International Journal of Artificial Intelligence and Applications (IJAIA), Vol.14, No.6, November 2023

LEARNING SPLINE MODELS WITH THE EM ALGORITHM FOR SHAPE RECOGNITION

Abdullah A. Al-Shaher, Yousef S. AlKhawari

Department of Computer Science and Information Systems, College of Business Studies, Public Authority for Applied Education and Training, Kuwait

ABSTRACT

This paper demonstrates how cubic Spline (B-Spline) models can be used to recognize 2-dimension non-rigid handwritten isolated characters. Each handwritten character is represented by a set of non-overlapping uniformly distributed landmarks. The Spline models are constructed by utilizing cubic order of polynomial to model the shapes under study. The approach is a two-stage process. The first stage is learning, we construct a mixture of spline class parameters to capture the variations in spline coefficients using the apparatus Expectation Maximization algorithm. The second stage is recognition, here we use the Fréchet distance to compute the variations between the spline models and test spline shape for recognition. We test the approach on a set of handwritten Arabic letters.

KEYWORDS

Spline Models, Cubic order of Spline Curves, handwritten Arabic Characters, Shape Recognition, Recognition by Alignment. Expectation Maximisation Algorithm, Unsupervised Learning, Fréchet distance.

1. INTRODUCTION

Shape matching has been the focus of many researchers for the past seven decades [1] and attracted many scholars in the field of pattern recognition [2], artificial intelligence [3], signal processing [4], image analysis [5], and computer vision [6]. The difficulties arise when the shape under study exhibits a high degree in shape variation: as in handwritten characters [7], digits [8], face detection [9], and gesture authentication [10]. For a single data, shape variation is limited and cannot be captured ultimately because single data does not provide sufficient information and knowledge about the data; therefore, multiple existence of data provide better understanding of shape analysis and manifested by mixture models [11]. Because of the existence of multivariate data under study, there is always the requirement to estimate the parameters that describe the data that is encapsulated within a mixture of shapes. Extensive efforts have been invested in analyzing alphabet of English [12], Latin [13], Chinese [14], Hebrew [15], and Hindi [16]; however, very little work has been done in analyzing handwritten Arabic characters and least considerable results achieved [17]. The subject of investigating Arabic letters shape analysis is complicated. Such difficulties arise from the nature structure of Arabic alphabet and words. Arabic script is cursive, continues, and similar without diacritics. For example: the word عل م has different meaning when secondaries are attached to the letters.

عِلْمْ	عَلَمْ	عُلِمْ	عَلَّمَ
Science	Flag	Noted	Taught

The literature demonstrates many statistical and structural approaches with various algorithms to model shape variations using supervised and unsupervised learning [18] algorithms. In precise, comes the powerful Expectation Maximization algorithm (EM) of Dempster [19] that has been used widely for such cases. The EM algorithm revolves around two step procedures. The expectation E step revolves around estimating the parameters of a log likelihood function and pass it to the Maximization M step. In a maximization (M) step, the algorithm computes parameters maximizing the expected log-likelihood found on the E step, the process is iterative one until all parameters come to stability. For instance, Jojic and Frey [20] have used the EM algorithm to fit mixture models to the appearance manifolds for faces. Bishop and Winn [21] have used a mixture of principal components analyzers to learn and synthesize variations in facial appearance. Vasconcelos and Lippman [22] have used the EM algorithm to learn queries for content-based image retrieval. Finally, several authors have used the EM algorithm to track multiple moving objects [23]. Revov et al. [24] has developed a generative model which can be used for handwritten character recognition. Their method employs the EM algorithm to model the distribution of sample points.

Spline models have been used widely in the signal processing community [25]. Spline models can describe more complex shapes that contain circles, closed contours, ellipses, and curved shapes and combine this description in mathematical and statistical framework. For example, Splines are used to model Functional data analysis [26], Computer Aided Design [27], automobiles [28], smoothed surfaces [29], curly shapes [30], and noisy materials such as clothing design [31]. Safraz [32] has investigated the issue of Arabic shape recognition by utilizing edge detection of machine printed letters to bring the shape under landmarks representation, then applying cubic polynomials on them. In deepest investigation, Safraz [33] addressed the subject of extracting the curve estimation by finding the optimal set of coefficients of Bezier function and dividing the curve into multiple segments ending with a smooth curve fitting.

In this research paper, we demonstrate how Spline models are used to recognize 2D handwritten shapes by fitting 3rd order of polynomial function to a set of landmarks points extracted for the shape under analysis. We then, train such Spline models to capture the optimal characteristics of the shapes in the training sets of shapes. Handwritten isolated Arabic characters are used and tested in this investigation.

2. SHAPE MODELLING

A spline curve is a mathematical representation to construct an interface that allows sequence of points $(x_i, y_i)\ \ for\ i = 1 \dots n$ to design and control the shape under study.

The parametric form of shape coordinates is called control points. The core concept of pattern recognition methods and approaches aim at constructing a classifier for a set of training patterns. Such a classifier determines which shape-class the input pattern belongs to; therefore, is known as object training or shape learning [34][35]. In real world applications such as object recognition, a certain previous knowledge is required beforehand. The utilization of such knowledge into a training set is the key element that will allow to increase of performance of shape classification.

Following the concept of constructing a training set of patterns undergo classification. The landmark patterns are collected as the object in question undergoes representative changes in shape. To be more formal, each landmark pattern consists of L labelled points. Suppose that there are T landmark patterns. The tth training pattern is represented using the long vector of landmark co-ordinates:

$$X_t = (x_1, y_1, x_2, y_2, x_3, y_3, \dots, x_T, y_T)^T \tag{11}$$

where the subscripts of the co-ordinates are the landmark labels. A spline curve is a curve that passes through these tth training pattern co-ordinates called interpolating curves. To establish interface, we approach the polynomial in the form of function of landmarks:

$$f(x,y) = y - a + bx + cx^2 + dx^3 \quad \text{.....} \tag{12}$$

+

The degree of the polynomial corresponds with the highest coefficient that is nonzero. The aim of using cubic spline is to provide a smooth curve that passes through the chosen landmarks for shape construction as well as support inflection points. Equation (12) can be revised in the form of cubic polynomials.

$$y = a + bx + cx^2 + dx^3 \tag{13}$$

Equation (13) can be rewritten in the form of solving linear equations to solve for the spline coefficients (a, b, c, d) obtained by the curve segment of 4 coordinates. The new mathematical formulation to solve for required coefficients, we present a matrix of linear equations.

$$Ma = Y \tag{14}$$

Where M is the matrix of linear equations

$$\begin{bmatrix} 1 & bx_i & cx_i^2 & dx_i^3 \\ 1 & bx_{i+1} & cx_{i+1}^2 & dx_{i+1}^3 \\ 1 & bx_{i+2} & cx_{i+2}^2 & dx_{i+2}^3 \\ 1 & bx_{i+3} & cx_{i+3}^2 & dx_{i+3}^3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} y_i \\ y_{i+1} \\ y_{i+2} \\ y_{i+3} \end{bmatrix} \tag{15}$$

The solution is

$$a = M^{-1} Y \tag{16}$$

Similar procedure is applied to the remaining coordinate segments. The resulting set of spline coefficients presented by

$$Y = (a_i, b_i, c_i, d_i, a_{i+1}, b_{i+1}, c_{i+1}, d_{i+1}, \ldots , a_L, b_L, c_L, d_L \tag{17}$$

And in unification form of the set spline coefficients to be

$$Y = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \ldots , \alpha_{L-1}, \alpha_L) \tag{18}$$

Suppose that each shape class ω have T training spline mean coefficients Υ_ω. The mean vector of is represented by

$$\mu = \frac{1}{T}\sum_{t=1}^{T} Y_t \tag{19}$$

The covariance matrix is then constructed.

$$\Sigma = \frac{1}{T}\sum_{t=1}^{T}(Y_t - \mu)\,(Y_t - \mu)^T \tag{20}$$

3. LEARNING SPLINE MODEL

In machine learning approaches [21], supervised learning depends on data labelling which the outcomes models are mapping algorithms between data and its counterparts. On the other hand, unsupervised learning data patterns assumed from the unlabeled input data, hence, the goal of unsupervised learning is to find the structure and variations from the input data. In another format, Unsupervised learning does not need any supervision architecture. Instead, it finds data variations and variable modes from the data by itself. To reproduce more complex variations in shape either a non-linear deformation or a series of local piecewise linear deformations must be employed. In this paper we adopt an approach based on mixtures of spline model distributions. Our reasons for adopting this approach are twofold. First, we would like to be able to model more complex deformations by using multiple modes of shape deformation. This arises when a set of training patterns contains examples from different shape classes. The second reason is that shape variations cannot be captured in a single mode of shape deformation, hence a mixture is constructed. We commence by assuming that the individual examples in the training set are conditionally independent of one another. The following approach is based on fitting a Gaussian mixture model to the set of training examples of spline coefficients. We further assume that the training data can be represented by a set of spline-classes **Ω**.

Each spline coefficient model **ω** has its own mean coefficients μ_ω and covariance matrix Σ_ω. With these ingredients we establish the likelihood function for the set of the curve patterns in

$$p(Y_t, t = 1 \dots T) = \prod_{t=1}^{T}\sum_{w=1}^{\Omega} p(Y_t \,|\mu_w, \Sigma_w) \tag{21}$$

Where the term $p(Y_t|\mu_w, \Sigma_w)$ is the probability for drawing curve pattern αt from the curve-class ω. Associating the above likelihood function with the Expectation Maximization algorithm, the likelihood function can be written to be iterative process of two steps. The process revolves around estimating the expected log-likelihood function iteratively in

$$q_L\left(C^{(n+1)}\middle|C^{(n)}\right) = \sum_{t=1}^{T}\sum_{w=1}^{\Omega} P(Y_t, \mu_w^{(n)}, \Sigma_w^{(n)})\ X\ ln\ p(Y_t|\mu_w^{(n+1)}, \Sigma_w^{(n+1)}) \tag{22}$$

Where the quantity and $\mu_w^{(n)}$ and $\Sigma_w^{(n)}$ are the estimated mean curve vector and variance covariance matrix both at iteration (n) of the algorithm. The quantity $p(Y_t, \mu_w^{(n)}, \Sigma_w^{(n)})$ is the a posteriori probability that the training pattern spline coefficients belong to the spline-class ω at iteration n of the algorithm. The term $p(\Upsilon_t|\mu_w^{(n+1)}, \Sigma_w^{(n+1)})$ is the probability of distribution of spline-pattern αt belonging to spline-class ω at iteration (n + 1) of the algorithm; thus, the probability density to associate curve- patterns Υ_t for (t = 1 … T) to class spline-class ω are estimated by the updated construction of the mean-vector $\mu_w^{(n+1)}$, and covariance matrix $\Sigma_w^{(n+1)}$ at iteration n+1 of the algorithm. According to the EM algorithm, it revolves around estimation of the expected log-likelihood function within two iterative processes. In the M or maximization step of the algorithm, our aim is to maximize the curve mean-vector $\mu_w^{(n+1)}$, and covariance matrix $\Sigma_w^{(n+1)}$, while, in the E or expectation step, the aim is to estimate the distribution of spline-patterns at iteration n along with the mixing proportion parameters for curve-class ω.

In the E, or Expectation step of the algorithm, the a posteriori spline-class probability is updated by applying the Bayes factorization rule to the spline-class distribution density at iteration (n+1). The new estimate is computed by

$$p\left(\Upsilon_t, \mu_w^{(n)}, \sum_w^{(n)}\right) = \frac{p(\Upsilon_t|\mu_w^{(n)}, \Sigma_w^{(n)})\, \pi_w^{(n)}}{\sum_{w=1}^{\Omega} p(\Upsilon_t|\mu_w^{(n)}, \Sigma_w^{(n)})\, \pi_w^{(n)}} \tag{23}$$

Where the revised spline-class ω mixing proportions $\pi_w^{(n+1)}$ at iteration (n + 1)is computed in

$$\pi_w^{(n+1)} = \frac{1}{T}\sum_{t=1}^{T} p(\Upsilon_t|\mu_w^{(n)}, \Sigma_w^{(n)}) \tag{24}$$

With that at hand, the distributed spline-pattern Υ t to the class-spline ω is Gaussian distribution and is classified according to

$$p\left(\Upsilon_t \middle| \mu_w^{(n)}, \Sigma_w^{(n)}\right) = \frac{1}{(2\pi)^L \sqrt{|\Sigma_w^{(n)}|}}\, exp\left[-\frac{1}{2}\left(\Upsilon_t - \mu_w^{(n)}\right)^T X \left(\sum_w^{(n)}\right)^{-1} X\ (\Upsilon_t - \mu_w^{(n)})\right] \tag{29}$$

In the M, or Maximization step, our aim is to maximize the curve-class ω parameters. The updated spline mean-vector $\mu_w^{(n+1)}$ estimate is computed by

$$\mu_w^{(n+1)} = \sum_{t=1}^{T} p(\Upsilon_t, \mu_w^{(n)}, \Sigma_w^{(n)})\Upsilon_t \tag{30}$$

And the new estimate of the spline-class covariance matrix is weighted by

$$\Sigma_w^{(n+1)} = \sum_{t=1}^{T} p\left(\Upsilon_t, \mu_w^{(n)}, \sum_w^{(n)}\right) X(\Upsilon_t - \mu_w^{(n)}) \left(\Upsilon_t - \mu_w^{(n)}\right)^T \quad (31)$$

Both E, and M steps are iteratively converged, the outcome of the learning stage is a set of spline-class ω parameters such as $\mu_w^{(n)}$ and $\Sigma_w^{(n)}$, hence the complete set of all spline-class **Ω** are computed and ready to be used for recognition.

4. SPLINE MATCHING

With the ingredients produced by the previous learning stage, the Spline model parameters μ_w and Σ_w belongs to the set of spline-classes Ω can be used for the purpose of recognition. To establish a mathematical setting, each control point in the test spline is mapped to its counterparts in the spline model. Hence, we use the Fréchet distance [36].

Fréchet distance evaluates the similarity between two spline curves. Let's denote μ_w is the spline mean and $\beta_t$data spline data curve. Fréchet distance is defined as the minimum cordlength sufficient to map two control points in both splines. The mapping continues until reaching the end of the spline curves. The distance is then the end of point traced forward along μ_w and one traveling forward along β_t, although the rate of travel for either point may not necessarily be uniform.

Spline Projection

Assume that the class-mean spline μ_w and the data spline curve β_t are projected into 2 dimension coordinate $(x_i, y_i), for\ i = 1 \ldots$ T space by estimating the y axes over the spline coefficients. the resulting vector of shape representation for the class-mean spline denoted by γ_t^{ω}

$$\gamma_t^{\omega} = (x_1, y_1, x_2, y_2, x_3, y_3, \ldots, x_T, y_T)^T \quad (32)$$

And test-data spline projection is denoted by α_t

$$\alpha_t = (x_1, y_1, x_2, y_2, x_3, y_3, \ldots, x_T, y_T)^T \quad (33)$$

Fréchet Matrix

The matrix of Fréchet distances is then computed for each pair of the control points of the class-mean spline γ_t^{ω} and the test-data spline α_t in

$$F = \begin{bmatrix} d_{(\gamma_{x_1,y_1},\alpha_{x_1,y_1})} & d_{(\gamma_{x_1,y_1},\alpha_{x_2,y_2})} & \cdots & d_{(\gamma_{x_1,y_1},\alpha_{x_T,y_T})} \\ d_{(\gamma_{x_2,y_2},\alpha_{x_1,y_1})} & d_{(\gamma_{x_{12},y_2},\alpha_{x_2,y_2})} & \cdots & d_{(\gamma_{x_2,y_2},\alpha_{x_T,y_T})} \\ & \vdots & & \\ d_{(\gamma_{x_T,y_T},\alpha_{x_1,y_1})} & d_{(\gamma_{x_T,y_T},\alpha_{x_2,y_2})} & \cdots & d_{(\gamma_{x_T,y_T},\alpha_{x_T,y_T})} \end{bmatrix} \quad (34)$$

where

$$d_{(\gamma_{x_i,y_i},\alpha_{x_i,y_i})} = \sqrt[2]{\left(\gamma_{x_i} - \alpha_{x_i}\right)^2 + \left(\gamma_{y_i} - \alpha_{y_i}\right)^2} \quad (35)$$

is the Euclidean distance between any pair of coordinates space. The matrix constructed is square non-symmetric matrix. We consider the diagonal cells of the matrix that explain the distance between control points pair for both objects under study, we choose the maximum value as a top limit for the final distance. The algorithm proceeds by iterating through the diagonal left to right, and for each column, calculates the distances and stores them if they are smaller than the reference maximum. The same process applies to rows.

$$F_{i,j} = max\ (d_{i,j}, min(F_{i-1,j}, F_{i,j-1}, F_{i-1,j-1})) \quad (36)$$

These values are compared with the other class-mean spline in the set Ω, the class-mean spline that has the minimum values is selected data spline is then grouped for that class.

5. Experiments

We have evaluated our approach with sets of isolated Arabic handwritten characters. Here, we have used 22 shape-classes for different writers, each with 40 training patterns. In total, we have tested the approach with 880 handwritten Arabic character shape patterns for testing and 1760 patterns for testing our approach. Figure 1 shows some training patterns used in this research paper. Figure 2 shows single shapes and their landmarks representation of being uniformly distributed along the skeleton of the shape under investigation.

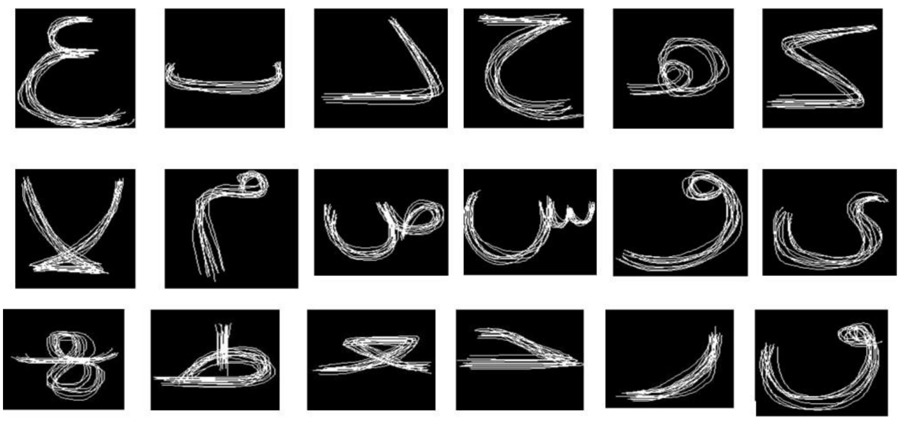


Figure 1: Sample handwritten character training sets.

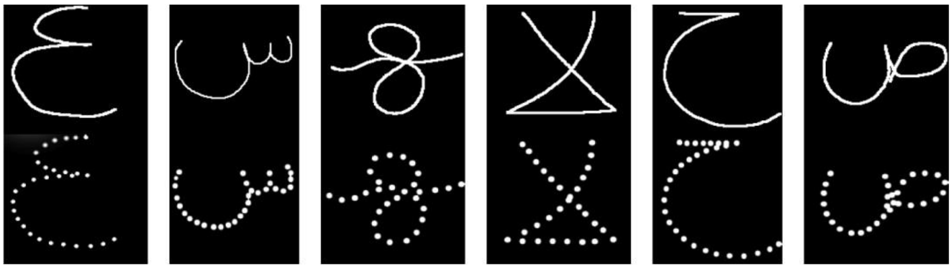


Figure 2: Training patterns and extracted landmarks.

In figure 3, sample cubic spline coefficients resulted from the training stage for spline-class ω. Here, each raw represents the set of coefficients for the handwritten character (Seen) and for each segment of the shape.

α_i	α_i	α_i	α_i
0.179	-1.0369	0.8579	144
0.1269	-1.3808	10.2539	48
0.179	-0.5	-0.679	144
0.1269	-1	7.8731	57
-0.8948	0.0369	-1.1421	143
0.3653	-0.6192	6.2539	64
0.4003	-2.6475	-3.7527	141
-1.588	0.4766	6.1114	70
1.2937	-1.4467	-7.847	135
0.9868	-4.2874	2.3006	75

α_i	α_i	α_i	α_i
-0.5751	2.4344	-6.8593	127
-2.3592	-1.327	-3.3138	74
0.0068	0.709	-3.7158	122
9.45	-8.4046	-13.045	67
-0.4521	0.7295	-2.2773	119
-8.4408	19.9454	-1.5046	55
-0.1983	-0.6269	-2.1748	117
1.3131	-5.3769	13.0638	65
0.2452	-1.2217	-4.0235	114
-0.8116	-1.4376	6.2493	74

α_i	α_i	α_i	α_i
0.2176	-0.4862	5.7314	109
-0.0665	-3.8726	0.9391	78
0.8846	0.1665	- 6.0511	103
2.0778	-4.0721	- 7.0056	75
-1.7559	2.8203	- 3.0643	98
-3.2445	2.1611	- 8.9166	66
2.1392	-2.4476	- 2.6916	96
9.9003	-7.5724	- 14.328	56
-1.8008	3.97	- 1.1692	93
-9.3566	22.1284	0.2282	44

α_i	α_i	α_i	α_i
0.064	-1.432	1.3684	94
2.5261	-5.941	16.415	57
0.5448	-1.24	-1.304	94
-1.748	1.6369	12.111	70
-0.243	0.3941	-2.151	92
1.4648	-3.606	10.141	82
-0.571	-0.336	-2.093	90
-0.112	0.7882	7.3234	90
0.5288	-2.05	-4.479	87
-0.018	0.4533	8.565	98

α_i	α_i	α_i	α_i
-0.544	-0.464	-6.993	81
-0.815	0.3985	9.4168	107
0.6464	-2.095	-9.551	73
0.2793	-2.047	7.768	116
-0.042	-0.156	-11.8	62
-0.302	-1.209	4.5113	122
0.5208	-0.281	-12.24	50
-0.071	-2.115	1.1867	125
-0.042	1.2811	-11.24	38
0.5874	-2.329	-3.258	124

α_i	α_i	α_{L-1} α_L	
-0.355	1.1567	- 8.802	28
-0.278	-0.567	- 6.155	119
0.4612	0.0919	- 7.553	20
0.5251	-1.402	- 8.124	112
-0.49	1.4755	- 5.986	13
-2.822	0.1738	- 9.351	103
1.498	0.0061	- 4.504	8
1.7645	-8.293	- 17.47	91
1.498	4.5	0.002	5
1.7645	-3	- 28.76	67

Figure 3: Spline coefficients for sample class-shape

In figure 4, illustrates the result from the training stage. First raw shows sample training sets, raw 2 shows the training stage Expectation Maximization initialization shape, raw 3, shows the projection of the spline training coefficients producing the landmarks, and raw 4 show the spline-mean for class shape representation. Figure 5 shows the convergence rate as a function per iteration number. The graph shows how associated distributed probabilities for the set of spline-classes Ω converged in a few iterations.

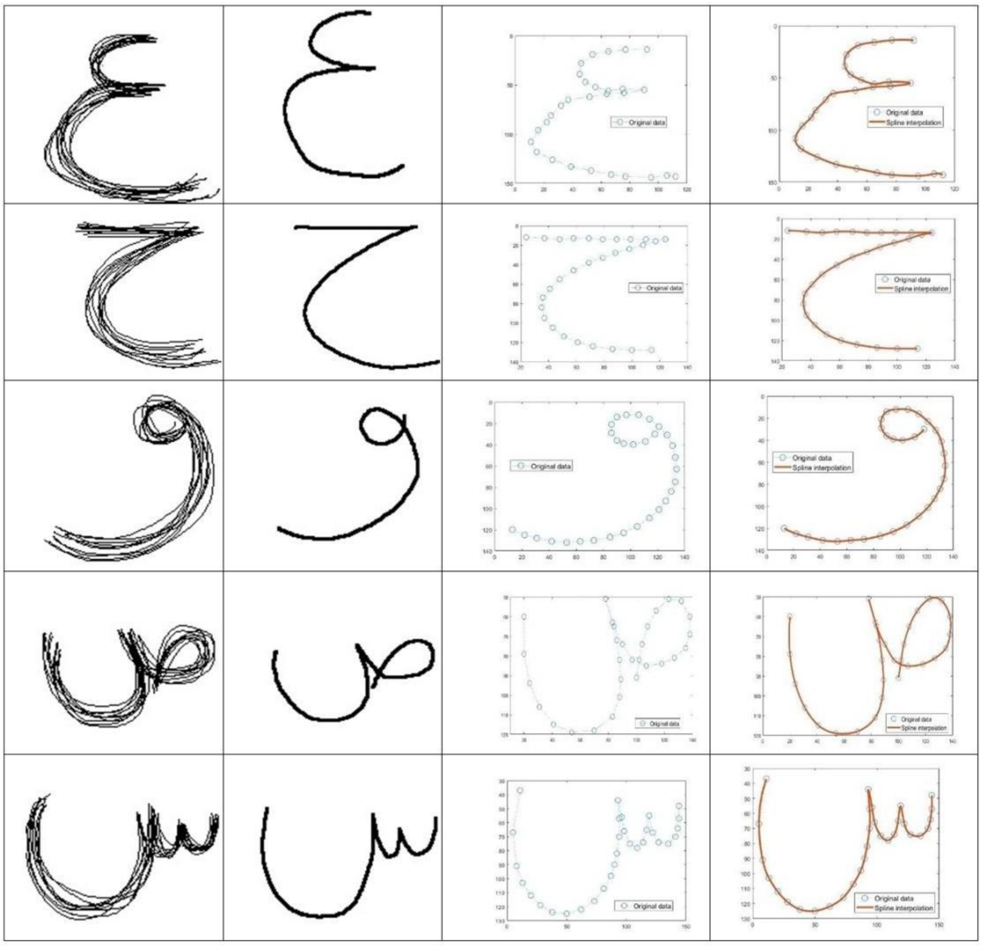


Figure 4: a) sample sets b) EM initialization c) Mean Shape landmarks d) Spline Mean Shape

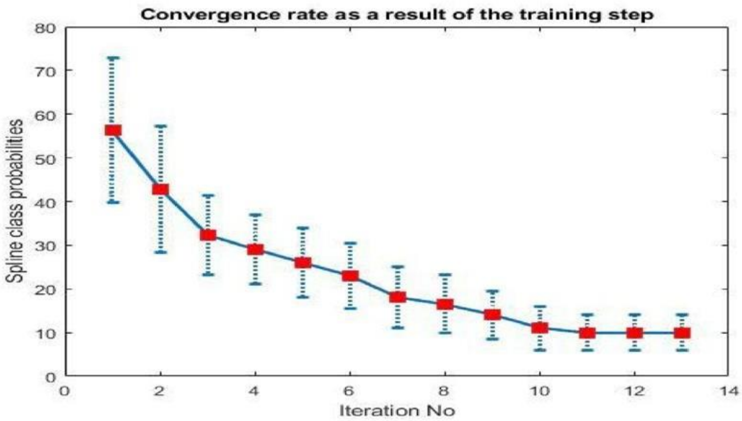


Figure 5: Convergence Rate as a function per iteration no.

Figure 6 shows the Fréchet distance table between the spline-mean shape and a testing shape pattern.

The graph shows the similarity between the spline curves which considers the location and ordering landmarks to the sample testing pattern landmarks. It is apparent from the table that the similarity is shown in the diagonal elements highlighted by green color. The diagonal distance between the landmarks for both the spline-mean shape and it's corresponding in the testing pattern. The similarity excluded by the minimum distance.

4.5	10.4	24.3	34.1	46.1	51.6	52.5	49.5	49.2	42.5	38.1	41.1	47.5	56.1	72.3	79.4	92.7	105	110	119
13.6	5.8	9.8	19.2	31.1	38.1	40.4	40	41.8	37.5	40.2	39.1	42	47.9	61.7	68.8	82.3	95	100	110
25.7	17.7	6.7	8.6	19	27	30.8	33.2	37.1	35.5	43.8	39.6	39.1	42	53	59.9	73.5	86	91.9	102
37.1	29.1	16.6	8.9	8.2	17.2	23.1	29.1	34.9	36.2	48.8	42.2	38.6	38.1	45.3	51.9	65.3	77.5	84.1	93.9
47.5	39.8	28.4	20.8	11	5.4	13	22.4	29.7	34	50.1	41.4	34.8	30.7	34.2	40.5	53.7	65.8	72.6	82.5
53.5	46.4	37	30.9	22	6.1	2.8	14.2	22	28.6	46.8	36.8	28	21.1	23.4	30	43.4	55.8	62.2	72
55.2	48.8	41.6	37.1	30.1	14.9	6.3	7.1	14.3	22.4	41.3	30.6	20.6	12.2	18.4	25.5	39.1	51.7	57.1	66.9
54	48.5	43.7	41	36.4	22.5	14.2	4	6.1	15	34	23	12.4	5	21	27.8	40.6	53.3	57.4	66.8
52.4	48.1	46.1	45.5	43.4	31.1	23.4	12	4.2	7.8	25.3	14.3	3.2	9.1	28.4	34.5	46.1	58.3	61.1	70
46.5	43.6	44.6	46.2	47	37.4	30.9	19.9	13	5	15.1	4.1	7.1	19.2	38.6	44.6	55.9	68	70.2	78.9
45	44.7	50.1	54.4	58.1	51	45.2	34.7	28.1	19.4	3	11.2	22	34.1	53.3	59	69.6	81.2	82.4	90.4
50	47.4	48.7	50.3	50.8	40.6	33.6	22.4	14.9	8.6	15.2	5.8	8.5	20	39.1	44.7	55.4	67.1	68.8	77.2
55.5	51.3	49.2	48.4	45.7	32.8	24.8	13.6	6.3	10.6	26.9	16.2	5.7	8.2	27.1	32.8	43.9	55.9	58.4	67.3
64.4	59.2	54.4	51.4	45.4	30.1	21.6	14.6	13.5	22	39.3	28.5	17.5	6.4	15	20.4	31.8	44.1	47.4	56.6
76.4	70.3	63.1	57.9	48.8	32.8	26	25.5	27.7	36.8	54.6	43.7	32.6	20.6	4	5.1	17.7	30.4	35.5	45.3
84.2	78.1	70.8	65.4	55.8	39.8	33.5	33.2	35	43.8	61	50.3	39.4	27.8	11.2	5	10	22.7	27.8	37.6
96.3	90.3	83	77.4	67.4	51.6	45.7	45.3	46.4	54.9	71.2	60.8	50.2	39.2	23.3	16.6	4.1	11.2	15.7	25.5
104	97.6	90.6	85	74.9	59.2	53.2	52.5	53.2	61.4	76.9	66.9	56.6	46	30.9	24.2	11.2	7.3	8.1	17.9
114	108	101	94.8	84.5	68.9	63.1	62.5	63	71.1	86.1	76.3	66.2	55.9	40.8	34	20.6	10	2.2	8.2
126	120	113	108	97.5	81.9	76.1	75	75	82.8	96.8	87.5	77.8	68	53.7	47	33.6	22.4	14.8	5

Figure 6: Fréchet distance

To take the investigation further, we demonstrate how the approach behaves under the presence of noise. In figure 7, we show how recognition rate is achieved when point position displacement error is applied. Test shape coordinates are being moved away from their original position. The figure shows the recognition rate fails to recognize shapes to their correct class ω in a few iterations and it fails completely when coordinates are moved away, yet, increasing variance significantly.

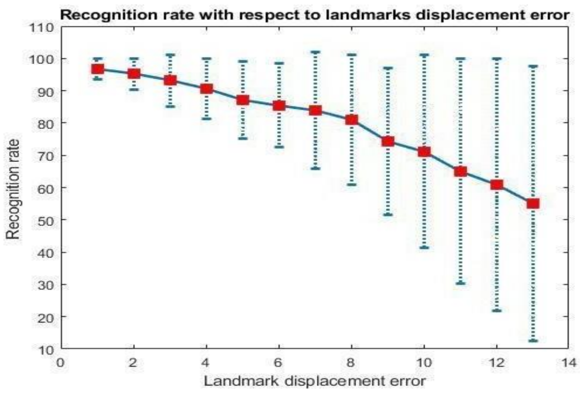
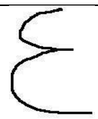
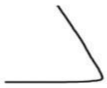
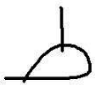
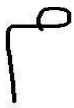
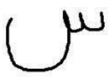
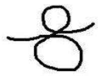
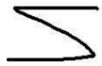
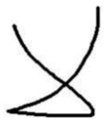
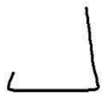


Figure 7: Recognition rate as a function per iteration no with point position error.

Table 1 shows recognition rates per spline-classes ω. Row 1 represents the shape name in Arabic alphabet. Row 2 represents the character shape. Row 3 lists the size of the testing patterns. Rows 4,5 show the correct and wrong registration of shapes to their spline-class ω.

Row 6 shows the recognition rate per spline-class ω. In total, we have achieved 96% recognition rate for such an approach.

Shape Name	Sample Shape	Test Size	Correct	False	Recognition Rate
Ain_1		80	78	2	97.5%
Baa		80	78	2	97.5%
Dal		80	79	1	98.8%
Faa		80	73	7	91.3%
Haa_1		80	79	1	98.8%
Ttah		80	76	4	95%
Hhah_1		80	71	9	88.8%
Ain_2		80	79	1	98.8%
Meem		80	76	4	95%
Seen		80	74	6	92.5%
Yaa		80	77	3	96.3%

Haa_2		80	78	2	97.5%
Waw		80	76	4	95%
Ain_2		80	79	1	98.8%
Hhah_2		80	74	6	92.5%
Kaf_1		80	79	1	98.8%
Lam_1		80	80	0	100%
Lam_2		80	73	7	91.3%
Raa		80	79	1	98.8%
Ssad		80	73	7	91.3%
Kaf_2		80	78	2	97.5%
Noon		80	79	1	98.8%
Total	22 classes	1760	1688	72	96%

6. CONCLUSION

In this research paper, we have shown how cubic spline 3rd degree polynomials can be used for recognizing handwritten Arabic characters. Each shape under testing is represented by a set of uniformly distributed landmarks along the skeleton. A cubic piecewise B-Spline is implemented on the shape to extract the coefficients needed for training set. A mixture of spline coefficients has been trained in a two-level hierarchy Expectation Maximization setting to capture the modes of spline parameters describing the pattern-sets. Recognition is then applied on a different set of testing patterns using the Fréchet distance mechanism. We have shown a remarkable recognition rate of 96%.

7. FUTURE WORK

In this research, there are some shortcomings to the approach. The first shortcoming is proving cubic b-spline 3rd order of polynomials is best representation for handwritten Arabic characters. The second is, the extracted parameters from the training stage have not been utilized for the purpose of recognition stage encapsulated within a statistical framework: for example, spline coefficients are not utilized for spline alignment. Secondly, there is a need to provide a hierarchical framework for spline alignment.

REFERENCES

[1] J. Wood, "Invariant pattern recognition: A review," Pattern Recognition, vol. 29, no. 1, 1996, 1-17.

[2] Anil, K. Jain,Robert P.W. Duin, and Jianchang Mao: “Statistical Pattern Recognition: A Review”, IEEE Pattern Analysis and Machine Intelligence, vol 22, No. 1, PP 4-37, 2000.

[3] P.F. Baldi and K. Hornik, "Learning in linear neural networks: A survey," IEEE Transactions on Neural Networks, vol. 6, no. 4, 1995, 837-858.

[4] T.Y. Kong and A. Rosenfeld, "Digital topology: introduction and survey," Computer Vision, Graphics, and Image Processing, vol. 48, no. 3, pp. 357-393, 1989.

[5] T.R. Reed and J.M.H. Dubuf, "A review of recent texture segmentation and feature extraction techniques," CVGIP - Image Understanding, vol. 57, no. 3, pp. 359-372, 1993.

[6] S. Sarkar and K.L. Boyer, "Perceptual organization in computer vision - a review and a proposal for a classificatory structure", IEEE Transactions on Systems Man and Cybernetics, vol. 23, no. 2, 1993, 382-399.

[7] Lorigo L. M., Govindaraju V., “Offline Arabic Handwriting Recognition: A Survey”, IEEE Trans Pattern Anal Mach Intelligence, Vol: 28(5): PP 712-24, 2006.

[8] Ishani Patel, Virag Jagtap, Ompriya Kale, “A Survey on Feature Extraction Methods for Handwritten Digits Recognition”, International Journal of Computer Applications, Vol 107, No 12, PP 1117, 2014.

[9] Muhammad Sharif, Farah Naz, Mussarat Yasmin, Muhammad Alyas Shahid, Amjad Rehman, “Face Recognition: A Survey”, Journal of Engineering Science and Technology Review, Vol 10, No 2, PP 166-177, 2017.

[10] Rafiqul Zaman Khan, Noor Adnan Ibraheem, “HAND GESTURE RECOGNITION: A LITERATURE REVIEW “,International Journal of Artificial Intelligence & Applications, Vol.3, No.4, pp 161-174, 2012.

[11] Bruce George Lindsay,Mary L. Lesperance, “A review of semiparametric mixture models”, Journal of Statistical Planning and Inference, Vol: 47 , No: 1 PP 29-39 , 1995.

[12] Patel, Shital P. Thakkar, "Handwritten Character Recognition in English: A Survey", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 2, 2015.

[13] Edyta LUKASIK, Malgorzata CHARYTANOWICZ, Marek MILOSZ, Michail TOKOVAROV, Monika KACZOROWSKA, Dariusz CZERWINSKI, Tomasz ZIENTARSKI, "Recognition of handwritten Latin characters with diacritics using CNN", ARTIFICIAL AND COMPUTATIONAL INTELLIGENCE, Vol. 69. No 1, 2021.

[14] Dai Ruwei,Liu Chenglin,Xiao Baihua, "Chinese character recognition: history, status and prospects", Frontiers of Computer Science in Chinavolume 1, pages126–136 (2007).

[15] Tabita L. Tobing, Sule Y. Yayilgan, Sony George, " Isolated Handwritten Character Recognition of Ancient Hebrew Manuscripts ", SOCIETY FOR IMAGING SCIENCE AND TECHNOLOGY, PP 35-39, 2022.

[16] Madhuri Yadav,Ravindra Kumar, Purwar,Mamta Mittal, "Handwritten Hindi character recognition: a review", IET Image Processing, Volume12,Issue11, November 2018, Pages 1919-1933.

[17] Pervez Ahmed and Yousef Al-Ohali, "Arabic Character Recognition: Progress and Challenges", J. King Saud Univ., Vol. 12, Comp. & Info. Sci., pp. 85-116,2000.

[18] Christopher M. Bishop, “Neural Networks for Pattern Recognition”, Clarendon Press, 1995, ISSN 0198538642.

[19] A. Dempster, N. Laird, D. Rubin, "Maximum likelihood from incomplete data via the em algorithm", J. Roy. Statist. Soc. Ser. 39 (1977) 1–38

[20] J. Brendan, N. Jojic, Estimating mixture models of images and inferring spatial transformations using the em algorithm, IEEE Computer Vision Pattern Recognition 2 (1999) 416–422.

[21] C. Bishop, J. Winn, Non-linear Bayesian image modelling, Proceedings of Sixth European

[22] N. Vasconcelos, A. Lippman, A probabilistic architecture for content-based image retrieval, Proceedings of International Conference on Computer Vision and Pattern Recognition, 2000, pp. 216–221.

[23] B. North, A. Blake, Using expectation-maximization to learn dynamical models from visual data, Image Vision Computing. 17 (8) (1999) 611–616.

[24] M. Revow, C. Williams, G.E. Hinton, Using generative models for handwritten digit recognition, IEEE Trans. Pattern Anal. Mach. Intell. 20 (2) (1996) 592–606

[25] M. Svoboda,L. Matiu-Iovan,P. Andea, B-spline interpolation technique for digital signal processing , International Conference on Information and Digital Technologies, 2015

[26] A.M. Aguilera, M.C. Aguilera-Morillo, Comparative study of different B-spline approaches for functional data. Mathematical and Computer Modelling 58 (2013) 1568–1579.

[27] E. Dimas, D. Briassoulis, 3D geometric modeling based on NURBS: a review, Journal of Advances in Engineering Software, (1999)741-751.

[28] A. Galvez, A. Iglesias, A new iterative mutually coupled hybrid GA-PSO approach for curve fitting in manufacturing, Journal of Applied Soft Computing, (2012),

[29] I. Zeid, Mastering CAD/CAM, McGraw Hill International Edition, New York, 2005.

[30] M. Sevaux, Y. Mineur, A curve-fitting genetic algorithm for a styling application, European Journal of Operational Research, 179 (2007) 895-905.

[31] R. Liutkevicius, A. Davidsonas, Surface Reconstruction from Partially Structured Noisy Cloud of Points Using B-Splines, Journal of Information Technology and Control, 42 (2013).

[32] Muhammed Safraz, Mutaza Khan, "Automatic outline capture of Arabic fonts", Journal of information Sciences, Vol 140, No. 3-4, PP 269-281,2002

[33] Muhammad Safraz, "Optimal Curve Fitting to Digital Data", Journal of WSCG, Vol 11, No. 1-3, 2003.

[34] AlShaher Abdullah, Hancock Edwin, Learning mixtures of Point Distribution models with the EM algorithm. Pattern Recognition, 36(2003) 2805-2818.

[35] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer Science, and Business Media, 2006.

[36] Thomas Heiter and Heikki Manila. Computing Discrete Fréchet Distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, Technical University Vienna, Austria, 1994.

AUTHORS

ABDULLAH A. AL-SHAHER is currently enrolled as associate professor faculty member in the Computer Science and Information Systems Department at the college of Business Studies in the Public Authority for Applied Education and Training, Kuwait. He received his Ph.D. in Computer Science from the University of York, York, United Kingdom in 2005. In 1995 he was awarded master's degree (honour list) in Computer Science from University of Denver, Denver, Colorado, USA, and a bachelor's degree from HustonTillotson University in 1988 Austin, Texas, USA. His main interest is Shape Recognition, Statistical Patten Recognition, Computer Vision, and Artificial Intelligence.

YOUSEF S. ALKHAWARI is currently enrolled as a lecturer faculty member in the Computer Science and Information Systems Department at the College of Business Studies in the Public Authority for Applied Education and Training (PAAET), Kuwait. He received his master's degree in computer science from California State University of Sacramento (CSUS), Sacramento, California, USA in 1995. In 1991 he earned his bachelor's degree in computer engineering from University of the Pacific (UOP), Stockton, California, USA. His main interest is Knowledge Based Systems and Artificial Intelligence and machine learning.