Journal of Engineering Science and Technology
Vol. 16, No. 3 (2021) 2719 - 2735

ANALYSIS OF FEATURE EXTRACTION AND CLASSIFICATION FOR OFFLINE ARABIC HANDWRITING WORD RECOGNITION

HAITHAM QUTAIBA GHADHBAN*,
MUHAINI OTHMAN, NOOR A. SAMSUDIN
Software Engineering Department, Uinversiti Tun Hussein Onn Malaysia,
Parit Raja, Batu Pahat, 86400, Johor, Malaysia
*Corresponding Author:gi170052@siswa.uthm.edu.my

Abstract

Arabic handwriting recognition is currently experiencing a massive rise in terms of analysis research. It is clear in the lecture review that researchers concentrate on Arabic handwriting recognition. A major increase has been reported in feature extraction and classification techniques compared to before. Offline Arabic handwriting recognition is an exceptionally significant research topic due to the cursive nature of written Arabic words and writing styles which makes the problem of recognizing Arabic words hard and challenging. In this paper, we investigate three feature extractors techniques Chebyshev Moments (CM), Statistical and Contour based Feature (SCF) and Zernike moment (ZER) for offline Arabic handwritten word recognition. We have considered a number of classes with different shapes of an image. These features have been trained separately with three classifiers namely, Extreme Learning Machine (ELM), Support Vector Machine (SVM) and Reduced Kernel Extreme Learning Machine (RKELM). The experiments were evaluated on IFN/ENIT database, experimental results show that shape of feature has a major impact on training classifiers.

Keywords: Extreme learning machine (ELM), Feature extraction, Reduced kernel extreme learning machine (RKELM), Support vector machine (SVM), Word recognition.

1.Introduction

A handwriting recognition system can be either online or offline. The offline handwriting system is based on optical character recognition (OCR) and is usually applied on scanned documents. On the other hand, online handwriting pressure is applied on digital instruments and sequence of points traced out by the pen. Offline handwriting recognition involves the automatic conversion of text in an image into letter codes which are usable within computer and text processing applications, and it is generally observed to be harder than online handwriting recognition. In the online case, features can be extracted from both the pen trajectory and the resulting image, whereas in the offline case only the image is available [1].

Arabic handwriting recognition is a highly important research topic due to different challenges. The cursive nature of writing the Arabic word, the connectivity between the Arabic letters, the huge number of Arabic words over 12 million, the different styles of writing, and the variation of various factors which makes the problem of recognition of Arabic words hard and challenging. Insignificant amount of research has been carried out in the field of Arabic handwritten recognition compared to Latin. Due to the complexity of Arabic text and limited Arabic databases recognition of Arabic text is at an earlier stage compared to the methods for recognition of Latin, Chinese, and Japanese text. Much work remains to be done in the field [2].

The modus operandi for handwriting recognition can be separated into two distinct techniques. While one technique uses segmentation algorithms for the identification of characters, the other is devoid of any segmentation process, and considers the whole word a compilation of characters [3]. The emphasis of this study is on the latter technique.

2. Research Aim

This work present multi feature extractors (CM) and (ZER) were brought into play to gauge the effect of different moments and number of orders on the recognition process. Subsequent to the utilization features of (SCF), Three classifiers trained individually with each one of feature extractors. The level of efficiency, under a variety of circumstances. We have taken under consideration the shape of image, number of windows and variation of classes. Scenarios were evaluated on an IFN/ENIT database. A comparative study, involving several established Arabic handwritten word recognition techniques, was conducted to assess the performance of our comparative analysis.

3.Background

This section starts with a brief review of Arabic handwriting recognition, then illustrates recent studies of feature extractors and classifiers that are related.

3.1. Arabic handwriting recognition

Arabic is spoken by approximately 422 million native and non–native speakers [4]. While spoken Arabic varies across regions, written Arabic has a standardized version for official communication across the Arab world. This written Arabic is sometimes called “Modern Standard Arabic” (MSA). Furthermore, the Arabic script has been adopted for use in a wide variety of languages besides Arabic. These languages include some non–Semitic languages like Persian, Kurdish, Malay, and Urdu. Thus, the ability to automate the interpretation of written Arabic could have widespread benefits.

Arabic handwriting recognition can also enable the automatic reading or searching of historical Arabic manuscripts. Since written Arabic has changed little over time, the same techniques developed for MSA can be applied to many Arabic handwritten manuscripts. Automatic processing of manuscripts can greatly increase the availability of their content because the writing in manuscript is usually neater than free handwriting, the recognition task may be simpler.

3.2. Related studies

Efforts aimed at overcoming the obstacles associated with handwriting recognition, have taken several different routes. While some researchers concentrated on enhancing the pre-processing of manuscripts, others came up with new techniques for letter segmentation [5]. There are also researchers in favour of handcrafted feature extraction and those who strived to improve the classification layer through the optimization of classifiers, or through the merging of two or more classifiers [6].

The emphasis of this undertaking is on two primary issues: feature extraction and classification. Moments are often employed to play the role of feature extractors. [7] utilized a full-order, to realize a precise re-enactment of arbitrary-size images. [8] introduced an innovative method that served to quicken the working out of 2-D discrete moments. [9] employed image block representation for binary images, and intensity slice representation for grayscale images, to offer an original process founded on Tchebichef polynomials, to attain the translation and scale invariants of Tchebichef moments. [10, 11] crafted SCF to facilitate the extraction of local information from word images, based on the structural details of these word images.

Khémiri et al. [12] employed a multi-dimensional form of representation, deriving from ZER to remove portions of the image, devoid of any beneficial data. The method proposed by Rabi et al. [13] starts off with the modelling of the sphere centre imaging point. Subsequently, based on the imaging model, a ZER procedure, stemming from an Error Function (ERF) edge model, is employed to pinpoint the sphere image’s subpixel edge. The sub-pixel edge points, and the modelled sphere centre imaging point, are then harnessed to establish the sphere centre in the image.

The wide variety of classifiers employed for Arabic handwriting recognition include hidden Markovian models (HMMs), artificial neural networks (ANNs) [5], learning vector quantization (LVQ) [14], SVM [15] and the fuzzy min-max neural network.

A review of the Arabic database makes clear that studies on the use of the ELM and SVM techniques have been somewhat few and far between [6, 8, 9]. Furthermore, thus far, there is no documentation of research related to the use of RKELM.

4. Feature Extraction

Feature extraction, which paves the way towards an improved recognition operation, is an integral part of every handwriting recognition scheme. The accuracy of handwriting recognition is influenced by a vast variety of features. The selection of an appropriate feature is determined by the mode of the text, the system employed for processing (offline or online), and the type of script involved (handwritten or printed).

4.1. Chebyshev Moments (CM)

The fundamental operation in the handwriting recognition field is orthogonal in nature. This circumstance does away with the necessity for any discrete approximation. A system of features with orthogonal aspect, less computation and more accurate calculation which is Chebyshev. This system is taken from [16], the moment of order $p + q$ is given in Eq. (1). The approach of CM is shown in Fig. 1.

$$T_{pq} = [\rho(p,N)\rho(q,N)]^{-1} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_p(x) t_q(y) f(x,y) \quad (1)$$

where $t_n(x)$ and $\rho(p,N)$ are provided in Eqs. (2) and (3)

$$t_n(x) = n! \sum_{k=0}^{n} (-1)^{n-k} \binom{N-1-k}{n-k} \binom{n+k}{n} \binom{x}{k} \quad (2)$$

$$\rho(p,N) = \sum_{x=0}^{N-1} \{t_p(x)\}^2 \quad (3)$$

$f(x,y)$ denotes the image pixel value at index x and y

Fig. 1. The same word was written twice in Chebyshev moments.

4.2. Statistical and Contour-based Feature (SCF)

Tamen et al . [10] crafted the statistical counter features are types of spatial domain features. They are generated based on statistical and geometrical description of the differential variation of ink vs. empty areas in the image. Such a description is associated with the class of words that is written. The approach is shown in Fig. 2. It starts with finding the contour of the ink description in the image. This is based on considering that black pixels represent the ink and white pixels represent the background. The features include the general density of the image and the results of the sliding window that is calculated for each position.

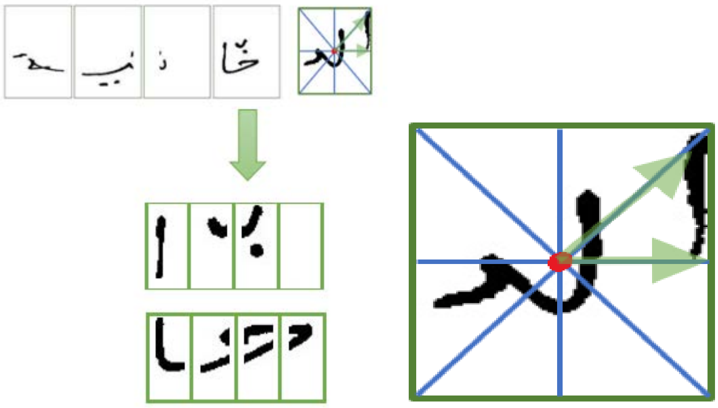
$$Features = \left(RP, IP, FDist, DensityB(j), NT(j)\right) \tag{4}$$

$where, RP$ denotes the relative position of center of mass for the window, IP denotes the points of intersections, $FDist$ denotes the farthest distance, $DensityB$ denotes the density of the block and NT number of intersections of block.

Algorithm: of calculating SCF features
Input
Image
SlidingWidow // width + height + overlap
Output
SCF
Start
1-Find the contour of the ink pixels
2-Calculate the density of Image //density= (sum of foreground pixels)/(sum of background pixels) DensityI and add it to SCF
3-Move the sliding window from right to left and find in each position RP,IP,FDist,NTi
    3.1.//RP=(relative position of center of mass.X/Width,relative position of center of mass.Y/Height)
    3.2.//IP= the points of intersection of lines passing by RP and going for the 8 Freeman directions with the contour
    3.3//FDist= the distance of the farthest point normalized by the window's widht in each direction
    3.4. Divide the window into n X m blocks and calcuate for each block density DensityB(j) and NT(j)
    //NT=number of transition from 0 to 1 in both vertical and horizontal directions
    3.5. Add (RP,IP,FDist, DensityB(j),NT(j)) to SCF, j=1,2..number of blocks
End
End

Fig. 2. Algorithm of calculating SCF features.

In order to express geometrically how we extract SCF features using the concept of window and the concept of block we give an illustrative example in Fig. 3. As we see in the Fig. 4, the word has been decomposed into five windows by moving the window in the image where each window contains a separate letter from the word. Considering that the letter itself combines a unique way of writing in the language, the features that are extracted from the window when it is positioned at the corresponding letter will be more discriminative. Furthermore, in order to reduce the effect of changes in the writing styles of letters, we use the block and direction which will produce similar values of features for similar letters despite their writing styles.



(a) Five windows and 2 x 4 **(b) Distancesblocks per window**

Fig. 3. Extraction features of SCF.

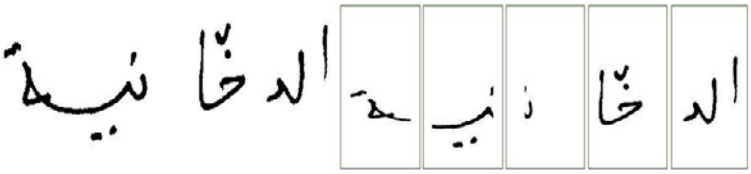


Fig. 4. Sliding window of width, height and overlap.

4.3. Zernike Moment (ZER)

Yap et al. [17] introduced Zernike Moment. Zernike polynomials are one of an infinite number of complete sets of polynomials in two variables ρ and θ which form an orthogonal basis set on the unit disk. Every Zernike polynomial [14] is combined of a real valued radial polynomial and complex azimuthal function as it is given in Eqs. (5) and (6). The approach of ZER is shown in Fig. 5.

$$V_{n,m}(\rho,\theta) = R_{n,m}(\rho)\mathrm{e}^{jm\theta} \tag{5}$$

where the radial polynomial is given as,

$$R_{nm} = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{S!(((n+|m|)/2)-s)!((n-|m|)/2)-s)!} \rho^{n-2s} \tag{6}$$

where n denotes order of the Zernike polynomial, m represents repetitions of Zernike polynomial which satisfy with the constraint of Eq. (7).

$$(n - |m|) = even$$

$$|m| \leq n \tag{7}$$

They are found mathematically by using the Eq. (8).

$$Z_{n,n} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho,\theta) V_{n,m}^{*}(\rho,\theta) \rho d\rho d\theta \tag{8}$$

where, $f(\rho, \theta)$ denotes an image function and considering that the image is given in discrete form, we write the discrete version of the equation for an image $N \times N$ in Eq. (9).

$$Z_{n,m} = \frac{n+1}{\lambda_n} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) V_{n,m}^{*}(x,y)$$

$$= \frac{n+1}{\lambda_n} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) R_{n,m}(\rho_{xy}) \mathrm{e}^{-jn\theta_{xy}} \quad (9)$$

$0 \leq \rho_{xy} \leq 1$

λ_n denotes a normalization factor, ρ_{xy} and θ_{xy} are provided in Eqs. (10) and (11).

$$\rho_{xy} = \frac{\sqrt{(2x-N+1)^2 + (N-1-2y)^2}}{N} \quad (10)$$

$$\theta_{xy} = \tan^{-1}\left(\frac{N-1-2y}{2x-N+1}\right) \quad (11)$$

x, y denotes the index of the pixels

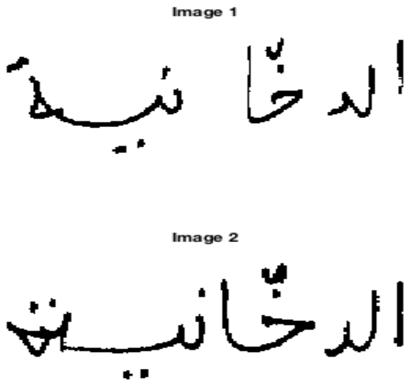


Fig. 5. The same word was written twice in Zernike moments.

5. Classification

The domain of handwriting recognition has been deeply studied for a couple of decades by researchers who have utilized dissimilar algorithms, like SVM, Multi-Layer Perceptron (MLP), HMM, Deep Networks (DNN), Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). The outcomes were various and satisfactory. These machines learning (ML) systems have demonstrated their reliability and performance in a large domain of applications as well as triumphing in optical character recognition (OCR) for Latin and Asian languages [18]. The major drawback of these architectures is the large number of parameters, so over-fitting can occur. This research focuses on applying these classifiers ELM, SVM and RKELM in domain of offline Arabic handwriting recognition words.

5.1. Extreme Learning Machine (ELM)

Conceived by [19, 20], this process is based on the fact that learning through examples, is a simple and straightforward process, for deciphering the status of an object. This is achievable without the need for much training, or any specific learning structure. Huang et al. [21] forwarded that the human brain is partially occupied by arbitrary neurons, whose activities are totally impervious to their surroundings. The technique they developed from this concept came to be known as the extreme learning machine (ELM). This technique has proven its ability to prevail over the drawbacks plaguing the back-propagation gradient algorithm, in the context of incompatible settings and a sluggish learning rate. Initially developed for the learning of a single layer feedforward neural network (SLFN), the ELP was later extended to include networks not requiring the support of a neural framework. The target output of the ELM can be calculated by the following equation:

$$t_k^i = \sum_{j=1}^{M} \beta_{kj}\, g_j(W, b, X)\,, k = 1,2, \dots, l. \tag{12}$$

Here the input weight is W and output weights are β where b is hidden layer, the matrix W and b are generated randomly. g_j is the activation function of hidden layer from Eq. (12).

$$T = \mathrm{H}\beta \tag{13}$$

where H is the output matrix of hidden layer and defined as:

$$\mathrm{H}(W, b, X) = \begin{bmatrix} g(w_1.x_1 + b_1) & \cdots & g(w_M.x_1 + b_M) \\ \vdots & \cdots & \vdots \\ g(w_1.x_N + b_1) & \cdots & g(w_M.x_N + b_N) \end{bmatrix}_{N\times M}$$

$\beta = [\beta_1, \beta_2, \dots, \beta_M]^T_{lxM}$ and

$$T = [t_1, t_2, \dots, t_N]^T_{lxM} \tag{14}$$

The output weight matrix $\beta = [\beta_1, \beta_2, \dots, \beta_M]^T_{lxM}$ can be determined analytically by the minimum norm least square solution:

$$\tilde{\beta} = \arg min_{\beta}\ ||\mathrm{H}\beta|| = \mathrm{H}^{+}T \tag{15}$$

where H^{+} is the M-P generalized inverse of H. if the H is non-singular, Eq. (15) can be re written as:

$$\tilde{\beta} = \mathrm{H}^{+}\mathrm{H}^{-1}\,\mathrm{H}^{T}T \tag{16}$$

5.2. Support Vector Machine (SVM)

Provided with a compilation of training data $(x_i, c_i)^N\ i = 1$, in which $x_i \in R$ and $c_i \in \{-1, 1\}$, SVM successfully identifies the best possible separating hyper plane, to fully extend the space between the two classes [22, 23] by means of:

$$minimizing: D = \frac{1}{2} w.w + C \sum_{i=1}^{N} \in_i) \tag{17}$$

depending on:

$c_i(w.x_i + b) \geqslant 1 - \in_i,\ \forall i \quad \in_i \geqslant 0,\ \forall i$

In which C is a parameter that facilitates an exchange between diminishing the training error and extending the distance $\frac{2}{||w||}$, of the dividing space of the two dissimilar classes. The data points utilized to locate the optimal hyper plane are known as vectors. Several kernel functions are required to regulate SVM parameters specific to the classification jobs where Gaussian RBF is crucial. There are three basic kernel functions which are shown in following equation:

- Polynomial kernel

$$k(x_i, x_j) = (s(x_i, x_j) + c)^d, d = 2,3,4\ ) \tag{18}$$

- Radial Basic Function kernel

$$k(X_I, y_j) = \exp\ (-\gamma \parallel X_i - X_j \parallel^2) \tag{19}$$

- Sigmoid kernel

$$k(x_i, x_j) = \tanh\ (ax_i . x_j - b) \tag{20}$$

5.3. Reduced kernel extreme learning machine (RKELM)

Introduced by Deng et al. [24] for a fast and accurate algorithm to obtain a nonlinear separating surface. Deng et al. [25] proved that RKELM can approximate any nonlinear functions accurately. In machine learning a kernel function is a measure of similarity (often based on a function involving a dot product) between samples of input data defined over a feature space $\mathcal{H}$ such that $\Phi: x \to \mathcal{H}$ notated as $k(x.x') = k(x_i.x_j)$.

The output of RKELM is given by the following equations:

$$f(x) = K(x, x') \left(\frac{1}{\lambda} + k(x, x')\right)^{-1} T \tag{21}$$

Equivalently:

$$f(x) = \begin{bmatrix} k(x, x'_1) \\ \vdots \\ k(x, x'_N) \end{bmatrix} \left(\frac{1}{\lambda}\ \Omega ELM_{ij}\right)^{-1} T \tag{22}$$

where λ is the stability factor and the kernel function defined on the base matrix H as:

$$\Omega ELM = HH^T = h(x_i)h(x_j) = K(x_i, x_j) = K(x, x') \tag{23}$$

Kernel-based learning approaches by potentially utilize a large amount of system memory for machine learning problems with huge datasets. Thus, a modification to KELM called "Reduced kernel Extreme Learning Machine" has been proposed. The RKELM take the KELM and instead of computing $k(x.x')$ over the entire set of input data, they compute $k(\tilde{x}.\tilde{x}')$ where $\tilde{x}$ is a randomly chosen subset of the input data. However, to the best of our knowledge, there is not one applied RKELM on Arabic handwriting word recognition.

6. Research Method

This work investigates three feature extractors techniques CM, SCF and ZER for offline Arabic handwritten recognition words and analysing the performance of each technique, the following section explains the details of our method.

6.1. Dataset

Pechwitz et al. [26] created the IFN/ENIT database by the Institute of Communications Technology (IFN) at Technical University Braunschweig in Germany and the Ecole Nationale d'Ingenieurs de Tunis (ENIT) in Tunisia.

Version 2.0 of this database consists of 26,459 images of the 937 names of cities and towns in Tunisia, contains 5 sets of *a b c d e* written by 411 different writers. The database contains 115,585 pieces of Arabic words (PAWs) and 212,211 characters. This database has been used and endorsed by many researchers of Arabic handwritten text recognition. A competition of Arabic handwritten text recognition was conducted using this database in 2005, 2007 and 2009. This database is limited to city names and thus contains limited vocabulary. As mentioned that feature extractors require normalization as images have different sizes, the efficiency of the CM, ZER and SCF were tested on different sets of images taken from *a b c d e* sets of the IFN/ENIT database. We have implemented two different scenarios, each set constituted by 22 and 10 classes of maximum and minimum number of images per class given in Table 1 During the recognition process, the dataset split for both Set 1 and Set 2 by approximating 70% of the total images in the considered sets are used for train and rest for test.

Table 1. Summary of the scenarios.

Data	Total	Train	Test	Max no. per class	Min no. per class	No. classes
Set 1	2946	2057	889	371	129	10
Set 2	5443	3800	1643	371	53	22

6.2. Normalization

Normalization methods aim to remove all types of variations from the written and standardized datasets obtained by normalizing all our inputs to a standard scale, this allows the classifier to perform quicker in learning the optimal parameters for each image. Additionally, it is useful to ensure that our inputs are roughly in the range of [-1 1] to avoid miscalculation, especially if input and target output are completely different. Equation defined as follows:

$$feature(x') = 2\left(\frac{x - \min(feature(x))}{\max(feature(x)) - \min(\min(x))}\right) - 1 \tag{11}$$

6.3. Research methodology

This work conducted intensive experiments to evaluate the proposed system. We took into consideration the effectiveness number of classes and shapes of an image. Each feature was through a few steps staring at set different shapes then normalized and split into train and test to evaluate the performance of each classifier.

Each feature has been extracted with different parameters such as number of images and shape. The generated features have been tested based on concepts of extractor such as moments and number of orders to get the accurate result. After selecting the fit parameters to generate features.

As already stated, the CM, ZER and SCF features have been trained separately with different shapes and windows, resulting in nine classifiers: ELM_{CM}, ELM_{ZER}, ELM_{SCF}, SVM_{CM}, SVM_{ZER}, SVM_{SCF}, $RKELM_{CM}$, $RKELM_{ZER}$ and $RKELM_{SCF}$ that are compared to each other. Further, we train SVM and RKLEM with different kernels and ELM with a set of different neuron network and activation functions to analyse the performance of ELM classifier. The research methodology is illustrated in Fig. 6.

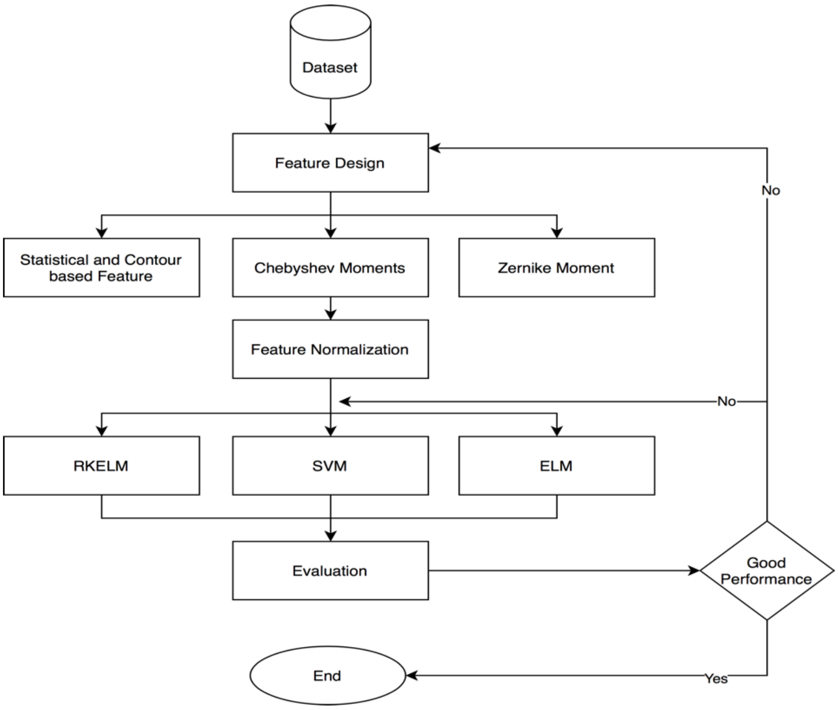


Fig. 6. Research methodology.

7. Results and Discussion

In the implementation, each one of these extractors has its own technique to extract information by either segmenting the image into small blocks (windows) such as SCF or scanning the whole image like CM and ZER to get relevant and interesting features. Furthermore, CM and ZER are based on the selected moments and number of orders, the experiment aims to find the best number of moments to proceed for the rest of the experiments. As shown in Table 2 the optimal moment for our experiment is 12 that achieved highest accuracy 90%, 88% with SVM and ELM respectively. Same experiment was carried out for ZER. The selecting order of moments from 4 to 20, the optimal were 10 moments with consideration of datasets.

Table 2. Selecting the optimal moment for CM.

Moments	**2**	**4**	**6**	**8**	**10**	**12**	**14**
SVM	71%	79%	84%	90%	89%	**90%**	90%
ELM	71%	77%	81%	85%	87%	**88%**	88%

As mentioned there were three classifiers for experiments, each classifier has its own technique and parameters to train the features as shown in Table 3 that ELM is based on activation function and neurons network. The experiment was conducted for testing the number of neurons with a range of 100 until 1000 neurons in a hidden layer. The selected neuron has been tested with five activation functions: Sigmoid, Sine, Hardlim, Tribas and Radbas. The best accuracy is accomplished for 200 neurons with Sigmoid function. The same experiment was repeated for SCF and ZER including all the shapes.

Table 3 illustrated the performance of activation functions in ELM. It seems likely that Sigmoid and Sine have the highest testing accuracy. The rest of activation functions had some fluctuations in performance. However, the rest of the experiment showed that Sigmoid is the best activation due to being differentiable across the entire domain and is easy to compute, also this function shows the probability as well.

Table 3. Validation accuracy for Set 1 CM features by 80 x 80.

Neurons	Sigmoid	Sine	Hardlim	Tribas	Radbas
100	85%	86%	28%	80%	85%
200	**88%**	88%	32%	82%	87%
300	87%	86%	38%	83%	87%
400	86%	85%	43%	83%	85%
500	84%	83%	43%	81%	84%
600	82%	81%	44%	82%	81%
700	79%	80%	51%	82%	81%
800	78%	77%	45%	82%	78%
900	77%	76%	48%	80%	77%
1000	72%	72%	49%	80%	73%

Table 4 provides an analysis of kernels in SVM, kernel is used to map our input space into high dimensionality feature space. The experiment was conducted with three kernels in SVM namely: Polynomial, RBF and Linear. All the features were tested with kernels to find the best accuracy. There are no such guarantees for one kernel to work better than the other. On the other hand, practical perspective tells us that RBF kernels in SVM are less prone to overfitting and much faster to train compared to rest of kernels. RBF kernel shows the non-linearity aspect of the data of the features which have given the RBF kernel higher performance.

Table 4. Validation the kernels of SVM for Set 1 CM features.

Kernels	Polynomial	RBF	Linear
Accuracy	84%	**90%**	88%

Same experiment was repeated for RKELM that contains five kernels: polynomial, RBF, IMQ, Sigmoid and Linear. Each kernel consists of one or more parameters, for which high performance is required as demonstrated in Table 5. The highest accuracy achieved with a polynomial kernel.

Table 5. Validation accuracy of RKELM with Set1 CM features.

Kernel	Number of order				
Polynomial	[0,5] 58%	[1,6] 81%	[0,3] 64%	[3,3] 82%	[2,5] 86%
RBF	2^2 77%			2^1 74%	
IMQ	[2,0.9] 75%			[1,0.9] 77%	
Sigmoid			[2^2,2] 20%		
Linear			[] 61%		

In Table 6, the obtained results of CM features with different shapes of image to determine how various shapes affect the recognition process. Different image shapes carry different information that's why the best image shape needs to be examined in detail. The purpose of image reshaping is to produce lower intense features which hastens the processing time.

Table 6 illustrates the CM feature trained with six classifiers and two datasets. The most noticeable size (80×80) had the highest accuracy in Set 1 at 90.59% with SVM classifier, the accuracy for set 2 dropped due to the increase in the number of classes. There has been a slight fluctuation in accuracy for different shapes of each classifier. Overall, it is clear the SVM had the highest accuracy in set 1, despite the fact the rest of classifiers are trained on the same shape. In set 2 there is slight improvement in accuracy by increasing the shape of images and the percentage of RKELM remained steady throughout nearly 76.61% with 128×128 shape.

Table 6. Results of CM with different shapes of image.

Dataset	Classifier	80 x 80	128 x 128	250 x 250
SET 1	SVM	90.59%	89%	90.05%
	ELM	88.40%	88.11%	88.18%
	RKELM	86.81%	86.50%	85%
SET 2	SVM	87%	88.35%	88.09%
	ELM	83%	83%	84%
	RKELM	76.02%	76.61%	76.30%

Table 7 provides results of ZER with six classifiers on two datasets. The highest accuracy achieved on SVM with (250x250) size nearly 76% close to other shapes and far lacking compared to other classifiers ELM and RKELM. On the same Set 1, ELM received accuracy close to 64% with 400 neurons and RKELM achieved good accuracy with polynomial kernel at shape size (128x128). On the other hand, set 2 was the highest accuracy with SVM linear kernel with shape (80x80), still a huge gap compared to other classifiers. In set 2 ELM were lacking far behind SVM due to random neurons between input layer and hidden layer.

Table 7. Results of ZER with different shapes of image.

Dataset	Classifier	80 x 80	128 x 128	250 x 250
SET 1	SVM	75%	74%	**76%**
	ELM	61%	**64%**	63%
	RKELM	66%	**68.19%**	68.10%
SET 2	SVM	**67%**	67%	67%
	ELM	54%	**55%**	54%
	RKELM	56%	56%	**57%**

Table 8 provides results of SCF with different window sizes because the images in datasets had different width as some of the images had more than one word. This feature took into consideration the segment image splitting into windows and blocks per window. We implemented this experiment on three different window sizes and two blocks each window with overlapping to capture the local information. The most noticeable changes were that by increasing the window size it impacted the accuracy. On set 1, the RKELM classifier achieved highest accuracy at 94% with 6 windows compared to ELM and SVM. However, we could notice the accuracy did not change much by increasing the number of classes. In Table 9, We compared our method's results to those of five other methods used to recognize Arabic handwritten word. Our proposed method successfully identified results of 94% for SET 1 and 91.80% for SET 2 with RKELM classifier.

Table 8. Results of SCF with different window sizes.

Dataset	Classifier	[4[2,3]]	[4[2,4]]	[5[2,3]]	[5[2,4]]	[6[2,3]]	[6[2,4]]
SET 1	SVM	90%	91%	91%	92%	92%	93%
	ELM	90%	91%	92%	93%	92%	92%
	RKELM	87%	90%	92%	93%	93%	94%
SET 2	SVM	89%	88%	90%	91.80%	91.77%	90%
	ELM	89%	91.50%	91.58%	91.65%	90%	90%
	RKELM	81%	84%	88%	90.20%	89%	90.06%

Table 9. Comparison of the accuracy with other methods in same dataset.

Feature	Dataset/Number of words	Accuracy
Statistical geometric feature [27]	IFN/ENIT, 6000 words	80.78%
Gabor feature [28]	IFN/ENIT, 12000 words	55.54%
Zernike and HU moments [29]	IFN/ENIT, 6000 words	78.5%
Concavity and distribution feature [30]	IFN/ENIT, 12000 words	87%
Baseline estimation [31]	IFN/ENIT, 6000 words	87.93%
This work	IFN/ENIT, 2946, 5443 words	CM=90.59%,88.35%. ZER=76%,67%. SCF=94%,91.80%

8. Conclusion

This paper presented three feature extractors CM, ZER and SCF. We took into consideration the shape and number of windows. Changing one shape parameter might affect some other parameters. The experiments were evaluated on IFN/ENIT images with unbalanced datasets. Then nine classifiers trained separately: ELM_{CM}, ELM_{ZER}, ELM_{SCF}, SVM_{CM}, SVM_{ZER}, SVM_{SCF}, $RKELM_{CM}$, $RKELM_{ZER}$ and $RKELM_{SCF}$ and each one of these classifiers have been trained with different shapes and number of windows. In ELM, Sigmoid was the best activation due to being differentiable across the entire domain and easy to compute. On other hand, the linear kernel showed much faster train compared to other kernels and RKELM showed significant results with polynomial kernel. As future work we plan to implement these techniques to other databases with a balance dataset to avoid overfitting in training.

Acknowledgment

The authors would like to thank Ministry of Higher Education Malaysia for supporting this research under fundamental Research Grant Scheme, Vot K213 (FRGS/1/2019/ICT02/UTHM/02/2) and Universiti Tun Hussein Onn Malaysia for Multidisciplinary Research Grant, Vot H511.

Nomenclatures

b	Hidden layer
C	Facilitates an exchange between diminishing the training error and extending the distance
DensityB	Density of block
FDist	Farthest distance
g_j	Activation function
H	Output matrix of hidden layer
H^+	Generalized inverse
IP	Points of intersections
m	Repetitions of Zernike
NT	Number of intersections of block
n	Donate order of Zernike polynomial
P, θ	Form an orthogonal basis set
Rnm	Radial polynomial
RP	Relative position of center
Tpq	Moments of order
t_k^i	Target out of ELM
W	Input weights
x,y	Image pixels
$\tilde{x}$	Random subset of data
x'	Normalized data
Greek Symbols	
β	Output weights
$\tilde{\beta}$	Minimum norm least square
λ_n	Normalization factor
λ	Stability factor

References

1. Azeem, S.A.; and Ahmed, H. (2013). Effective technique for the recognition of offline Arabic handwritten words using hidden Markov models. *International Journal on Document Analysis and Recognition*, 16(4), 399-412.
2. Ghadhban, H.Q.; Othman, M.; Samsudin, N.A.; Ismail, M.N.; and Hammoodi, M.R. (2020). Survey of offline arabic handwriting word recognition. *International Conference on Soft Computing and Data Mining*. 358-372.
3. Chen, L.; Yan, R.; Peng, L.; Furuhata, A.; and Ding, X. (2017). Multi-layer recurrent neural network based offline Arabic handwriting recognition. *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*. Nancy, France, 6-10.

4. Wikipedia, contributors. (2019). Arabic. *Wikipedia, The free encyclopedia*. [Online].Available:https://en.wikipedia.org/w/index.php?title=Arabic&oldid =879025032.
5. AbdulKader, A. (2006). A two-tier Arabic offline handwriting recognition based on conditional joining rules. *Summit on Arabic and Chinese Handwriting Recognition*, 70-81.
6. Elleuch, M.; Lahiani, H.; and Kherallah, M. (2015). Recognizing arabic handwritten script using support vector machine classifier. *15th International Conference on Intelligent Systems Design and Applications (ISDA)*. Marrakech, Morocco, 551-556.
7. Bayraktar, B.; Bernas, T.; Robinson, J.P.; and Rajwa, B. (2007). A numerical recipe for accurate image reconstruction from discrete orthogonal moments. *Pattern Recognition*, 40(2), 659-669.
8. Althobaiti, H.; and Lu, C. (2018). Arabic handwritten characters recognition using support vector machine, normalized central moments, and local binary patterns. *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*. 121-127.
9. Hassan, A.K.A.; Mahdi, B.S.; and Mohammed, A.A. (2019). Arabic handwriting word recognition based on scale invariant feature transform and support vector machine. *Iraqi Journal of Science*, 60(2), 381-387.
10. Tamen, Z.; Drias, H.; and Boughaci, D. (2017). An efficient multiple classifier system for Arabic handwritten words recognition. *Pattern Recognition Letter*, 93, 123-132.
11. Ghosh, M.; Malakar, S.; Bhowmik, S.; Sarkar, R.; and Nasipuri, M. (2019). Feature selection for handwritten word recognition using memetic algorithm. *Advances in Intelligent Computing*, 103-124.
12. Khémiri, A.; Echi, A.K.; and Elloumi, M. (2019). Bayesian Versus Convolutional networks for arabic handwriting recognition. *Arabian Journal for Science and Engineering*, 1-19.
13. Rabi, M.; Amrouch, M.; and Mahani, Z. (2017). Hybrid HMM/MLP models for recognizing unconstrained cursive arabic handwritten text. *International Conference on Advanced Information Technology, Services and Systems*. 438-448.
14. Alizadeh, E.; Lyons, S.M.; Castle, J.M.; and Prasad, A. (2016). Measuring systematic changes in invasive cancer cell shape using Zernike moments. *Integrative Biology*, 8(11), 1183-1193.
15. Fredo, A.R.J.; Abilash, R.S.; Femi, R.; Mythili, A.; and Kumar, C.S. (2019). Classification of damages in composite images using Zernike moments and support vector machines. *Composite Part B : Engineering*, 168, 77-86.
16. Mukundan, R.; Ong, S.H.; and Lee, P.A. (2001). Image analysis by Tchebichef moments. *IEEE Transaction on Image Processing*, 10(9), 1357-1364.
17. Yap, P.T.; Raveendran, P.; and Ong, S.H. (2001). Chebyshev moments as a new set of moments for image reconstruction. *IJCNN'01. International Joint Conference on Neural Networks. Proceedings.* Washington, USA, 4, 2856-2860.
18. Cireşan, D.; and Meier, U. (2015). Multi-column deep neural networks for offline handwritten Chinese character classification. *2015 International Joint Conference on Neural Networks(IJCNN)*. Killarney, Ireland, 1-6.

19. Huang, G.B.; Chen, L.; and Siew, C.K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transaction on Neural Networks*, 17(4), 879-892.
20. Huang, G.B.; (2014). An insight into extreme learning machines: random neurons, random features and kernels. *Cognitive Computation*, 6(3), 376-390.
21. Huang, G.B.; Zhou, H.; Ding, X.; and Zhang, R. (2011). Extreme learning machine for regression and multiclass classification. *IEEE Transaction on System, Man, and Cybernetics, Part B(Cybernetics)*, 42(2), 513-529.
22. Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 121-167.
23. Huang, W.; Chen, C.; Sarem, M.; and Zheng, Y. (2008). Overlapped rectangle image representation and its application to exact legendre moments computation. *Geo-spatial Information Science*, 11(4), 294-301.
24. Deng, W.; Zheng, Q.; and Zhang, K. (2013). Reduced kernel extreme learning machine. *Proceedings of the 8th International Conference On Computer Recognition Systems CORES 2013*. 63-69.
25. Deng, W.Y.; Ong, Y.S., and Zheng, Q.H. (2016). A fast reduced kernel extreme learning machine. *Neural Networks*, 76, 29-38.
26. Pechwitz, M.; Maddouri, S.S.; Märgner, V.; Ellouze, N.; and Amiri, H. (2002). IFN/ENIT-database of handwritten Arabic words. *Proceeding of CIFED*, 2, 127-136.
27. Tavoli, R.; Keyvanpour, M.; and Mozaffari, S. (2018). Statistical geometric components of straight lines (SGCSL) feature extraction method for offline Arabic/Persian handwritten words recognition. *IET Image Processing*, 12(9), 1606-1616.
28. Al-Khayat, M. (2014). *Learning-based Arabic word spotting using a hierarchical classifier*. Ph.D. Thesis. Concordia University.
29. Jayech, K.; Mahjoub, M.A.; and Ben Amara, N.E. (2016). Arabic handwritten word recognition based on dynamic bayesian network. *International Arab Journal of Information Technology*, 13(6B), 1024-1031.
30. Metwally, A.H.; Khalil, M.I.; and Abbas, H.M. (2017). Offline Arabic handwriting recognition using hidden Markov models and post-recognition Lexicon matching. *Computer Engineering and Systems (ICCES), 12th International Conference on*. 238-243.
31. Rabi, M.; Amrouch, M.; and Mahani, Z. (2018). Cursive Arabic handwriting recognition system without explicit segmentation based on hidden markov models. *Journal of Data Mining and Digital Humanities*.