

A Novel Classification Method with Cubic Spline Interpolation

Husam Ali Abdulmohsin^{1,*}, Hala Bahjat Abdul Wahab² and Abdul Mohssen Jaber Abdul Hossen³

¹Computer Science Department, Faculty of Science, University of Baghdad, Baghdad, Iraq

²Faculty of Computer Science, Technology University, Baghdad, Iraq

³Computer Science Department, Al-Turath University College, Baghdad, Iraq

*Corresponding Author: Husam Ali Abdulmohsin. Email: husam.a@sc.uobaghdad.edu.iq

Received: 23 February 2021; Accepted: 15 May 2021

Abstract: Classification is the last, and usually the most time-consuming step in recognition. Most recently proposed classification algorithms have adopted machine learning (ML) as the main classification approach, regardless of time consumption. This study proposes a statistical feature classification cubic spline interpolation (FC-CSI) algorithm to classify emotions in speech using a curve fitting technique. FC-CSI is utilized in a speech emotion recognition system (SERS). The idea is to sketch the cubic spline interpolation (CSI) for each audio file in a dataset and the mean cubic spline interpolations (MCSIs) representing each emotion in the dataset. CSI interpolation is generated by connecting the features extracted from each file in the feature extraction phase. The MCSI is generated by connecting the mean features of 70% of the files of each emotion in the dataset. Points on the CSI are considered the new generated features. To classify each audio file according to emotion, the Euclidian distance (ED) is found between each CSI and all MCSIs of all emotions in the dataset. Each audio file is classified according to the nearest MCSI to the CSI representing it. The three datasets used in this work are Ryerson Audio-Visual Database of Emotional Speech and Song (RAVD ESS), Berlin (Emo-DB), and Surrey Audio-Visual Expressed Emotion (SAVEE). The proposed work shows fast classification and high accuracy of results. The classification accuracy, i.e., the proportion of samples assigned to the correct class, using FC-CSI without feature selection (FS), was 69.08%, 92.52%, and 89.1% with RAVDESS, Emo-DB, and SAVEE, respectively. The results of the proposed method were compared to those of a designed neural network called SER-NN. Comparisons were made with and without FS. FC-CSI outperformed SER-NN on Emo-DB and SAVEE, and underperformed on RAVDESS, without using an FS algorithm. It was noticed from experiments that FC-CSI operated faster than the same system utilizing SER-NN.

Keywords: Classification methods; machine learning; cubic spline interpolation; Euclidean distance; emotional speech dataset; RAVDESS; Emo-DB; SAVEE



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Numeric data are often difficult to analyze, and functions to link the data are hard to find. In 1998, cubic spline interpolation (CSI) was proposed, which connects pairs of data points using unique cubic polynomials, generating a continuous and smooth curve [1]. A spline curve is described by a sequence of polynomials called spline curve segments, and a spline surface is a mosaic of surface patches [2,3].

Interpolation has many applications, all with the purpose of smoothing. The fundamental concept of CSI is based on the engineer's tool used to draw smooth curves through several points. The spline consists of weights attached to a flat surface. A flexible strip is bent across each of these weights, resulting in a pleasingly smooth curve. The mathematical spline is similar in principle. The points, in this case, are numeric data. The weights are the coefficients of cubic polynomials used to interpolate the data. These coefficients "bend" a line so that it passes through each data point without erratic behavior or breaks in continuity [1]. CSI is used to determine rates of change and cumulative change over an interval. CSI was applied to compute the heat transfer across the thermocline depth of three lakes in the study area of Auchi in Edo State, Nigeria [4]. Spline interpolation methods include linear, quadratic, cubic Hermite, and cubic [5], with different applications. A planning path method based on cubic spline interpolation was proposed to smooth a robot's moving path [6]. Biodiesel production from waste cooking oil was optimized using CSI and response surface methodology in a mathematical model [7].

We propose an algorithm with third-order CSI curves to classify emotions in three datasets. We are aware of no previous use of CSI in classification, although some research has adopted the idea of this work to discover the wave behavior of an audio signal to classify emotions, using a convolutional neural network (CNN) and not CSI. Related work discovered signal behavior directly from raw data, while FC-CSI works with features extracted from raw data. A network based on time distribution CNN, CNN, and RNN was proposed, without traditional feature extraction to classify emotions, achieving 88.01% classification accuracy on the seven emotions of Emo-DB [8]. A time continuous end-to-end SER prediction system to classify valence and arousal emotions in the RECOLA dataset was proposed [9]. A combination of CNN and LSTM networks learned the representation of a speech signal from raw data. A CNN was proposed to learn and classify emotional features from the spectrogram representation of a speech signal [10], merging feature extraction and classification, with respective classification accuracies of 79.5% and 81.75% on the RAVDESS and IEMOCAP datasets.

CSI has been used in many applications other than classification. Third-order polynomial CSI was used to fit the stress-strain curve of a standardized specimen of SAE 1020 steel hot-rolled flat [11]. Market power points were calculated in different operating conditions and CSI was used to interpolate between them to suggest an appropriate operating condition for a given level of market power [12]. Damage to buildings after a seismic event was assessed by setting boundary conditions based on data from accelerometers at the base and roof and estimating the influence on the other floors [13]. CSI was used in an empirical mode decomposition method to analyze nonstationary and nonlinear signals in communication [14]. Based on its local characteristics in the time domain, the signal was decomposed to a series of complete orthogonal intrinsic mode functions. CSI was used to connect the minimum and maximum signal values to lower and upper envelopes, respectively, and calculate their means. CSI was used to visualize data generated by millions of trajectory frames of molecular dynamic simulations by speeding up the calculation of the atomic density (volumetric map) with a 3D grid from molecular dynamic trajectory data [15].

We propose a classification method utilizing CSI. Classification methods are used in applications such as speech emotion recognition. Regardless of the high accuracy performance achieved by classification algorithms adopting ML methods, their training time is high. We aim to minimize this and propose a classification algorithm that uses cubic spline interpolation, which is a new field in classification research.

The limitation of FC-CSI is the conversion of features from 1D to 2D. To draw a curve on a 2D plane, features must be represented in 2D coordinates, i.e., X and Y. In the proposed algorithm, the feature value is considered the Y-axis, and the sequence of the feature is the X-axis. The distance between features on the X-axis is determined by trial and error, which affects the accuracy, and to find the perfect distance affects system time consumption.

The remainder of this paper is organized as follows. Section 2 explains the proposed algorithm, Section 3 discusses experimental results, and Section 4 relates our conclusions and proposes future work.

2 Proposed Method

We design and implement a classification method. Fig. 1 shows its block diagram. Below, we discuss the blocks in Fig. 1 and explain the data flow of the proposed work.

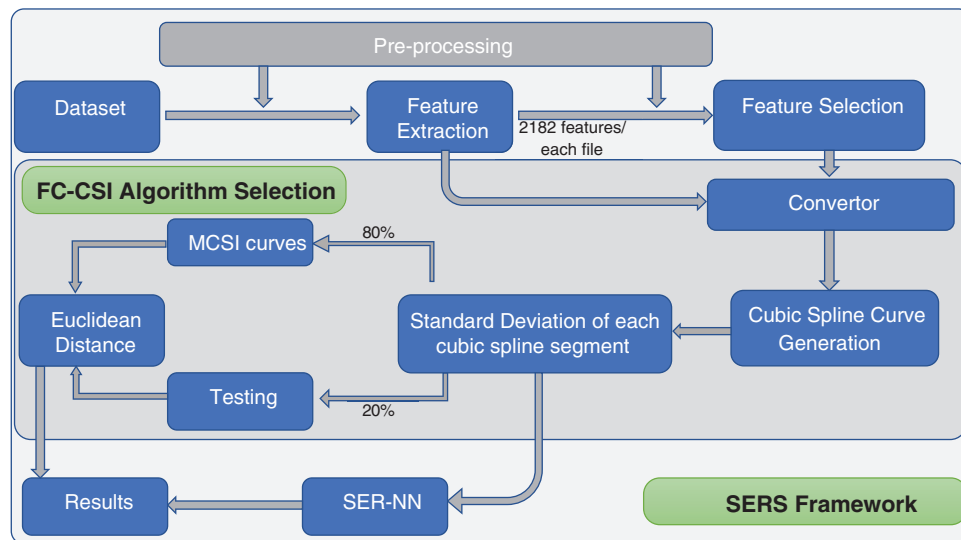


Figure 1: Block diagram of proposed SERS system

2.1 Hardware and Software Platform

All experiments were performed using MATLAB R2019a on a computer with an Intel Core I7-8750H CPU @2.2 GHz with 32 GB of RAM and a Windows 10 operating system.

2.2 Datasets

The RAVDESS [16], Emo-DB [17], and SAVEE [18] datasets were used in this work, and were chosen according to the following criteria:

- Recorded in three frequencies;
- Representing the main emotions according to Paul Ekman's definition [19];
- Two languages were used in recording them;
- Gender balance was required to be available.

The specifications and properties of the datasets are shown in Tab. 1.

Table 1: Datasets used in this work

	RAVDESS	Emo-DB	SAVEE
Actors	24 (12 Females, 12 Males)	10 (5 Females, 5 Males)	4 (Males only)
Language	English (North American)	Germany	English (British)
Texts used to record the dataset	2	10	15
Number of emotions	8 (6 main emotions)	7 (5 main emotions)	7 (6 main emotions)
Emotions represented (Number of samples by emotion)	Fear (192), Disgust (192), Happy (192), Neutral (96), Sad (192), Anger (192), Surprise (192), Calm (192)	Fear (69), Disgust (46), Happy (71), Neutral (79), Sad (62), Anger (127), Bored (81)	Fear (60), Disgust (60), Happy (60), Neutral (120), Sad (60), Anger (60), Surprise (60)
Number of features	2182	2182	2182
Number of samples	1440	535	480
Speech signal frequency	48000 Hz	16000 Hz	44100 Hz
Sample size	16 bits	16 bits	16 bits
Average human classification accuracy	72%	84%	66.5 ± 2.5 %
Bit rate	768 kbps	256 kbps	705 kbps

2.3 Experimental Setup

Each audio file in a dataset was converted to a one-dimensional vector. Three preprocessing functions were applied to each vector:

1. All files were grouped according to emotion;
2. Silent parts at the beginning and end (and not in the middle) of each audio file were removed. This was the first function applied to the data.
3. Data in each audio vector were normalized between 0 and 100:

$$x_i = \frac{(x_i - Min)}{(Max - Min)} * 100, \quad (1)$$

where x is the data read from the audio file; i is the number of values representing the audio file; and Min and Max are the minimum and maximum values, respectively, found in the file. The factor of 100 was discovered through trial and error. This step was applied after feature extraction.

One way to evaluate the performance of the classification method was to compare its results with those of a predefined neural network (NN). We used one hidden layer with 10 nodes. The audio files in each of the

three datasets were randomly divided into subsets of 70% for training, 15% for validation, and 15% for testing, and no files were in multiple subsets. The training function used was “trainscg,” the divide function was “dividerand,” and the divide mode was “sample” [20].

2.4 Feature Extraction

The number of features extracted from each audio file was 2182. Fifteen feature types were extracted from each audio file: entropy, zero crossing (ZC), deviation of ZC, energy, deviation of energy, harmonic ratio, Fourier function, Haar, MATLAB fitness function, pitch function, loudness function, Gammatone cepstral coefficient (GTCC) according to time and frequency, and MFCC function according to time and frequency. The deviations (SDs) of these 15 features were calculated using 10 degrees on either side of the mean (including 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, and 2.5). The feature extraction function will be called the feature extraction mean deviation (FE-MD) function.

2.5 Converter

This step receives its input from feature extraction and sends its output to cubic spline curve generation, as shown in Fig. 1. It is known that each feature is represented by one numeric value. In order to find the CSI between all features, each feature has to be converted to a two-dimensional coordinate (X and Y). Hence, we assumed the value of the feature to be the Y-axis, and assumed the feature sequence index in the feature vector of each audio file to represent the X-axis of that specific feature. Many experiments were tested, choosing the best spacing between each two features on the X-axis, (i.e., the length of the cubic spline segment), and the best segment length found was 30, which means the first three features will have values of zero, 30, and 60, respectively, on the X-axis, and so on. These are called the control points, or knots, in interpolation [21].

2.6 Cubic Spline Interpolation Generation

Numeric data are commonly difficult to analyze, and a function to successfully link data is hard to find.

In 1998, mathematicians came up with CSI [1]. Fig. 2 shows an example of a single polynomial between control points (x_{i-1}, y_{i-1}) and (x_i, y_i) . A spline curve is composed of a sequence of polynomials called spline curve segments, and a spline surface is a mosaic of surface patches [2,3].

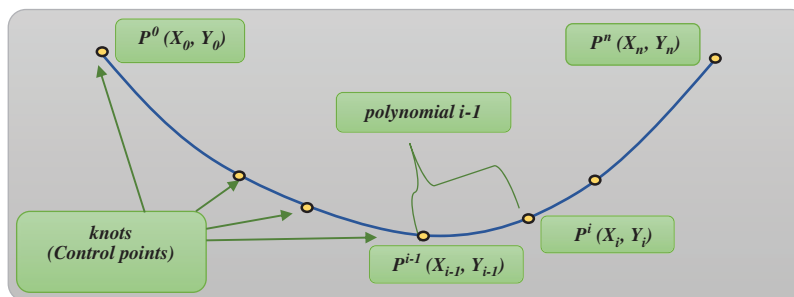


Figure 2: Interpolation with cubic splines between seven points, using flexible rulers bent to follow predefined points (“control points”) in yellow

Interpolation algorithms find sets of unique polynomials that form the main spline curve to interpolate all the control points and generate new secured points between them. Changes in control point positions can affect the curvature of the generated polynomials. We chose spline interpolation for three reasons: 1) regardless of polynomial degrees, interpolation error can be small; 2) it avoids Runge’s phenomenon

[22,23] because when high-degree polynomials are used, oscillation can occur between points; 3) the generated polynomials serve as a technique to sketch smooth secured curves [24].

Interpolation has many applications, and all follow the concept of smoothening [1,21]. We first explain the concept of the degree of a polynomial. A polynomial is a mathematical expression that can be formed from constants and symbols (like x and y), which are also called variables or indeterminants. The constants and variables are connected by means of multiplication, addition, and exponentiation to positive integer powers. A polynomial of a variable (x) can be written as

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x^1 + a_0 \quad (2)$$

where a_0, a_1, \dots, a_n are constants, and x is the variable (indeterminant). The word “indeterminant” indicates that x represents no specific value, and therefore can take any value. A function that maps the result of substitution with a value is called a polynomial function. Eq. (2) can be written as

$$\sum_{k=0}^n a_k x^k \quad (3)$$

Therefore, a polynomial can be written as either zero or as a sum of a limited number of nonzero terms, each the product of a number (coefficient) and a finite number of variables raised to a positive integer power. The power is called the degree, and the degree of a polynomial is the largest degree among all terms with nonzero coefficients. An example of a cubic polynomial is

$$-5x^3 + 4x^2 + 1x + 7 \quad (4)$$

The degree of the second term is 2, and the coefficient of the term is +4. The degree of the polynomial is the degree of the highest term, i.e., three.

A third-degree spline is called a cubic spline. Let us assume a cubic spline in the interval $x_0 \leq x_i \leq x_n$ is generated by a set of piecewise polynomials $s_i(x)$. The standard form of the cubic spline is

$$R_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad (5)$$

where $i = 1, 2, \dots, n-1$; and n is the number of control points. Hence, $n-1$ is the number of cubic polynomials that will form the cubic spline interpolation. The first and second derivatives of these $n-1$ equations are fundamental in cubic spline interpolation methodology, and these are respectively shown as [25]

$$R_i'(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2 \quad (6)$$

$$R_i''(x) = 2c_i + 6d_i(x - x_i). \quad (7)$$

From these functions, three matrices can be generated (see Fig. 3), and we can write

$$Ax = b. \quad (8)$$

Eq. (5) can be solved to obtain coefficients $a, b, c,$ and d of the cubic spline polynomial [26–29].

2.7 Standard Deviations of Cubic Spline Segments

This step generates CSI with equal lengths, so 10 deviation degrees (0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2 (standard deviation), 2.25, and 2.5) will be found for all points in each polynomial to construct the final spline curve. After this step, CSIs representing all audio files in the dataset will be randomly divided into two parts, 70% to find the MCSI curves and 30% for testing.

$$A = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \dots & h_{n-1} \\ 0 & h_1 & 2(h_1 + h_2) & \dots & h_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix}$$

$$x = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

Figure 3: Matrices used to find the four coefficients (a, b, c, and d) of CSI

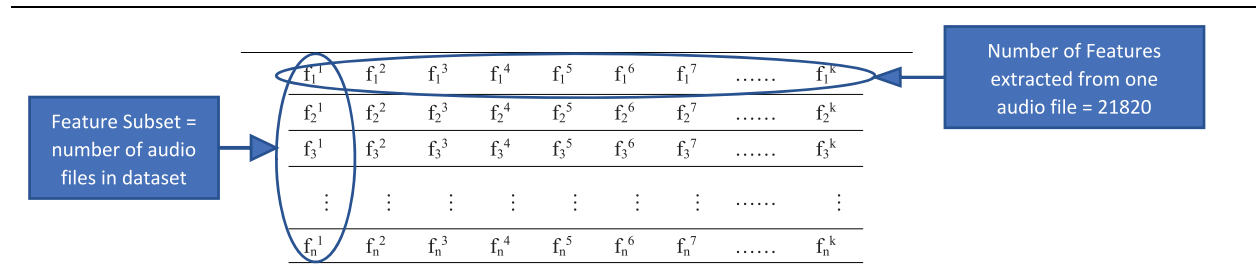
2.8 Finding Mean Cubic Spline Interpolation Curves

Each MCSI curve represents a different emotion, so the number of MCSI curves is the number of emotions represented in the dataset, i.e., 8, 7, and 7 for RAVDESS, EMO-DB, and SAVEE, respectively. MCSI is calculated as

$$MCSI(m) = \left(\begin{matrix} n \\ m = 1 \end{matrix} \right) \left(\begin{matrix} x = 21820 \\ j = 1 \end{matrix} \right) \left(\sum_{i=1}^{y(m)} CSI(i,j)/\text{length}(y(m)) \right), \tag{9}$$

where $m = 1, 2, \dots, n$; n is the number of emotions in the dataset; $j = 1, 2, \dots, x$, where x is the number of points forming each CSI representing each audio file; $i \in y(m)$, where vector $y(m)$ stores 70% of the audio files indices randomly selected from each emotion. The feature subset is $(f_{ij}, f_{i+1j}, f_{i+2j}, \dots, f_{nj})$, where $\text{length}(y(m))$ equals 1152, 428, and 384, for RAVDESS, EMO-DB, and SAVEE, respectively. At the end of this step, m MCSIs will be generated, where m equals the number of emotions. Each MCSI represents a different emotion though a unique curve. The summation will add the same 70% sample from each feature subset, as defined in [Tab. 2](#). The MATLAB code of MCSI generation is shown in [Fig. 4](#).

Table 2: Structure of features extracted from CSI



2.9 Testing

The audio files in the dataset will be divided into two groups, training and testing. The 70% for training will be used to find the MCSIs, and the remaining 30% of the audio files in the dataset, will be used to test the classification accuracy of the proposed work. This 30% is 432, 160, and 144 audio files for RAVDESS, EMO-DB, and SAVEE, respectively.

```

Function MCSI_Generation (CSI_curves)
% Note, n = number of emotions = 8 for RAVDESS, m1 = 7 for EMO-DB, m1 = 7 for SAVEE.
% Note, y = vector of length m, each cell stores 70% of the indexes of the random selected samples of the
corresponding emotion.
For m = 1 : n
    For j = 1 : 21820
        MCSI(m) = ( n / (m=1) ) ( x=21820 ) ( Σ_{i=1}^{y(m)} CSI (j, i) ) / length(y(m))
    End (j)
End (m)

```

Figure 4: MATLAB code for MCSI generation

2.10 Euclidean Distance

The standard Euclidean distance function is used in this work [30]:

$$ED_{nm} = \sum_{i=j=1}^{n=21820} \sqrt{((X_{jm} - X_{in})^2 + (Y_{jm} - Y_{in})^2)}, \quad (10)$$

where (X_{in}, Y_{in}) are the coordinates of the CSI samples determined for testing, and (X_{jm}, Y_{jm}) are the coordinates of the MCSI curve that represents one of the emotions in the dataset. The matrix that will be generated from Eq. (10) and the MATLAB code in Fig. 5 will be an $n \times m$ matrix, where n is the number of testing audio samples and m is the number of emotions in the dataset. Each audio sample will be classified according to the closest MCSI to the CSI that represents it.

```

Function Euclidean_Distance (MCSI_curves, CSI_curves)
% Note, m1 = number of emotions = 8 for RAVDESS, m1 = 7 for EMO-DB, m1 = 7 for SAVEE
% Note, n1 = 30% total samples = testing samples = 432 for RAVDESS, n1 = 160 for EMO-DB, n1 = 144 for
SAVEE
For m = 1 : m1
    For n = 1 : n1
        For i = j = 1 : 21820
            ED (n, m) = ED (n, m) + √((X_{jm} - X_{in})^2 + (Y_{jm} - Y_{in})^2)
        End (i)
    End (n)
End (m)

```

Figure 5: MATLAB code for Euclidean distance function

3 Experimental Results

We analyze the performance of SERS. The best classification accuracy results will be discussed with respect to the fewest features selected. Experiment 1.1 (Exp1.1) and experiment 1.2 (Exp1.2) were respectively performed before and after deploying the pre-designed feature selection t-test fitness (FS-TF) algorithm.

3.1 Experiment 1.1: SERS Performance Analysis Before Deploying Feature Selection Algorithm

Through Exp1.1, the classification performance of SERS is calculated in two different approaches. First, utilizing the FC-CSI algorithm and second through utilizing the SER-NN. Both approaches are implemented without deploying the FS-TF algorithm.

3.1.1 SERS Accuracy Performance

Figs. 6–8 show the classification accuracy results of Exp1.1 after running the FC-CSI classification algorithm. Figs. 9–11 show the classification accuracy results of Exp1.1 after running the SER-NN classification algorithm. Tab. 3 shows the results of Exp1.1 implemented on the 2182 FE-MD features extracted from the RAVDESS, EMO-DB, and SAVEE datasets, without feature selection.

	Neutral	Calm	Happy	Sad	Angry	Fear	Disgust	Surprise	
15	3	2	4	0	1	1	3	51.72%	
7.69%	10.34%	6.89%	13.79%	0%	3.44%	3.44%	10.34%	48.28%	
1	32	0	7	0	3	2	0	71.11%	
2.22%	16.41%	0%	5.55%	0%	6.66%	4.44%	0%	28.89%	
0	0	26	4	0	1	2	1	76.47%	
0%	0%	13.33%	1.76%	0%	2.94%	5.88%	2.94%	23.53%	
0	2	1	17	0	4	0	2	65.38%	
0%	7.69%	3.84%	8.71%	0%	15.38%	0%	7.69%	34.62%	
0	0	2	2	31	2	1	3	75.6%	
0%	0%	4.87%	4.87%	15.89%	4.87%	2.43%	7.31%	24.4%	
0	0	3	2	2	18	1	2	64.28%	
0%	0%	10.71%	7.14%	7.14%	9.23%	3.57%	7.14%	35.72%	
1	0	4	2	4	3	30	1	66.66%	
2.22%	0%	8.88%	4.44%	8.88%	6.66%	15.38%	2.22%	33.34%	
2	1	0	0	1	6	1	26	70.27%	
5.4%	2.7%	0%	0%	2.7%	16.21%	2.7%	13.33%	29.73%	
78.94%	84.21%	68.42%	44.73%	81.57%	47.36%	78.94%	68.42%	69.07%	
21.06%	15.79%	31.58%	55.27%	18.43%	56.64%	21.06%	31.58%	8.21%	

Figure 6: Exp1.1, SERS, FC-CSI, confusion matrix with RAVDESS, without FS

	Fear	Disgust	Happy	Bored	Neutral	Sad	Angry	
13	0	2	0	0	1	0	81.25%	
13.54%	0%	13.79%	0%	0%	3.44%	0%	18.75%	
0	9	0	1	0	0	0	90%	
0%	9.37%	0%	10%	0%	0%	0%	10%	
0	0	11	0	0	0	2	84.25%	
0%	0%	11.45%	0%	0%	0%	15.38%	15.75%	
0	0	0	15	0	0	0	100%	
0%	0%	0%	15.62%	0%	0%	0%	0%	
0	0	0	0	15	1	0	93.75%	
0%	0%	0%	0%	15.62%	6.25%	0%	6.25%	
0	0	0	0	0	10	0	100%	
0%	0%	0%	0%	0%	10.41%	0%	0%	
0	0	1	0	0	0	23	95.83%	
0%	0%	4.16%	0%	0%	0%	23.95%	29.73%	
100%	100%	78.57%	93.75%	100%	83.33%	92%	92.52%	
0%	0%	21.43%	6.25%	0%	16.67%	8%	7.48%	

Figure 7: Exp1.1, SERS, FC-CSI, confusion matrix with Emo-DB, without FS

We observe the following after analyzing the results:

- Results from Exp1.1 show that FC-CSI outperformed SER-NN FC on Emo-DB and SAVEE and underperformed on RAVDESS. This leads to the conclusion that, FC-CSI does not perform well on datasets of low human-accuracy performance.
- The most misclassified samples are those of the sad and fear emotions in RAVDESS, and neutral, sad, and disgust in SAVEE, which shows the drawback of FC-CSI in recognizing low-amplitude emotions.

	Angry	disgust	Fear	Happy	Sad	Surprise	Neutral	
9	0	0	0	1	0	0	90%	
11.5%	0%	0%	0%	10%	0%	0%	10%	
1	11	1	0	0	0	3	68.75%	
6.25%	14.1%	6.25%	0%	0%	0%	18.75%	31.25%	
1	0	10	0	0	0	0	90.9%	
9.09%	0%	12.82%	0%	0%	0%	0%	9.1%	
0	0	0	10	0	0	0	100%	
0%	0%	0%	12.82%	0%	0%	0%	0%	
0	0	0	0	10	1	2	76.92%	
0%	0%	0%	0%	12.82%	7.69%	15.38%	23.08%	
0	0	0	1	0	10	0	90.9%	
0%	0%	0%	9.09%	0%	12.82%	0%	9.1%	
0	0	0	0	0	0	18	100%	
0%	0%	0%	0%	0%	0%	23.07%	0%	
81.81%	100%	90.9%	90.9%	90.9%	90.9%	78.26%	89.10%	
18.19%	0%	9.1%	9.1%	9.1%	9.1%	21.74%	10.9%	

Figure 8: Exp1.1, SERS, FC-CSI, confusion matrix with SAVEE, without FS

	Neutral	Calm	Happy	Sad	Angry	Fear	Disgust	Surprise	
1	10	1	0	0	0	0	2	1	71.4%
4.6%	0.5%	0.0%	0.0%	0.0%	0.0%	0.9%	0.5%	28.6%	
3	25	0	1	0	1	0	0	0	33.3%
1.4%	11.6%	0.0%	0.5%	0.0%	0.5%	0.0%	0.0%	0.0%	16.7%
0	0	17	0	0	3	0	3	73.9%	
0.0%	0.0%	7.9%	0.0%	0.0%	1.4%	0.0%	1.4%	26.1%	
2	0	1	14	0	2	0	2	66.7%	
0.9%	0.0%	0.5%	6.5%	0.0%	0.9%	0.0%	0.9%	33.3%	
0	0	2	0	22	0	0	0	91.7%	
0.0%	0.0%	0.9%	0.0%	10.2%	0.0%	0.0%	0.0%	8.3%	
0	0	0	2	0	30	1	0	90.9%	
0.0%	0.0%	0.0%	0.9%	0.0%	13.9%	0.5%	0.0%	9.1%	
0	1	1	1	0	1	25	1	33.3%	
0.0%	0.5%	0.5%	0.5%	0.0%	0.5%	11.6%	0.5%	16.7%	
0	0	5	1	2	2	1	30	73.2%	
0.0%	0.0%	2.3%	0.5%	0.9%	0.9%	0.5%	13.9%	26.8%	
66.7%	32.6%	65.4%	73.7%	91.7%	76.9%	36.2%	81.1%	30.1%	
33.3%	7.4%	34.6%	26.3%	8.3%	23.1%	13.8%	18.9%	19.9%	

Figure 9: Exp1.1, SERS, SER-NN, confusion matrix with RAVDESS, without FS

3.1.2 SERS Influence on Dataset and Emotion

Fig. 12 shows the results of Exp1.1, the influence of the FC-CSI and SER-NN classification algorithms on the three datasets, and the emotions represented in those datasets.

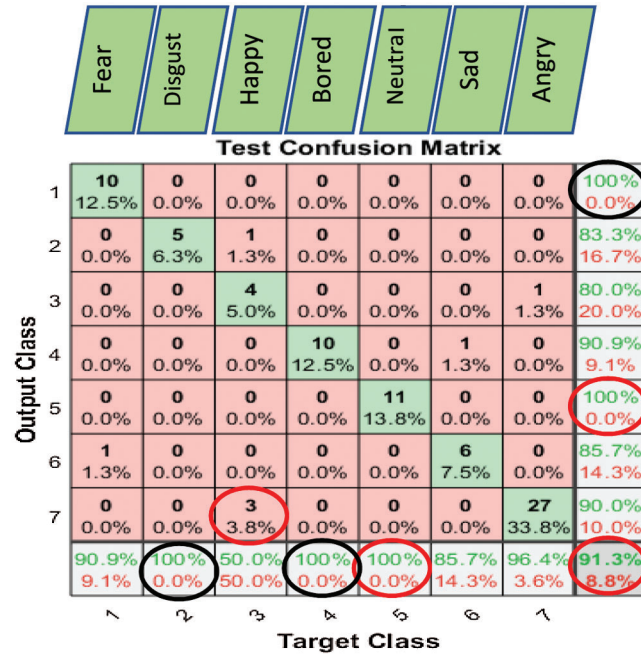


Figure 10: Exp1.1, SERS, SER-NN, confusion matrix with EMO-DB, without FS

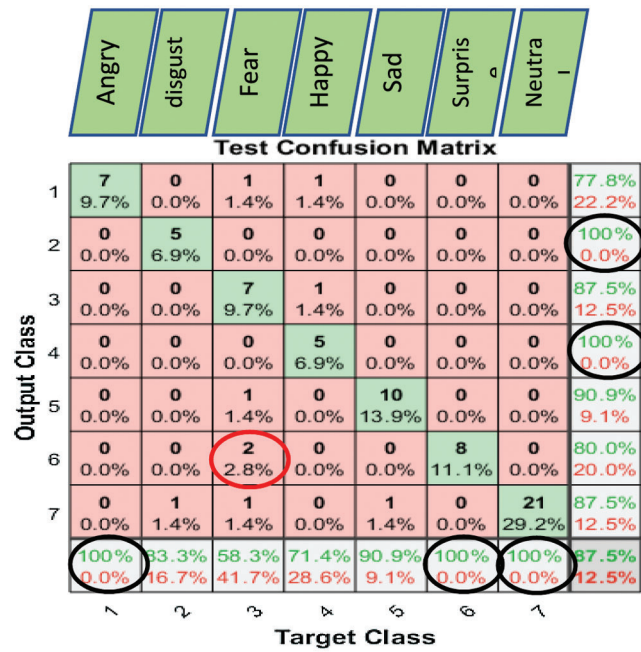
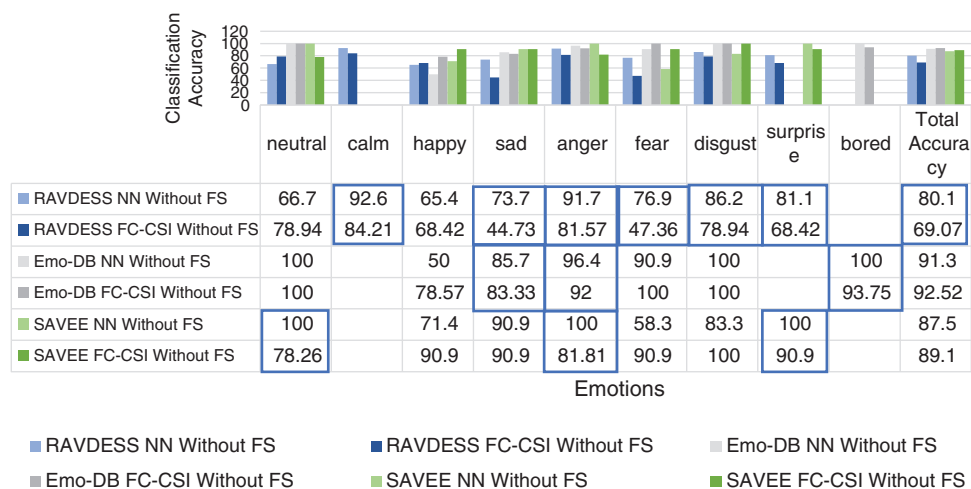


Figure 11: Exp1.1, SERS, SER-NN, confusion matrix with SAVEE, without FS

Table 3: FC-CSI classification accuracy compared to designed SER-NN

	Proposed classification method	NN
RAVDESS	69.07%	86.1%
Emo-DB	92.52%	96.3%
SAVEE	89.1%	91.7%

**Figure 12:** Exp1.1, SERS bar chart, result comparison between FC-CSI and SER-NN, without FS

We observe the following:

- FC-CSI showed lower performance for emotions with low amplitude, such as calm and bored;
- FC-CSI showed average performance for emotions that contain many silent fragments, such as neutral;
- FC-CSI failed to improve performance for the anger emotion on all three datasets;
- The performance for all emotions of FC-CSI on RAVDESS did not appear similar to that of Emo-DB and SAVEE, except for the happy emotion, where FC-CSI performed better on all three datasets.

3.2 Experiment 1.2: SERS Performance Analysis After Deploying FS-TF Feature Selection Algorithm

Through Exp1.2, the classification performance of SERS was calculated by two different approaches, first utilizing the FC-CSI algorithm, and then SER-NN. Both approaches were implemented by deploying the FS-TF algorithm.

3.2.1 SERS Accuracy Performance

The column charts in Figs. 13 and 14 compare the performance of FC-CSI and SER-NN after applying FS-TF on the 2182 FE-MD features. FC-CSI suffered from feature selection because, with more features, the possibility of different shapes of CSI curves increased. Therefore, decreasing the number of features affected the performance of FC-CSI, but improved the performance of SER-NN.

Regardless of the best classification accuracy results gained from the NN, the proposed classification method accomplished the results shown in Tab. 3 using a smaller number of features, as shown in Tab. 4, which shows less complexity.

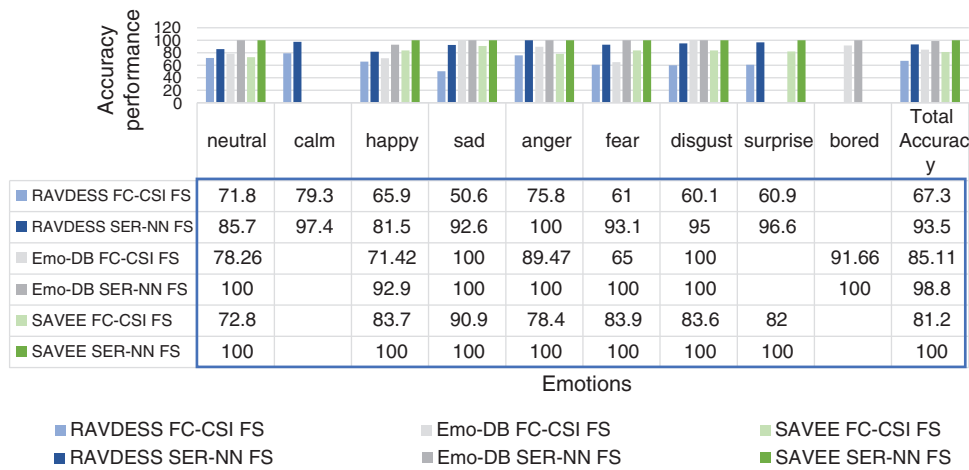


Figure 13: Exp1.2, SERS bar chart, results comparison between FC-CSI and SER-NN, with FS

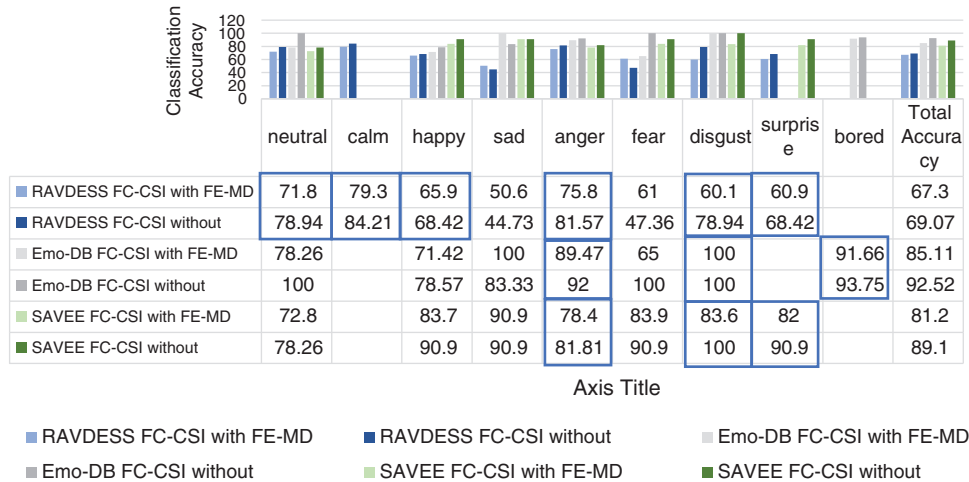


Figure 14: Exp1.5, SERS bar chart, results comparison study between FC-CSI before and after FS

Table 4: Number of features used to classify features using proposed classification method and designed NN.

	Proposed classification method	NN
RAVDESS	285	333
Emo-DB	104	247
SAVEE	89	270

3.2.2 SERS Influence on Dataset and Emotion

We compared the performance of FC-CSI before and after FS-TF to show how SERS suffered from the FS algorithm, and how it affected the emotions in each dataset (see Fig. 14). The results show that the neutral, calm, happy, anger, disgust, surprise, and bored emotions were quite affected by the FS algorithm, except for sad and fear, which means that FS generally affected the performance of FC-CSI on all three datasets.

3.2.3 General Clarification Aspects of FC-CSI Algorithm Performance

We clarify the results of FC-CSI. As discussed, the algorithm finds MCSI curves through the mean of 80% of the audio samples in the dataset, where each MCSI represents a different emotion. We find the Euclidean distance between the CSIs of each audio sample in the testing samples and classify each sample to the nearest MCSI. The MCSI curves that represent the emotions in RAVDESS, Emo-DB, and SAVEE are shown in Figs. 15–17, respectively.

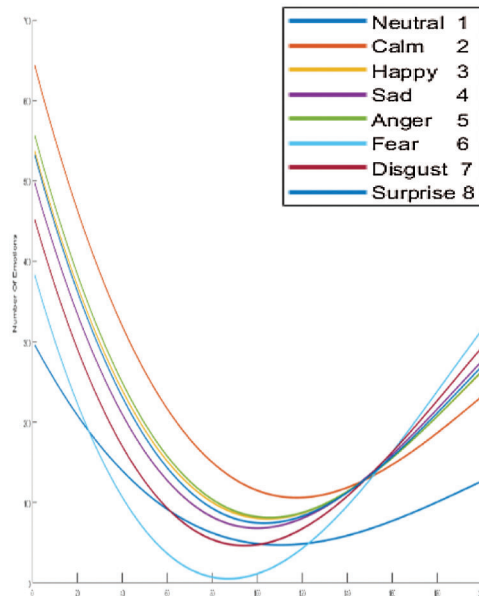


Figure 15: Exp1.1, SERS MCSI curves with RAVDESS, without FS

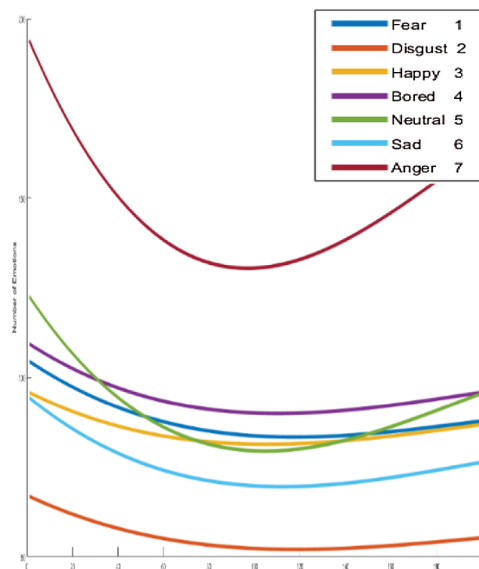


Figure 16: Exp1.1, SERS MCSI curves with EMO-DB, without FS

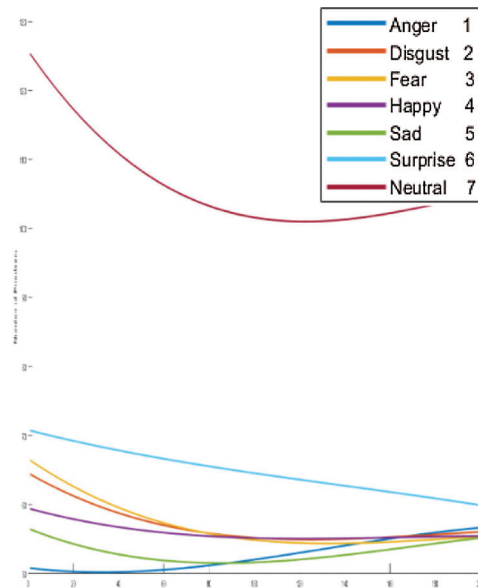


Figure 17: Exp1.1, SERS MCSI curves with SAVEE, without FS

It is obvious from looking at the curves in Figs. 15–17 that the MCSI curves related to Emo-DB and SAVEE are not similar. They intersect in some areas and are parallel in others, but as observed, each MCSI curve has a part that is not similar to any other MCSI curve in the same dataset, whereas the RAVDESS MCSI curves are parallel in many areas and intersect in others, which indicates the low performance of FC-CSI with RAVDESS.

4 Conclusions and Future Work

FC-CSI introduces a new research area in classification. No related work was found that deployed curves in classification, specifically the CSI. The most important conclusion is that CSI is a powerful data analysis tool. Splines correlate data efficiently and effectively, no matter how random the data may seem. Once the algorithm for spline generation is produced, interpolating data becomes easy. It was also noticed that statistical classification methods need less time than SER-NN methods, because SER-NN's need time to learn and validate, whereas statistical methods require no learning. A set of speech signals representing a certain emotion shares a similar CSI behavior, which can be used in SER systems. Curve classification does not suffer from distinguishing conflicted emotions such as calm and neutral, or happy and anger, and they are gender-, speaker-, and language-independent. Possible future work based on this work is as follows. First, it would be more challenging to develop the proposed classification algorithm to deal with audio samples of unequal length. Second is to deploy a more complex distance measurement function, such as Fréchet distance. Third, we used CSI, but it is worth trying other types of curves, such as Bezier curves, with a higher dimension, such as three dimensions.

The proposed FC-CSI classification algorithm can also be applied to voice, speaker, and gender recognition, and can be used in classification systems such as image detection and object recognition. It can also be used to detect diseases and infections.

Acknowledgement: Thanks, and appreciation to everyone who contributed to this scientific research, especially the researchers cited in this paper.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] S. McKinley and M. J. C. O. T. R. Levine, "Cubic spline interpolation," in *Math 45: Linear Algebra*, 1st ed., vol. 45. CA, USA: Eureka, pp. 1049–1060, 1998.
- [2] B. A. Barsky, "Computer graphics and geometric modeling using Beta-splines," in *Computer Science Workbench*, 1st ed., vol. 2. Berlin Heidelberg, GmbH, NY, USA: Springer-Verlag, pp. 1–154, 2013.
- [3] R. H. Bartels, J. C. Beatty and B. A. Barsky, "An introduction to splines for use in computer graphics and geometric modeling," in *Computer Graphics*, 1st ed., vol. 1. CA, USA: Morgan Kaufmann, pp. 1–476, 1996.
- [4] C. R. Chikwendu, H. K. Oduwale and S. I. Okoro, "An application of spline and piecewise interpolation to heat transfer (cubic case)," *Mathematical Theory and Modeling*, vol. 5, no. 6, pp. 28–39, 2015.
- [5] K. J. J. O. T. S. Erdogan, "Spline interpolation techniques," *Journal of Technical Science and Technologies*, vol. 2, no. 1, pp. 47–52, 2013.
- [6] J. Lian, Y. Wentao, X. Kui and L. Weirong, "Cubic spline interpolation-based robot path planning using a chaotic adaptive particle swarm optimization algorithm," *Mathematical Problems in Engineering*, vol. 2020, no. 1, pp. 1–20, 2020.
- [7] M. Gülüm, M. K. Yesilyurt and A. J. F. Bilgin, "The performance assessment of cubic spline interpolation and response surface methodology in the mathematical modeling to optimize biodiesel production from waste cooking oil," *Fuel*, vol. 255, no. 1, pp. 115778, 2019.
- [8] W. Lim, D. Jang and T. Lee, "Speech emotion recognition using convolutional and recurrent neural networks," in *Proc. APSIPA IEEE*, Jeju, Korea (South), pp. 1–4, 2016.
- [9] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou *et al.*, "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proc. ICASSP IEEE*, Shanghai, China, pp. 5200–5204, 2016.
- [10] S. Kwon, "A CNN-Assisted enhanced audio signal processing for speech emotion recognition," *Sensors*, vol. 20, no. 1, pp. 183, 2020.
- [11] O. C. Duarte and P. A. A. M. J. I. J. O. A. E. R Junior, "Application of cubic spline interpolation to fit the stress-strain curve to SAE, 1020 Steel," *International Journal of Advanced Engineering Research and Science*, vol. 4, no. 11, pp. 84–86, 2017.
- [12] S. Ahmed, F. Fini, S. P. Karthikeyan, S. S. Kumar, S. S. Rangarajan *et al.*, "Application of cubic spline interpolation in estimating market power under deregulated electricity market," in *Proc. TENCON 2015-2015 IEEE Region 10*, Macao, China, pp. 1–6, 2015.
- [13] K. Kodera, A. Nishitani and Y. Okihara, "Cubic spline interpolation based estimation of all story seismic responses with acceleration measurement at a limited number of floors," *Japan Architectural Review*, vol. 3, no. 4, pp. 435–444, 2020.
- [14] H. Li, Q. Xuyao, Z. Di, C. Jiabin and W. Pidong, "An improved empirical mode decomposition method based on the cubic trigonometric B-spline interpolation algorithm," *Applied Mathematics and Computation*, vol. 332, no. 1, pp. 406–419, 2018.
- [15] D. R. Roe, "Improving the speed of volumetric density map generation via cubic spline interpolation," *Journal of Molecular Graphics and Modelling*, vol. 104, no. 1, pp. 107832, 2021.
- [16] S. R. Livingstone and F. A. Russo, "The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PLOS ONE*, vol. 13, no. 5, pp. e0196391, 2018.

- [17] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier and B. Weiss, "Database of German emotional speech," in *Proc. INTERSPEECH*, Lisbon, Portugal, pp. 1–4, 2005.
- [18] P. Jackson and S. Haq, *Surrey Audio-Visual Expressed Emotion (SAVEE) Database*. Guildford, UK: University of Surrey, 2014.
- [19] P. Ekman, "Basic emotions," in *Handbook of Cognition and Emotion*, 8th ed., vol. 98. Chichester, West Sussex, UK: John Wiley & Sons, Inc., pp. 1–16, 1999.
- [20] L. Liu, J. Chen and L. Xu, "Realization and application research of BP neural network based on MATLAB," in *Proc. 2008 Int. Seminar on Future BioMedical Information Engineering*, Wuhan, China, IEEE, 2008.
- [21] D. Salomon, "Curves and surfaces for computer graphics," in *Graphics*, 1st ed., vol. 1. NY, USA: Springer Science and Business Media, Inc., pp. 1–466, 2007.
- [22] J. Russell and R. Cohn, "Runge's phenomenon," in *Mathematics*, 1st ed., vol. 46. Berlin, Germany: Zeitschrift für Mathematik und Physik, pp. 1–678, 2012.
- [23] J. P. Boyd and F. Xu, "Divergence (Runge phenomenon) for least-squares polynomial approximation on an equispaced grid and Mock-Chebyshev subset interpolation," *Applied Mathematics and Computation*, vol. 210, no. 1, pp. 158–168, 2009.
- [24] A. Ralston and P. Rabinowitz, "A first course in numerical analysis," in *Numerical Analysis*, 2nd ed., vol. 1. NV, USA: Dover Publications, Inc., Courier Corporation, pp. 1–302, 2001.
- [25] D. M. Abed, A. M. Jaber and A. Rodhan, "Improving security of ID card and passport using cubic spline curve," *Iraqi Journal of Science*, vol. 57, no. 4A, pp. 2529–2538, 2016.
- [26] S. R. AbdulMonem, A. M. J. Abdul Hussien and M. A. Abbas, "Developing a mathematical method for controlling the generation of cubic spline curve based on fixed data points, variable guide points and weighting factors," *Engineering and Technology Journal*, vol. 33, no. 8B, pp. 1430–1444, 2015.
- [27] J. A. Abdul-Mohssen, "3D surface reconstruction of mathematical modelling used for controlling the generation of different Bi-cubic B-spline in matrix form without changing the control points," *Engineering and Technology Journal*, vol. 34, no. 1B, pp. 136–152, 2016.
- [28] C. A. Hall and W. W. Meyer, "Optimal error bounds for cubic spline interpolation," *Journal of Approximation Theory*, vol. 16, no. 2, pp. 105–122, 1976.
- [29] I. J. Schoenberg, "Contributions to the problem of approximation of equidistant data by analytic functions. Part B. On the problem of osculatory interpolation," *Quarterly of Applied Mathematics*, vol. 4, no. 2, pp. 112–141, 1946.
- [30] D. Cohen, L. B. Theodore and D. Sklar, "Precalculus: A problems-oriented approach, enhanced edition," in *Algebra & Trigonometry*, 6th ed., vol. 1, United Kingdom: Cengage Learning, pp. 1–1184, 2004.