two class sets of SVM into multiclass classifier as many researchers do [20] because the results obtained in a two class experiment proved that the efficiency of SVM in the classification of both different and dissimilar Arabic character belong to various character pairs and vice versa.

For multiclass experiments, we used about 30% of the total Arabic handwritten characters of the AHD/AMSH database, and 85% of these are used for training SVM while the remaining 15% are used for testing the SVM classifier as shown in **Table 5. Fig. 4** shows a snapshot illustrating our SVM classification GUI. **Table 4** summarizes classification recognition rates of SVM classifier carried on AHD/AMSH database. The different five writer age categories show different recognition rates, and the lowest error classification rate was found in the age category (36-45 years), as illustrated in Fig.5. This is a normal finding due tothe maturity and clarity of the writer's handwriting. Most of the unrecognized handwritten Arabic characters are those from segmentedwords or sub-words that are inclined or skewed, from a horizontal line, by an angle more than 38o. Therefore, after repeating the same experiments by excluding these highly skewed words or sub-words, we found that the average classification rate had improved by 1.1% - 1.95%, and reachedup to 94.51% as shown in **Table 5 and Fig. 5.** This proposed system had achieveda high recognition rates with printed Arabic characters of different font types as shown in **Table 6and Table 7.**

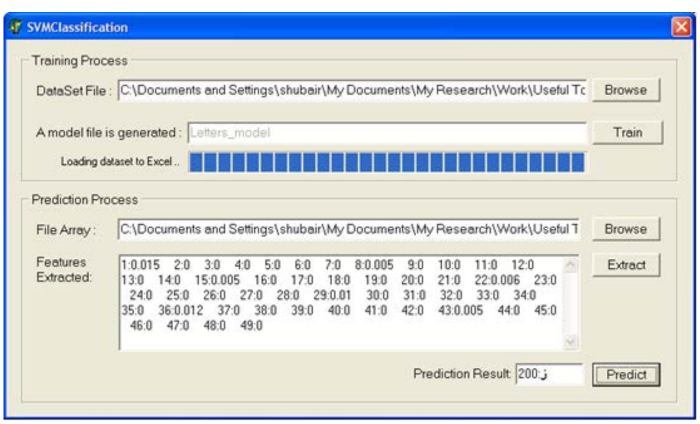


Fig.4. Asnapshot from our SVM classification GUI.

Table 5. The Performance of SVM Classifier Character Recognition Systems Handwritten on AHD/AMSH database

ISSN 2229-5518

categories	writers		Letters (total)	Experiment dataset (30% of total letters)	Training (85%)	Testing (15%)	Recognized letters	Recognition Rate* %	Recognition Rate** %
	M	F							
5-15 Years	7	7	9590	2877	2445	432	367	84.95	86.90
16-25 Years	12	10	15070	4521	3843	678	625	92.18	93.93
26-35 Years	8	9	11645	3493.5	2969	524	500	95.42	96.89
36-45 Years	9	8	11645	3493.5	2969	524	513	97.90	99.00
Above 45	5	7	8220	2466	2096	370	349	94.32	95.84
Total			56170	16851	14323	2528	2344		
Average Recognition rate before removing highly skewed Arabic handwritten words								92.96%	
Average Recognition rate after removing highly skewed Arabic handwritten words									94.51%

* Recognition Rate before removing highly skewed Arabic handwritten words

** Recognition Rate before removing highly skewed Arabic handwritten words

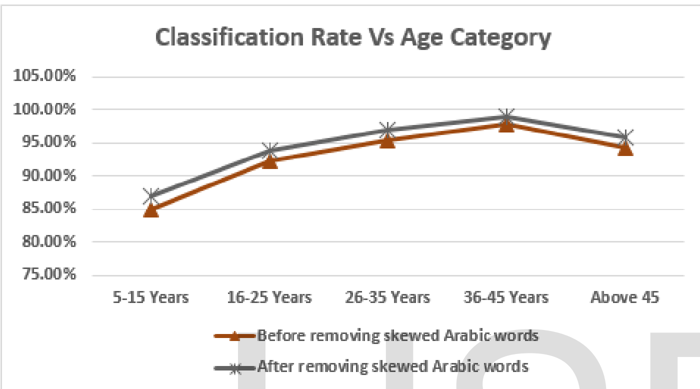


Fig. 5.Classification rate versus five-age category of AHD/AMSH database

Table 6. The Performance of SVM Classifier Character Recognition Systems Handwritten on printed Arabic fonts (size 14).

Font Name	No of Letters' Shapes	Trained (100%)	Tested (85%)	Recognized letters	Correctness Rate %
Arial	107	107	91	87	96
Arabic Transparent	107	107	91	87	96
Times New Roman	107	107	91	88	97
Traditional Arabic	107	107	91	90	99
Total		428	364	352	
Average Recognition Rate					97

Table 7. Arabic Character Set

Character	Shapes							No of Shapes	Character	Shapes							No of Shapes
Alif	ا	ـأ	ـئ	ئـ	ء	أ	ؤ	7	Dhad	ض	ضـ	ـض	ـضـ				4
Baa	ب	بـ	ـب	ـبـ				4	Tta	ط	طـ	ـط	ـطـ				4
Thaa	ت	تـ	ـت	ـتـ	ة	ـة		6	thaa	ظ	ظـ	ـظ	ـظـ				4
thaa	ث	ثـ	ـث	ـثـ				4	Ain	ع	عـ	ـع	ـعـ				4
Jeem	ج	جـ	ـج	ـجـ				4	Ghain	غ	غـ	ـغ	ـغـ				4
Hhaa	ح	حـ	ـح	ـحـ				4	Faa	ف	فـ	ـف	ـفـ				4
Khaa	خ	خـ	ـخ	ـخـ				4	Qaf	ق	قـ	ـق	ـقـ				4
Dal	د	ـد						2	Kaf	ك	كـ	ـك	ـكـ				4
Thal	ذ	ـذ						2	Lam	ل	لـ	ـل	ـلـ				4
Raa	ر	ـر						2	Meem	م	مـ	ـم	ـمـ				4
Zay	ز	ـز						2	Noon	ن	نـ	ـن	ـنـ				4
Seen	س	سـ	ـس	ـسـ				4	Haa	ه	هـ	ـه	ـهـ				4
Sheen	ش	شـ	ـش	ـشـ				4	Waw	و	ـو						2
Sad	ص	صـ	ـص	ـصـ				4	Yaa	ي	يـ	ـي	ـيـ				4
Total (107)																	

5.Conclusion

In this work, we proposed a new Arabic offline handwritten character recognition system based on a hybrid feature extraction technique to select RISF and Chain code RFC features with SVM as a classifier. For training and testing the SVM, we used AHD/AMSH database and Microsoft Office font's database. The obtained results showed that this system is reliable and its use could be extended to any other cursive handwriting The average recognition rate achieved, based on 16851 Arabic handwritten character collected from different age writers, was about 92.96% and it reached 94.51% by removing the skewed word or sub-word before segmentation process. The error rate was noticeably reduced when the system was trained and tested by different font types of Arabic printed characters thus improving the average recognition rate by 2.41% compared to the average recognition rate of handwritten Arabic characters.

In general, by using the newly designed Matlab function for determining the repetition of FC code of Arabic handwritten characters and SVM classifier, it is found that the technique is more robust for relatively eligible handwritten character rather than hard to decipher the cursive word.

References

[1] Zeki A., "The Segmentation Problem in Arabic Character Recognition: The State of Art," Proceedings of the IEEE Information and Communication Technologies, pp. 11- 26, 2006.

[2] Choudhary A., "A Review of Various Character Segmentation Techniques for Cursive Handwritten Words Recognition", International Journal of Information & Computation Technology, Volume 4, Number 6, pp. 559-564, 2014.

[3] Rehman, A., Saba, T., "Off-Line Cursive Script Recognition: current advances, comparisons and remaining problems", ArtifIntell Rev, 37, pp. 261-288., 2012.

[4] Verma, B., Blumenstein, M., "Pattern Recognition Technologies and Applications: Recent Advances", Information Science Reference (An Imprint of IGI Global Publications), Hershey, New York, pp. 1-16, 2008.

[5] Nawaz S., Sarfraz M., Zidouri A., and Al-Khatib W., "An Approach to Offline Arabic Character Recognition Using Neural Networks", ICECS, vol. 3, pp. 1328-1331, 2003.

[6] Shubair A., Al-Nassiri A, and Rosalina A.,"Off-Line Arabic Handwritten Word Segmentation Using Rotational Invariant Segments Features", The International Arab journal of Information Technology, Vol. 5, No. 2, April, 2008.

[7] Al-Saleh A. Ali El-Zaart, AlSalman A., “Dot Detection of Optical Braille Images for Braille Cells Recognition”, Computers Helping People with Special Needs, Lecture Notes in Computer Science Volume 5105, pp 821-826, 2008.
[8] Ashraf Elnagar*, RahimaBentrcia, “A Multi-Agent Approach to Arabic Handwritten Text Segmentation”, Journal of Intelligent Learning Systems and Applications, 2012, 4, 207-215
[9] Muhammad Razzak, Muhammad S. and S. A. Hussain, “Locally baseline detection for online Arabic script based languages character recognition”, International Journal of the Physical Sciences Vol. 5(7), pp. 955-959, July, 2010
[10] AL-Nassiri A., "Recognizing Isolated Handwritten Arabic Characters Using Hybrid of Modified Directional Element Feature and General Autoassociative Memory," in Proceedings of the ACIT'2005, AL-Isra University, Jordan, 2005.
[11] Lorigo L. and Govindaraju V., "Offline Arabic Handwriting Recognition: A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 712-724, May2006.
[12] AL Nassiri A., Shubair A.,” A New Arabic (AHD/AMSH) Handwritten Database, ACIT2007, Tishreen University, Syria, November 26-28, 2007.
[13] AL-Nassiri A., “Neural: Freeman Chain Approach Versus Classical Approach for Arabic Character Recognition System," in Proceedings of the SPECT'200, Florida, USA, pp. 624-629,2001.
[14] Gonzalez R., Woods R., Eddins S., “Digital Image Processing using MATLAB”, 2nd edition, Gatesmark Publishing, 2009.
[15] V. Vapnik,“Statistical Learning Theory ”, John Wiley And Sons, (1998)
[16] Burges C.J.C, “A tutorial on support vector machines for pattern recognition”, Data Mining and Knowledge Discovery, 2(2):161-167, 1998.
[17] NayakJ.,Naik B., and Behera H. S, “A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications & Challenges”, International Journal of Database Theory and Application Vol.8, No.1 (2015), pp.169-186.
[18] Muhammad NaeemAyyaz, Imran Javed and Waqar Mahmood.” Handwritten Character Recognition Using Multiclass SVM Classification with Hybrid Feature Extraction”, Pak. J. Eng.& Appl. Sci. Vol.10, Jan. 2012 (p. 57-67-)
[19] Parveen Kumar, Nitin Sharma, and Arun Rana,” Handwritten Character Recognition using Different Kernel-based SVM Classifier and MLP Neural Network (A COMPARISON)”, International Journal of Computer Applications (0975–8887) Volume 53, No.11, September, 2012.
[20] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt, “On-Line Handwriting Recognition with Support Vector Machines: A Kernel Approach”, IWFHR, PP 49-54, 2002