



Facial expression recognition with dynamic cascaded classifier

Abubakar M. Ashir¹ · Alaa Eleyan² · Bayram Akdemir¹

Received: 30 April 2018 / Accepted: 9 March 2019 / Published online: 19 March 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

In this paper, a new approach for facial expression recognition has been proposed. The approach has imbedded a new feature extraction technique, new multiclass classification approach and a new kernel parameter optimization for support vector machines. The scheme of the approach begins with feature extraction from the input vectors, and the extracted features are transformed into a Gaussian space using compressive sensing techniques. This process ensures feature vector dimensionality reduction and matches the features vectors with radial basis function kernel used in support vector machines for classification. Prior to classification, an optimized parameter for support vector machines training is automatically determined based on an approach proposed which relies on the receiver operating characteristics of the support vector machine classifier. With the optimized kernel parameter, new proposed multiclass classification approach is used to finally classify any vector. In all the experiments conducted on the two facial expression databases with different cross-validation techniques, the proposed approach outperforms its counterparts under the same database and settings. The results further confirmed the validity and advantages of the proposed approach over other approaches currently used in the literature.

Keywords Facial expression recognition · Compressive sensing · Support vector machine · Radial basis function kernel

1 Introduction

The Quest for information and intelligence gathering constitutes an integral part of modern day technological demands. This has momentarily grown ever since the major breakthrough in digital technology. Automatic identification systems have drawn a lot of interests from researchers driven by an insatiable need for such systems in many applications. Among the numerous methods for identity establishments and information gathering are face recognition and facial expression recognition. Facial expression recognition (FER) is an emerging technology

with an increasing patronage from many areas such as security, medical diagnosis, psychology, human–computer interaction (HCI) and entertainments. One of the very recent applications of FER is in marketing where goods and services are considered *stimuli* and the consumers' *response* (satisfaction to the products or services) is measured automatically and objectively through FER. However, the challenges in FER are enormous and tasking in nature [1].

Automatic recognition of facial expressions from face images is more complicated than standard face recognition [2, 3]. Despite the fact that both face recognition and FER need to adjust for common image alterations (i.e., rotation, illumination, scaling, etc.), facial expression may depend on human ethnicity, culture and even the nature and the strength of the *stimuli* leading to an expression. In addition, there are intrinsic correlations among the seven expression classes (e.g., anger, disgust, happy, fear, neutral, sad and surprise) making some pairs of classes easy to be recognized; some others, hard. Expressions such as sadness and anger may be difficult to differentiate in some individuals [2]. In the context of algorithm implementation, the general framework of FER remains the same. The algorithm framework begins from facial image preprocessing, feature

✉ Abubakar M. Ashir
muhammad.ashir@admotomasyon.com

Alaa Eleyan
aeleyan@avrasya.edu.tr

Bayram Akdemir
bakdemir@ktun.edu.tr

¹ Department of Electric and Electronics Engineering, Selcuk University, Konya, Turkey

² Department of Electric and Electronics Engineering, Avrasya University, Trabzon, Turkey

extraction, classification and finally decision making based on the output of the classifier algorithm [3].

In the hierarchy of FER algorithm framework, feature extraction and classifier algorithms have been the most widely investigated research topics all in an attempt to achieve an improved performance and robustness. This is not to downplay the contributions of the other stages which may be valuable as well and contribute to overall performance. In the realm of feature extraction algorithms, FER is generally classified into appearance-based (holistic) and feature-based [2]. Whereas the former used the whole facial image to extract the salient features, the later tries to take some geometrical and localized neighborhood measurements of some key points (action units) within the face image such as eye, nose, mouth and the chin [3]. Among the popular general methodologies for feature extraction include local binary pattern, discrete cosine transform (DCT), discrete wavelet transform (DWT), complex wavelet transform (CWT), curvelet transform (CT), Gabor wavelet transform (GWT) and many others [4–12]. Feature extractors such as DWT and GWT have a multi-resolution approach by breaking down the image into sub-bands of frequencies or *wavelets* before extracting features. These feature extractors are hugely successful in many pattern recognition applications such as texture, face, retina and finger print recognition, but the same cannot be said of FER due to its complex nature [2]. For instance, authors in [11] applied PCA by extracting principal eigenvectors of the data as a feature and not so impressive results were reported. Due to the low performance of the conventional feature extraction algorithms in FER, researchers deploy multi-classification and fusion techniques at features, score and decision level of the both feature extractor and classifier algorithms. One group of feature extractors with best performance in FER is the multi-resolution feature extractors like GWT. Gabor transform has been hugely successful in many face-related applications and still remains an excellent feature extractor for face-related patterns [6]. Its success can be attributed to its higher capacity (i.e., many number of tunable parameters) and response similar to receptive fields of simple cells in the primary visual cortex of human eyes [1] Authors in [3–7] implemented variants of GWT feature extractors to obtain more robust features for FER problems. However, GWT and other multi-resolution algorithms demand intensive computation and higher memory usage [7]. Due to these setbacks, most of the times dimensionality reduction algorithms are required to trim down the size of the feature vectors. Some of the state-of-the-art dimensionality reduction algorithms are discussed in [8–11]. This becomes a hindrance in real-time applications where speed and simplicity are of great interest.

On the other hand, classifier algorithms also have resounding contributions toward the overall performance of any pattern recognition problems. Distance metric classifiers such as Manhattan (δ_{l_1}), Euclidean (δ_{l_2}) and Cosine (δ_{\cos}) have been used in the literature for their simplicity and less computation [7]. These classifiers do not require training. But in a more tasking and machine learning problems, distance measures-based classifiers may not have the upper hand due to difficulties in learning the patterns, and some more sophisticated and state-of-the-art learning-based classifiers such as support vector machine (SVM) and neural network (NN) are frequently used. These learning-based classifiers equally have their challenges as well which include convergence (i.e., ability to learn the pattern), training error, generalization error, suitable parameter selections and so on [13, 14]. Moreover, most of these classifiers were initially designed for binary classification, and the extension of their original goal to multi-classification problems has been one of the active fields of research in pattern recognition problems.

The generalization error is being considered as one performance enhancer for such classifiers and is linked to both errors on the training examples and complexity of the classifier [13, 15]. Trade-off exists between higher-capacity classifiers (i.e., with large number of adjustable parameters) and low-capacity classifiers (i.e., within sufficient number of adjustable parameters). Low-capacity classifiers might not be able to learn the task at all but when they do they exhibit good generalization due to their low complexity. On the other hand, higher-capacity classifiers can learn any classification pattern according to the learning rules without error, but they generally tend to exhibit poor generalization. However, a good generalization performance can be achieved with higher-capacity classifiers when the capacity of the classification function is matched to the size of the training set [13, 15, 16]. A smart way to improve such classifiers generalization capacity is by optimizing kernel parameters with cross-validation data. Authors in [15] proposed a method for automatically optimizing multiple kernel parameters as a way of improving classifiers generalization accuracy using standard steepest decent algorithms, whereas [2] applied a heuristic optimization approach based on particle swarm optimization (PSO). In general, researchers usually resorted to naïve search techniques or use parameters values based on experience to avoid excessive computational overheads especially in a database with a large number of samples.

The manuscript proposes a new approach to facial expression recognition with contributions at feature extraction and classification levels. The feature extraction process utilizes the compressive sensing technique with statistical analysis of the extracted compressed facial signal

to represent a more robust feature representation for each individual facial expression class. In the classification stage, a dynamic classifier referred as dynamic cascaded classifier (DCC) within the scope of this work has been proposed. DCC leverages on the proposed feature extraction approach to produce an adaptive classification approach with kernel matching techniques which lead to low training error and improve generality of the classification process. In summary, the major contribution of the paper includes (a) new feature extraction techniques and (b) proposes an optimized technique for automatic parameter selection in multilevel classification and (c) dynamic cascaded classifying techniques.

The paper has been categorized into five sections as follows: Sect. 1 contains the introduction and review of the related works. Section 2 presents the theoretical background of some key concepts used in the proposed approach. Section 3 presents the contributions in this paper. In Sect. 4 experimental results are presented along with the performance comparison and discussion. Conclusions are drawn in Sect. 5.

2 Theoretical background

2.1 Support vector machine

Support vector machines are a generally supervised learning algorithm used in classification and regression [13]. The original training algorithm for SVM is referred to as maximum margin training algorithm. For linearly separable patterns, it finds a hyperplane line in a space that maximally separates two patterns. This hyperplane is represented by a decision function $D(x)$ for a pattern vector x with n -dimension belonging to either class A or B . Now, the inputs to the training algorithm become a sequence $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where y is the class level for each pattern in x which only takes two values for binary classification according to (1) (Fig. 1) [13].

$$\begin{cases} y_k = 1, & \text{if } x_k \in \text{class } A \\ y_k = -1, & \text{if } x_k \in \text{class } B \end{cases} \quad (1)$$

During the training phase, the decision function $D(x)$ is determined according to the rule in (2).

$$\begin{cases} x \in A, & \text{if } D(x) > 0 \\ x \in B, & \text{otherwise} \end{cases} \quad (2)$$

The decision function $D(x)$ has two forms of representations which are the (i) direct space (3) and (ii) dual space (4). The two representations work similarly to maximize the margin of distance M between the patterns and the separating hyperplane represented by $D(x)$ [11].

$$D(x) = \sum_{k=1}^n \omega_k \varphi_i(x) + b \text{ or } D(x) = w \cdot x + b \quad (3)$$

$$D(x) = \sum_{k=1}^n \alpha_k k(x_k, x) + b \text{ or } D(x) = \alpha \cdot x + b \quad (4)$$

where in direct space form, ω and b are the parameters (weights and bias) to be adjusted during training and $\varphi_i(x)$ are predetermined function of a pattern x of dimension n , referred to as kernel function. In dual-space representation α_k and b become the adjustable parameters (weights and bias) and $k(x_k, x)$ is the dual kernels function. A number of kernel functions have been shown to be suitable for the original maximum margin SVM classifier training rules [14]. These kernel functions include perceptron, polynomial, RBF and arc-tangent functions.

With direct space kernels function $\varphi_i(x)$, the distance between a hyperplane and a pattern x is given by $\frac{D(x)}{\omega}$, where ω is the l_2 -norm of the training weights. Assuming that a margin M between class boundary A and B and a pattern x exists, then for every pattern x_k with class level y_k in pattern vector x will be trained to fulfill the inequality in (5) which aims at maximizing the hyperplane margin between patterns and the hyperplane:

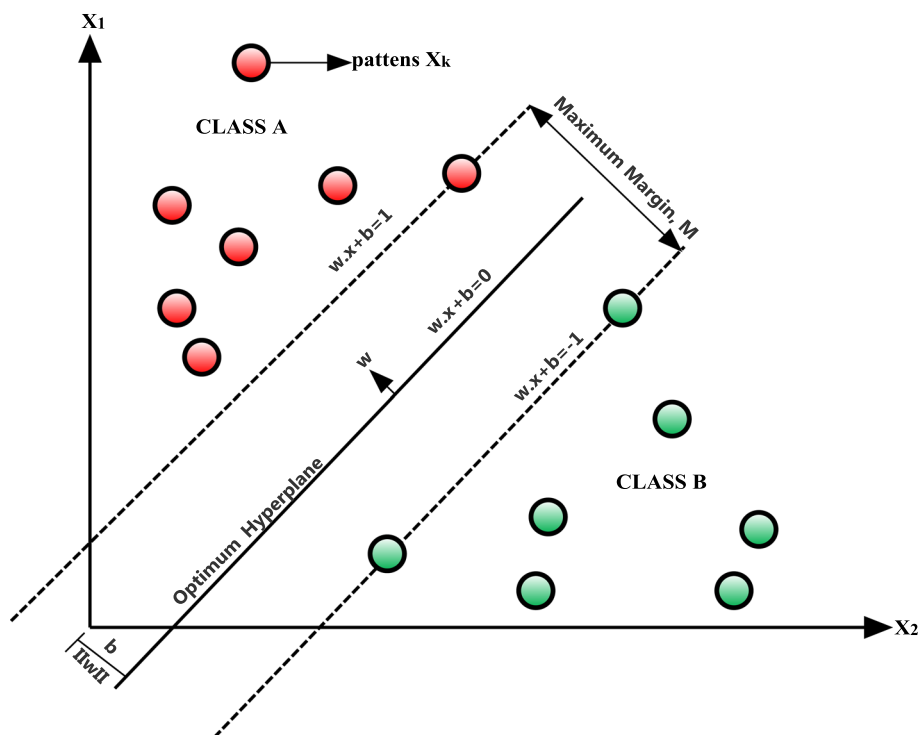
$$\frac{y_k D(x_k)}{\omega} \geq M \quad (5)$$

Now, the problem is reduced to an optimization whereby the objective is to find the weights vector ω which maximizes the margin M in accordance with (5) (Fig. 1).

2.2 Multiclass support vector machine

Most of the earlier machine learning algorithms were initially designed for binary classification, e.g., perceptron learning and SVM [17]. However, in practice multiclass classification problems are very prevalent in real world applications. Over times, researchers have proposed number of multi-classification algorithms based on combination of many binary classifiers or solving the whole classification at ones as an optimization problem. Research in this field is still ongoing. For instance, SVM multiclass learning is generally conducted in two ways. The first approach constructs a number of binary SVM classifiers and then combines them to predict a class to which a new input vector belongs. The most known among these algorithms are: *one-versus-rest (1-vs-r)*, *one-versus-one (1-vs-1)*, *decision directed acyclic graph (DDAG)* and *error-correcting output code (ECOC)* [13, 18]. The second approach considers the whole dataset as one optimization problem and tries to solve the problem in one single step, e.g., “Crammer and Singer” approach [19]. However, this method is very difficult and computationally expensive to

Fig. 1 Maximum margin SVM



solve due to its complexity and is hardly used in practical SVM multiclass classification problems [17].

In *one-versus-rest* as it’s popularly referred to, for datasets with K number of unique classes, for each class, a binary SVM classifier model is constructed (trained) totaling to K -SVM models. Each of the K th SVM is obtained by training the datasets with all members of K th class having the same positive class level (i.e., 1), whereas all the remaining training vectors in the datasets are assigned same negative class level (i.e., -1 or 0). During prediction, a new input vector x is tested in all the K binary SVMs and the output of their decision function $D(x)$ (3) is computed. The classifier model which produces the largest function output is assumed to be the class which the input vector x belongs [20]. For details about *one-versus-one*, *decision directed acyclic graph (DDAG)* and *error-correcting output code (ECOC)*, refer to [21–24].

2.3 Radial basis function

Applications of RBF in machine learning algorithms such as radial basis neural networks originally surfaced in the literature in 1988 through the work of Broomhead and Lowe [25]. They developed the opinion that in most feed-forward networks; the kernel function performs a simple curve fitting operation in a higher-dimensional space. They further demonstrated that learning was synonymous to producing a best fit surface in that space (i.e., higher-dimensional space) to a finite set of data points (training set)

and generalization was equivalent to interpolating the test data on this fitting surface [20, 25]. RBF is any real-valued function, $\phi(x, c)$, whose value depends only on the distance from the origin ($c = 0$), or alternatively on the distance from some other point c , called a *center*, so that:

$$\phi(x, c) = \phi(\|x, c\|) \tag{6}$$

The ℓ_2 -norm operator $\| \cdot \|$ is usually Euclidean distance, although other distance functions are also possible. Most commonly used RBF are Gaussian, multi-quadratic, inverse quadratic and inverse multi-quadratic functions [13]. Gaussian radial function kernel commonly used is represented in (7).

$$\phi(x, c) = e^{-\left(\frac{\|x-c\|^2}{2\sigma^2}\right)} \tag{7}$$

where sigma σ is the standard deviation of the kernel, which is adjustable.

2.4 Compressive sensing theory

After the landmark Nyquist–Shannon sampling theory, an evolution into the digital and information technology era had its major boosts. The theory provided a limit for efficient reconstruction of an analog signal sampled at regularly spaced intervals (i.e., period). It states that if a signal is sampled by a frequency at least twice its bandwidth, then the signal can be perfectly reconstructed from its samples [26]. Sooner enough, researchers realized sampling at Nyquist rate results in a large number of samples resulting

to large data samples, inefficient use of communication channels and huge processing of data. Inspired by the information theory [27], a number of signal encoding algorithms are evolved over time in an attempt to reduce the signal size samples without loss of information. Compressive sensing (CS) is one of the data compression algorithms with even more radical approach to data compression [28].

Generally, in data compression algorithms, a N length signal x in R^N space is transformed into a linear sum of its basis functions $\psi_i (i = 1, 2, \dots, N)$ and scaling vector s as in (8). The signal x is K -sparse if it has at most, K nonzero and, $(N - K)$ zero coefficients in s . The case of interest in CS is when $K \ll N$ [28–30]. CS theory addressed three inefficiencies of the conventional compression techniques. First, the initial number of samples N may be large even if the desired K is small. Second, the set of all N transform coefficients $\{s_i\}$ must be computed even though all but K of them will be discarded. Third, the locations of the large coefficients must be encoded, thus introducing an overhead [28].

$$x = \sum_{i=1}^N s_i \psi_i \tag{8}$$

CS considers a general linear measurement process that computes $M < N$ inner products between x . The measurement y of x can be obtained from a $M \times N$ measurement matrix $\Phi = [\phi_1, \phi_2, \phi_3, \dots, \phi_N]$, with column vectors $\phi_i (i = 1, 2, \dots, M)$. Using vector form of (8), a new measurement vector y is computed.

$$y = \Phi x = \Phi \Psi s = \Theta s \tag{9}$$

where $\Theta = \Phi \Psi$ and coefficients vector s can be obtained $s = \Theta^T y$. The measurement matrix must allow the reconstruction of the N length signal x from $M < N$ measurements provided it satisfies (10) for an arbitrary $3K$ -sparse vector v . This condition is referred to as restricted isometry property (RIP) [28].

$$1 - \epsilon \leq \frac{\|\Theta v\|_2}{\|v\|_2} \leq 1 + \epsilon \tag{10}$$

where v is any vector sharing the same non-zero entries as s and for some $\epsilon > 0$. The authors in [20] proved that the measurement matrix satisfying (10) can be obtained with high probability from independent identically distributed (i.i.d) random variables of the Gaussian probability density function with mean zero and variance $1/N$.

3 Proposed approach

3.1 Proposed feature extraction

One of the important attributes of any feature extraction algorithm is its ability to represent the original row information in another form (features) while retaining the original data uniqueness. In the proposed approach, arithmetic mean difference (AMD) and CS were proposed as the feature extractors. Initially, a 2D image sample I of size $M \times N$ is vectorized and then normalized within an interval $[0 \ 1]$ to form a column vector x of size $Q \times 1$, where $Q = M \times N$. The AMD vector x^{amd} of the normalized image vector, x is computed as follows:

$$x_i^{amd} = x_i - \frac{1}{Q} \sum_{i=1}^Q x_i \quad (i = 1, 2, \dots, Q) \tag{11}$$

The final feature f_v is computed using x^{amd} and measurement matrix Φ , obtained using the CS theory (described above) as follows:

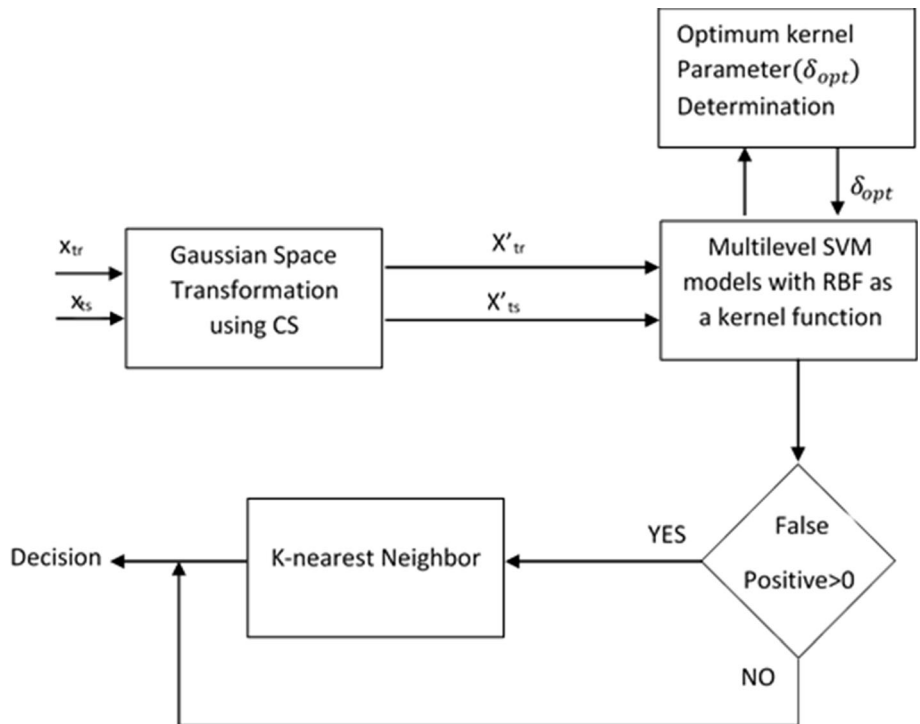
$$f_v = \Phi \cdot x^{amd} \tag{12}$$

where ‘ \cdot ’ is the dot product operator.

3.2 Dynamic cascaded classifier

The objective of DCC is to form a classifier capable of learning any training patterns with minimum error and a good generalization capacity. The trade-offs between training error and generalization error of the learning-based classifiers have received lots of attention. As discussed in the preceding section, higher-capacity classifiers can learn any training pattern without error but exhibits poor generalization due to their complexity [13]. However, it has been shown that a good generalization performance can be achieved when the capacity of the classification function is matched to the size of the training set or by minimizing the complexity of the classifier [13]. Therefore, the proposed DCC addressed the issue of minimizing training error using CS by matching the training sets with the classifier’s kernel. The second problem of generalization is addressed by proposing an automatic way of determining the classifier optimum kernel’s parameter. Proper tuning or selection of the classifier’s kernel parameters has been associated with low generalization error [17]. Based on this, DCC has been designed as a three-stage classifier built around binary SVMs with RBF as a kernel function. Figure 2 represents the schematic of the proposed DCC with training data x_{tr} and testing data x_{ts} as inputs.

Fig. 2 Proposed DCC schematic diagram



3.2.1 Stage one

The first stage in the cascade is a prerequisite to the succeeding stages. The aim here is to transform the training vectors into a random variable Gaussian space so that the training sets match the Gaussian RBF kernel. CS with a random variables Gaussian measurement matrix ensures this goal is achieved. Since such kernel basically performs a task synonymous to curve fitting [20], best fit with least mean square error is estimated. The Gaussian transformation of the training patterns can be realized based on (9) as described in CS theory. In accordance with the theory, the measurements preserved the class uniqueness of each input vector x . In addition, in this way a dimensionality reduction is also realized since the measurements matrix Φ can be set to predefined length. For a training vector x , the Gaussian transformation according to CS will produce a vector x' such that:

$$x' = \Phi \cdot x \tag{13}$$

3.2.2 Stage two

This is the strongest link in the cascade where maximum margin binary SVM classifier models are created and trained with RBF as kernel function. The numbers of models created are equals to the number of classes in the training sets as in *one-versus-rest* multiclass training. Sequel to this training, an optimum value of RBF sigma

(σ_{opt}) is determined from the portion of datasets which have not been used for training. The two key processes in the link are described in the following subsections.

3.2.2.1 Proposed automatic RBF kernel parameter selection

A key factor to optimizing the classifiers generalization error is proper selection of the kernel’s parameter [14]. The major goal here is to automatically select kernel parameter that minimizes the classifier’s generalization error. To optimize this parameter (σ) for RBF kernel, an optimization technique based on the receiver operating characteristic (ROC) analysis of the classifier has been proposed. A cross-validation with the portion of the database which has not been used during training is used to generate the responses of the K trained binary classifiers.

Assuming K binary SVM models are created based on training dataset, where K is equals to the total number of expressions classes in the dataset. Measurement matrixes, i.e., false accept rate (FAR) and false reject rate (FRR), are generated using the cross-validation dataset. For each cross-validation vector x , K scores are recorded being the outputs of K binary SVM decision function $D(x)$. Since each cross-validation vector x can only belong to only one 1 binary SVM model (i.e., a particular expression class), there are $P(p = 1)$ number of genuine claim and $N(n = k - 1)$ number of imposters claims to be made and hence FAR and FRR can be computed for each cross-validation vector x , as follows:

$$\text{FAR} = \frac{1}{N} \sum_{i=1}^N 1 * (\delta(I_i, T_i) > \gamma) \quad (14)$$

$$\text{FRR} = \frac{1}{P} \sum_{i=1}^P 1 * (\delta(I_i, T_i) < \gamma) \quad (15)$$

where $\delta(I_i, T_i)$ is the measurement matrix between the input vector I and template vector T and γ is the hard decision threshold. The matrix $\delta(I_i, T_i)$ is equivalent to the SVM decision function $D(x)$, and the threshold γ is equal to zero as used in binary SVM. Both expressions $(\delta(I_i, T_i) > \gamma)$ and $(\delta(I_i, T_i) < \gamma)$ are evaluated as 1 if true, otherwise are evaluated as 0.

An ideal situation is where both FRR and FAR are equivalent to zero and hence the cumulative objective function F , for M cross-validation vectors, is defined as follows;

$$F = \min \sum_{j=1}^M \left\{ \frac{1}{N} \sum_{i=1}^N 1 * (\delta(I_i, T_i) > \gamma) + \frac{1}{P} \sum_{i=1}^P 1 * (\delta(I_i, T_i) < \gamma) \right\} \quad (16)$$

The RBF kernel parameter σ , which is to be optimized and determined automatically, is arbitrarily initialized and then updated using (17).

$$\sigma_{\text{new}} = \sigma_{\text{old}} + \Delta\sigma \quad (17)$$

$\Delta\sigma$ is the incremental step of σ and is determined based on the present and preceding output of the objective function F to ensure speedy convergence.

$$\Delta\sigma = 0.1 * (F_{\text{old}} - F_{\text{new}}) \quad (18)$$

σ_{new} is returned as the optimal value optimum sigma γ , in the case of convergence or maximum number of iterations is reached.

3.2.2.2 Single-branch decision tree multiclass classification approach Here variant of multiclass SVM proposed in [31] is adopted. The classification approach is similar to *one-versus-rest* algorithm in the training phase but differs in the prediction. During prediction, the winning class to which a new input vector x belongs is evaluated via a single-branch decision tree. The tree has a single branch with *decision nodes* equal to the number of binary SVMs created during training. Each decision *node* represents a unique binary SVM with two leaves (left and right) corresponding to the output response of the *node* when new input vector x is presented to it. Vector x traverses from the *root* of the tree to last node along the *evaluating path*. At each decision node, $D(x)$ is computed and if $D(x) > 0$, the corresponding class on the left side of that node will be assigned a logical 1, whereas if $D(x) \leq 0$, the class is assigned logical 0. At the end of evaluation, only classes

with logical 1 are considered, whereas classes with zero output are eliminated. In the end, candidates with output 1 are further proofed for three possibilities until an undisputed winner is found [31].

3.3 Stage three

Conceptually, the third stage deals with an exceptional circumstance where convergence has not been achieved after a predetermined number of iterations. Sequel to this, a simple classifier but with different orientation (i.e., dissimilar approach) to the binary SVMs used in second stage will suffice. Because a feature may still exist of a testing vector x , associating it with its rightful class which the second stage is unable to exploit. To keep the approach simple having gone far through the cascades, k -nearest neighbor based on δ_{ℓ_2} -norm (Euclidean) distance measure has been proposed to be used in this stage, though other classifiers may work as well. For an n -dimensional training vector x_{tr} and testing vector x_{ts} , their Euclidean distance is measured as;

$$\delta_{\ell_2}(x_{tr}, x_{ts}) = \|x_{tr} - x_{ts}\|^2 \quad (16)$$

4 Simulation results

To evaluate the performance of the proposed approach, two standard methods for performance evaluation in FER are adopted. These methods include expression specific performance and the general cross-validation results which give the overall performance of the algorithm. Three popular FER databases were used to evaluate the performance of the proposed approach. Extensive experiments were conducted on these FER databases which include Japanese Female Facial Expression (JAFFE) [5] and Cohn-Kanade (CK) [32] and MMI facial expression database [33]. On all databases, identity-independent (II) approach was adopted, i.e., all the samples in the database were grouped into their constituents' expression classes (i.e., anger, disgust, fear, happy, neural, sad and surprise) independent of the person to which the samples belong. This implies that based on the seven basic facial expressions considered, there are maximum of seven possible classes independent of the size of the database. The results obtained from the proposed approach were validated into categories:

- (a) Comparisons with existing state-of-the-art SVM multilevel classification algorithms. In both cases, proposed RBF kernel parameter selection is used to select the kernel's optimum parameter (sigma).

- (b) Comparison with results obtained by other researchers under the same databases and experimental procedures.

The results presented in Tables 1, 2 and 3 compared proposed DCC results with other multiclass classification approaches. The other approaches were implemented using the same proposed feature extraction approach and the optimized kernel parameter computed with proposed approach for parameter selection in this paper.

4.1 Experiments on JAFFE database

The JAFFE database has 213 images from 10 subjects each having 3 to 4 sample images per expression [5]. 210 images were used in this context. Before the training, all the samples were grouped into seven classes based on their expression contents (II). *Leave-one-subject-out (LOSO)* standard cross-validation was used for training and evaluation. In LOSO, at each run of the training, one sample from each of the seven groups was used as a testing set while the remaining samples were used as training set. Each run of the training was repeated with different test samples until each sample was uniquely used as testing data as well as training data. The recognition rate was given as the average performance over all the runs. Figure 3 contains a cross section of the JAFFE database. The vertical columns represent seven PI expression classes, whereas each row represents seven PD expression classes for a single subject.

4.2 Experiments on CK database

CK database contains 97 subjects and total of 8795 sample images [32]. Two standard cross-validation procedures were used here (i.e., *LOSO* and *n-fold* validation). *LOSO* cross-validation technique could be very exhaustive and computationally intensive [1], and hence, *LOSO* cross-validation was run only once at a fixed feature vector length. For the *n-fold* validation, the tenfold cross-

Table 2 LOPO cross-validation results CK+ database using pi grouping

Length	Sigma	<i>l-vs-l</i>	<i>l-vs-rest</i>	<i>DDAG</i>	<i>ECOC</i>	<i>DCC</i>
100	3.80	90.76	96.50	95.99	95.70	97.22
200	4.90	94.50	97.30	97.80	98.25	97.89
300	7.85	92.90	96.91	98.10	96.12	97.98
400	7.90	95.10	97.01	98.20	97.93	98.35
500	11.42	92.32	97.02	97.01	97.81	97.72
600	12.30	90.80	97.89	96.29	96.72	96.01

validation method was used. In tenfold cross-validation using II approach, all the sample images were initially grouped into their constituent's classes based on expression information. Each of the expression group is divided into 10 subclasses with each subclass having the same number of samples. Then, the 10 subclasses from each unique group were recombined (using random selection) with the remaining subclasses from other unique groups to form tenfold with each fold having all the unique group samples (i.e., unique expression classes) and the same size. During training, onefold was used as the testing set whereas the remaining ninefolds were used as training set. The training was repeated 10 times, each time with a different fold as a testing set whereas the remaining ninefolds as the training set. The recognition rate was given as the average performance from the 10 cross-validation runs.

It is noteworthy that the original CK database contains 640×490 sized samples with non-facial background information. In this experiment, all face samples were cropped to remove irrelevant background information and resized to 256×256 . Moreover, every expression in CK database starts from neutral to peak level. In these experiments neural samples were excluded from the non-neutral facial expression classes. Examples of preprocessed images from CK database are shown in Fig. 4.

Table 1 LOPO cross-validation results across databases (JAFFE JF, MMI and CK) using pi grouping

Length	Sigma			<i>l-vs-l</i>			<i>l-vs-rest</i>			<i>DDAG</i>			<i>ECOC</i>			<i>DCC</i>		
	JF	CK	MM1	JF	CK	MM1	JF	CK	MM1	JF	CK	MM1	JF	CK	MM1	JF	CK	MM1
100	8.1	4.0	1.2	72.4	91.7	89.3	76.2	97.6	93.3	75.7	97.0	94.0	77.6	97.7	92.1	90.5	97.7	96.1
200	9.8	5.8	2.0	69.5	95.6	90.0	76.2	98.3	93.7	76.2	98.0	94.0	78.1	98.2	94.5	92.9	97.5	96.8
300	11.2	7.5	2.8	71.9	94.0	90.0	78.6	98.0	95.0	80.0	98.0	94.8	82.4	97.2	95.0	92.4	98.3	98.1
400	12.3	7.3	3.4	75.2	96.2	92.3	81.4	98.5	95.4	80.5	98.1	96.2	80.0	98.3	96.0	90.5	99.3	98.6
500	13.4	10.4	4.0	74.8	93.3	91.5	78.1	97.6	95.6	83.3	97.9	96.3	81.9	98.1	97.3	94.3	97.1	98.4
600	14.2	11.9	4.9	79.1	91.9	90.0	82.4	97.7	94.3	85.7	97.2	96.0	82.4	96.7	95.7	92.4	96.5	97.7

Table 3 Confusion matrix with DCC on JAFFE database

Length = 500	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	90.2	5.3	0.0	0.0	0.0	7.3	0.0
Disgust	3.4	94.7	3.7	0.0	0.0	0.0	0.0
Fear	0.4	0.0	93.0	2.0	0.0	0.0	0.0
Happy	0.0	0.0	0.0	95.7	3.3	0.0	0.0
Normal	0.0	0.0	0.0	2.3	96.3	1.7	0.0
Sad	3.3	0.0	0.0	0.0	0.4	91.0	0.0
Surprise	2.7	0.0	3.3	0.0	0.0	0.0	100

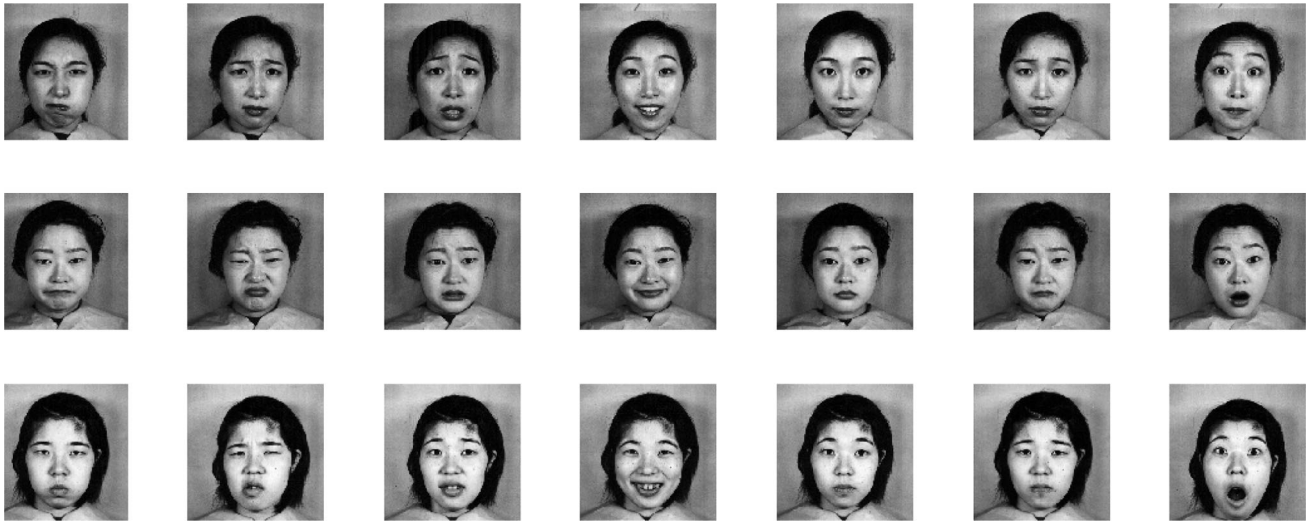


Fig. 3 Cross section of JAFFE database



Fig. 4 Cross section of preprocessed CK database

4.3 Experiments on MMI database

The MMI database was compiled from more than 20 students and research staff members’ images that are from both genders (44% female), ranging in age from 19 to

62 years [33]. For each session, an image sequence is captured that has neutral faces at the beginning and the end. During the experiments, 121 samples sequences were selected from 28 subjects. Seven emotion classes were and LOSO cross-validation procedures. In each sequence, the

first frames are chosen as the neutral while peak frames were used to a facial expression class. The cross-validation results at different feature vector length are presented in Table 1, where Table 4 presents the confusion matrix using the proposed approach.

4.4 Experiments on extended Cohn-Kanade (CK+) database

The C+ database is an extension of the original CK database. It is made up of facial behavior of 210 adults were recorded with two synchronized Panasonic AG-7500 cameras. Participants were 18 to 50 years of age, 69% female, 81% Euro-American, 13% Afro-American and 6% other groups. Participants were instructed by an experimenter to perform a series of 23 facial displays; these included single action units and combinations of action units. Each display began and ended in a neutral face with any exceptions noted. Image sequences for frontal views and 30-degree views were digitized into either 640 × 490 or 640 × 480 pixel arrays with 8-bit gray-scale or 24-bit color values [34].

The same approach of cross-validation was adopted here during the experiments as in the CK dataset training. The results obtained are shown in Table 2.

4.5 Expression-specific performance results

Apart from the cross-validation results, one of the most important indicators on how well FER algorithm performs is, its performances based on expression modes. Sequel to this fact, confusion matrices have been computed on the three databases and presented in Table 3, 4 and 5. The confusion matrices are computed where the proposed approach has its best cross-validation results, i.e., for JAFFE, the best cross-validation result is 94.3% at 500 feature vector length.

4.6 Comparison with other approaches in the literature

Here performance comparison is performed with other state-of-the-arts approaches under the same settings and databases to with the proposed approached. Moreover, the gain in performance between the proposed method and the others also included as contained in Tables 6 and 7.

4.7 Discussions

A new approach has been proposed which is effective, competitive and promising based on the experimental results obtained. The experimental results validate the consistency and good performance of this proposed approach over most of the state-of-the-art approaches compared with. Some novel contributions contained in this paper that could be seen as the backbone to the good performance exhibited by the proposed approach are worthy of being highlighted:

- (a) AMD and CS feature extraction proposed to be used along DCC are simple and help to drastically avoid exhaustive computations as is the case in many feature extractions.
- (b) The concept of using compressive sensing to project the input vectors into random Gaussian space has been instrumental and a prerequisite to ensure that the input vectors, no matter the type, are matched with the RBF kernel of the SVM classifiers. This proposed method has the combined the benefits of dimensionality reduction and generally making it possible for the SVM kernel to be able to learn any pattern with minimum error irrespective of the vectors type. By doing so, the approach makes it possible for SVM classifiers to learn even with a training pattern which conventional SVM might have hitherto fail to learn.
- (c) It could be inferred that the proposed approach for RBF kernel parameter optimization has been very effective. This could be testified from the fact that, all the multiclass classification algorithms used in the

Table 4 Confusion matrix with DCC on MMI database

Length = 400	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	97.0	0.3	0.0	0.0	0.0	0.0	0.0
Disgust	1.4	98.3	2.7	0.0	0.0	0.0	0.0
Fear	0.0	0.0	96.7	0.0	0.0	0.0	0.0
Happy	0.0	0.0	0.0	100	0.0	0.0	0.0
Neutral	0.0	1.4	0.0	0.0	100	2.3	0.0
Sad	1.3	0.0	0.3	0.0	0.0	97.7	0.0
Surprise	0.3	0.0	0.3	0.0	0.0	0.0	100

Table 5 Confusion matrix with DCC on CK database

Length = 400	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	99.0	1.3	0.0	0.0	0.0	1.1	0.0
Disgust	0.7	98.3	0.0	0.0	0.0	0.0	0.0
Fear	0.0	0.0	99.4	0.0	0.0	0.0	0.0
Happy	0.0	0.0	0.0	100	0.0	0.0	0.0
Normal	0.0	0.4	0.0	0.0	100	0.0	0.0
Sad	0.3	0.0	0.3	0.0	0.0	98.9	0.0
Surprise	0.0	0.0	0.3	0.0	0.0	0.0	100

Table 6 Comparison with different approaches on JAFFE database

References	Feature extraction	Dimension reduction	Classifier	Recog. rate (%)	Performance gain (%) by proposed method
[3]	Radial encoded gabor		SVM	89.67	4.63
[35]	LPQ +es_LBP-s	cr_LPP	SVM	76.67	17.63
[36]	LBP	PCA	SVM	53.80	40.50
[36]	LBP	GDA	SVM	65.70	28.60
[36]	LBP	Adaboost	SVM	65.71	28.59
[37]	Gabor Histogram	MV-Boost	SVM	58.70	35.60
[38]	LBP	SRC	SVM	62.90	31.40
[39]	CT	LLE	SVM	63.81	30.40
[39]	ASM	N/A	SVM	68.50	25.80
[40]	LBP + Gabor	N/A	Ensembles SVMs	70.00	24.30
[41]	Gabor Filter	N/A	Fusion SVMs	72.00	22.30
[42]	MB-LGBP	N/A	Two level SVMs	74.18	20.12
[43]	LNBPOA	–(Tenfold)	SVM	95.50	– 1.20
[43]	GEM + LNBPOA	–(Tenfold)	SVM	63.00	31.30
[43]	GEM + LNBPTA	–(Tenfold)	SVM	62.40	31.90
[43]	LBP (no 3D features)	–(Tenfold)	SVM	81.50	12.80
[44]	LDN CODES	–	SVM	90.10	4.20
[45]	TR1DGPA	–	NN classifier	92.20	2.10
[46]	Boosted-LBP	–	SVM	81.00	13.30
[47]	Gabor encoded		FLD	89.60	4.70
Proposed	Proposed	CS	SBTD SVM	94.30	0.00

experiments have good performance. Moreover, the optimization approach for parameter selection is automatic and a convergence is realized in all the conducted and is consistent. However, this proposed automatic parameter selection is limited here to a single parameter optimization. Multiple parameter selection using the proposed method could be computationally expensive as it may require numerous combinations of the multiple parameters to be optimized.

- (d) In summary all the approaches proposed together achieved simplicity, good performance, less classifier generalization error, ability to learn pattern and consistency.

4.7.1 DCC dynamic responses

Figures 5, 6 and 7 show the receiver operating characteristics (ROC) responses of the DCC classifier at different feature vector lengths. These responses compared FAR, FRR and the RBF kernel's parameter using the proposed approach for optimum parameter selection. The show the behavioral pattern of the FAR and FRR before, during and after convergence. From all the three scenarios, it can be seen that the optimum kernel's parameter scales up linearly as the length of the feature vectors increases. It is not surprising how the FRR takes the shape of the RBF kernel, because all the input vectors were transformed into Gaussian space before they were trained with RBF kernel. It is

Table 7 Comparison with different approaches on CK database

References	Subjects	Classes	Validation	Recognition rate (%)	Performance gain (%) by proposed method
[3]	97	7	Tenfold	91.51	6.33
[43]	–	7	Tenfold	97.40	0.44
[43]	–	7	Tenfold	50.20	47.64
[43]	–	7	Tenfold	89.70	8.14
[44]	96	7	Tenfold	96.60	1.24
[46]	–	7	Tenfold	88.90	8.94
[47]	96	7	Tenfold	91.50	6.34
[48]	76	6	Fivefold	90.90	6.94
[49]	90	6		93.66	4.18
[50]	97	6	Tenfold	96.26	1.58
[51]	90	6	LOSO	96.33	1.51
[52]	96	7 (6)	Tenfold	88.4 (92.1)	9.44 (5.74)
[53]	90	7	Tenfold	86.90	10.94
[54]	90	6	LOSO	93.80	4.04
[55]	97	6	–	93.80	4.04
[56]	96	7	LOSO	94.60	3.04
[57]	96	6	Tenfold	86.80	11.04
Proposed	97	7	Tenfold	97.84	0.00

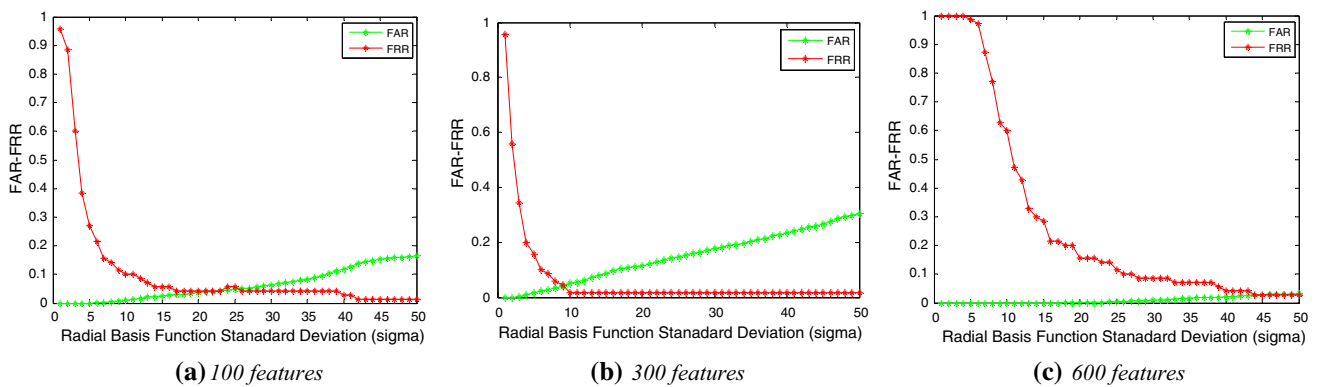


Fig. 5 DCC ROC responses for PI LOPO (JAFPE) training at different feature vector length

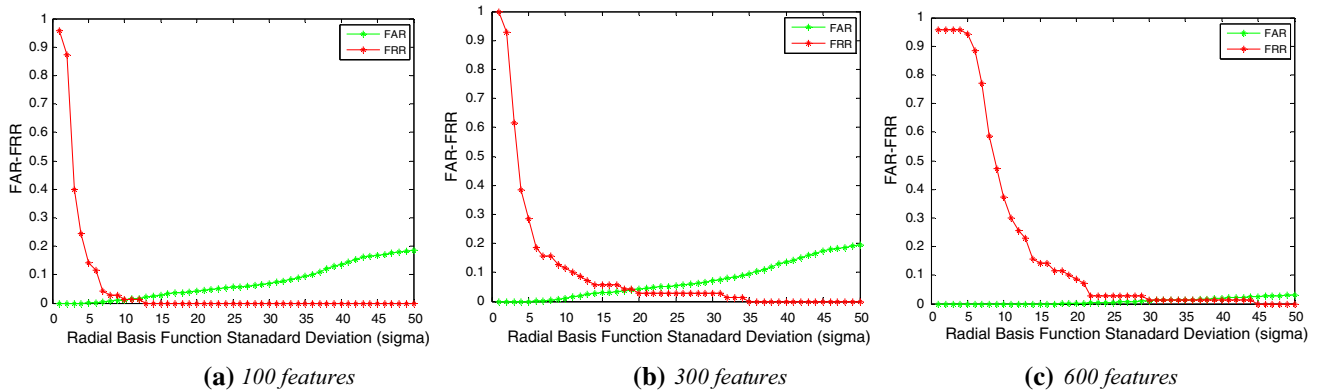


Fig. 6 DCC ROC responses for PD LOPO (JAFPE) training at different feature vector length

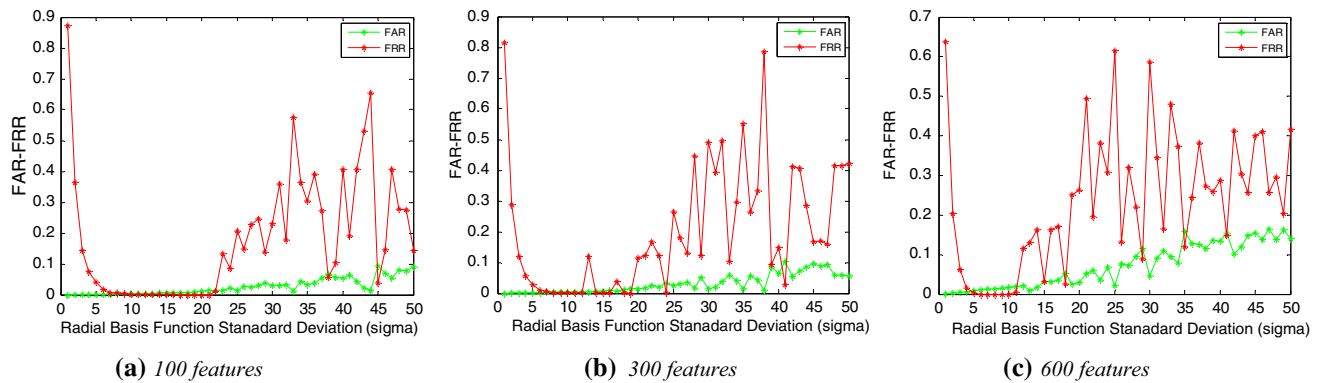


Fig. 7 DCC ROC responses for tenfold (CK) training at different feature vector length

also noteworthy that, the FRR of the DCC in CK (PI) and JAFFE (PI) becomes highly unstable after its best operating point or convergence point is attained. Parameter chosen in that region will result in a very poor generalization performance. Hence, in a situation whereby the error difference of the objective function is set to a value close to zero there is a chance that the convergence may be skipped and a region of instability is experienced. In this exceptional circumstance the training is stopped by the predetermined maximum number of allowable iterations. At this point, third stage of the classifier may be used for reinforcement. In general, the DCC ROC plots testify that the best kernel parameter can always be found within the search space based on the proposed approach for parameter selection in preceding section.

5 Conclusions

A new approach for facial expression recognition has been proposed. The proposed approach was conceived to produce good feature extraction and classifier algorithms for overall performance of the process. The combination of arithmetic mean difference and compressive sensing as feature extractor has been validated by the experimental results and has advantage of less computation. Similarly, the compressive sensing techniques used to transform feature vector into Gaussian space in the proposed DCC ensures that the classifier can operate with any feature extraction algorithm and also capable of learning any training patterns. The experimental results have also shown that the proposed automatic parameter selection approach will always converge to an optimum value based on the rules used for search. Furthermore, the single-branch decision tree proposed has also lived to expectations and has a less overheads in evaluating solution compared to its counterparts. The overall approach has been able to achieve effectiveness and simplicity in feature extraction

and reduced training and generalization error of the classifier which together results in good performance.

Compliance with ethical standards

Conflict of interest We wish to submit a new manuscript entitled “Facial Expression Recognition with Dynamic Cascaded Classifier” for consideration in the Neural Computing and Applications Journal. We confirm that this work is original and has not been published elsewhere nor is it currently under consideration for publication elsewhere and there is no conflict of interest whatsoever.

References

1. Fasel B, Juergen L (2003) Automatic facial expression analysis. *J Pattern Recognit Soc* 36:259–275
2. Minand T, Feng C (2013) Facial expression recognition and its application based on curvelet transform and PSO_SVM. *Int J Light Electron Opt* 124:5401–5406
3. Wenfei G, Cheng X, Venkatesh YV, Dong H, Hai L (2012) Facial expression recognition using radial encoding of local gabor features and classifier synthesis. *J Pattern Recognit Soc* 45:80–91
4. Shiqing Z, Lemin L, Zhijin Z (2012) Facial expression recognition based on gabor wavelets and sparse representation. In: *Proceedings of ICSP*, pp 816–819
5. Michael JL, Shigeru A, Miyuki K, Jiro G (1998) Coding facial expressions with gabor wavelets. In: *Proceedings of AFGR*, pp 200–205
6. Shishir B, Ganesh KV (2008) Recognition of facial expressions using Gabor wavelets and learning vector quantization. *J Eng Appl Artif Intell* 21:1056–1064
7. Baochang Z, Shiguang S (2007) Histogram of Gabor Phase Patterns (HGPP): a novel object representation approach for face recognition. In: *Tran. IP*, pp 57–68
8. Gentile C, Li S, Kar P, Karatzoglou A, Zappella G, Etrud E (2017) On context-dependent clustering of bandits. In: *Proceedings of international conference on machine learning, PMLR*, pp 1253–1262
9. Li S., Karatzoglou A, Gentile C (2016) Collaborative filtering bandits. In: *Proceedings of international conference on research and development in information retrieval, SIGIR*, pp 539–548

10. Korda N, Szörényi B, Li S (2016) Distributed clustering of linear bandits in peer-to-peer networks. In: Proceedings of international conference on machine learning, ICML, pp 1301–1309
11. Yimo G, Zhengguang X (2008) Local Gabor phase difference pattern for face recognition. In: Proceedings of ICPR, pp 1–4
12. Turk M, Pentland A (1991) Eigenfaces for recognition. *J Cogn Neurosci* 3:71–86
13. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of COLT'92, pp 144–152
14. Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for support vector machines. *J Mach Learn* 46:131–159
15. Vapnik V (1991) Principles of risk minimization for learning theory. In: Proceedings of NIPS, pp 832–838
16. Vapnik V (1998) Statistical learning theory. Wiley, New York
17. Hsu C, Lin C (2002) A comparison of methods for multi-class support vector machines. *IEEE Trans Neural Netw* 13:415–425
18. Wang Z, Xu W, Hu J, Guo J (2010) A multiclass SVM method via probabilistic error-correcting output codes. In: Proceedings of ITA, pp 1–4
19. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. In: Tran TPAMI, pp 711–720
20. Lowe D (1989) Adaptive radial basis function nonlinearities, and the problem of generalisation. In: Proceedings of ANN, pp 171–175
21. Knerr S, Personnaz L, Dreyfus G (1990) Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: Fogelman-Soulie F, Hérault J (eds) Neurocomputing: algorithms, architectures and applications, NATO ASI. Springer, Berlin
22. Friedman JH (1996) Another approach to polychotomous classification. Technical report, Stanford Department of Statistics. <https://ci.nii.ac.jp/naid/10017594776/>
23. Platt JC, Cristianini N, Shawe-Taylor J (1999) Large margin DAGs for multiclass classification. <http://papers.nips.cc/paper/1773-large-margin-dags-for-multiclass-classification.pdf>
24. Dietterich TG, Bakiri G (1995) Solving multiclass learning problems via error-correcting output codes. *J Artif Intell Res* 2:263–286
25. Broomhead DS, Lowe D (1988) Radial basis functions, multi-variable functional interpolation and adaptive networks. Royal Signals And Radar Establishment Malvern (United Kingdom)
26. Shannon CE (1949) Communication in the presence of noise. In: Proceedings of IRE, vol 37, pp 10–21
27. Shannon CE (1948) A Mathematical theory of communication. *Bell Syst Tech J* 27:379–423
28. Emmanuel CJ (2006) Compressive sampling. In: Proceedings of ICM, vol 3, pp 1433–1452
29. Baraniuk RG (2007) Compressed Sensing [Lecture Notes]. In: Proceedings of SPM, vol 24, pp. 118–124
30. Eleyan A, Kose K, Cetin E (2013) Image feature extraction using compressive sensing. In: Proceedings of AISC, pp 177–184
31. Ashir AM, Eleyan A (2017) Facial expression recognition based on image pyramid and single branch decision tree. In: Proceedings of SIVIP, vol 9, no 1, pp 1–8
32. Kanade T, Cohn JF, Tian Y (2000) Comprehensive database for facial expression analysis. In: Proceedings of FG2000-ICAFGR, Grenoble, France, pp 46–53
33. Valstar M, Pantic M (2010) Induced disgust, happiness and surprise: an addition to the MMI facial expression database. In: Proceedings of international conference on language resources and evaluation, workshop on EMOTION, pp 65–70
34. Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z, Matthews I (2010) The Extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition—workshops, pp 94–101
35. Wei-Lu C, Jian-Jiun D, Jun-Zuo L (2015) Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection. *Int J Signal Process* 117:1–10
36. Ying Z, Fang X (2008) Combining LBP and Adaboost for facial expression recognition. In: Proceedings of ICSP, pp 1461–1464
37. Guo G, Dyer R (2007) Facial expression recognition based on Gabor histogram feature and MVBoost. *J Comput Res Dev* 44:1089–1096
38. Huang MW, Wang ZW, Ying ZL (2010) A new method for facial expression recognition based on sparse representation plus LBP. In: Proceedings of ICISP, pp 1750–1754
39. Cai L, Yin Z (2009) A new approach of facial expression recognition based on Contourlet transform. In: Proceedings of ICWAPR, pp 275–280
40. Zavaschi T, Koerich A, Oliveira L (2011) Facial expression recognition using ensemble of classifiers. In: Proceedings of IC-ASSP, pp 1489–1492
41. Shan C, Gong S (2011) Facial expression analysis across databases. In: Proceedings of MT, pp 317–320
42. Zhang Z, Xu C, Wang JX, Chen XN (2012) Facial expression recognition based on MB-LGBP feature and multi-level classification. *J Adv Intell Soft Comput* 129:37–42
43. Moeini A, Faez K, Sadeghi H, Moeini H (2016) 2D facial expression recognition via 3D reconstruction and feature fusion. *J Vis Commun Image R* 35:1–14
44. Rivera AR, Castillo JR, Chae O (2013) Local directional number pattern for face analysis: face and expression recognition. *IEEE Trans Image Process* 22(5):1740–1752
45. Liu S, Ruan Q, Wang C, An G (2012) Tensor rank one differential graph preserving analysis for facial expression recognition. *Image Vis Comput* 30:535–545
46. Shan C, Gong S, McOwan PW (2009) Facial expression recognition based on local binary patterns: a comprehensive study. *Image Vis Comput* 27(6):803–816
47. Gu W, Xiang C, Venkatesh YV, Huang D, Lin H (2012) Facial expression recognition using radial encoding of local Gabor features and classifier synthesis. *Pattern Recognit* 45:80–91
48. Yeasin M, Bullot B, Sharma R (2004) From facial expression to level of interest: a spatio-temporal approach. In: Proceedings of CVPR, pp 922–927
49. Aleksic PS, Katsaggelos AK (2006) Automatic facial expression recognition using facial animation parameters and multi-stream HMMS. In: Tran. IFS, pp 3–11
50. Li ZS, Imai J, Kaneko M (2010) Facial expression recognition using facial-component-based bag of words and PHOG descriptors. In: Proceedings of IMT, pp 1003–1009
51. Shan C, Gong S, McOwan PW (2005) Robust facial expression recognition using local binary patterns. In: Proceedings of ICIP, pp 370–373
52. Zhao GY, Pietik M (2007) Dynamic texture recognition using local binary patterns with an application to facial expressions. In: Tran. PAMI, pp 915–928
53. Bartlett MS, Littlewort G, Fasel I, Movellan R (2003) Real-time face detection and facial expression recognition: development and application to human computer interaction. In: Workshop on HCI-CVPR
54. Littlewort G, Bartlett M, Fasel I, Susskind J, Movellan J (2004) Dynamics of facial expression extracted automatically from video. In: Workshop on face processing in video

55. Tian Y (2004) Evaluation of face resolution for expression analysis. In: Workshop on face processing in video
56. Rudovic O, Pavlovic V, Pantic M (2012) Multi-output Laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. In: Proceedings of international conference on CVPR, pp 2634–2641
57. Zhong L, Liu Q, Yang P, Liu B, Huang J, Metaxas DN (2012) Learning active facial patches for expression analysis. In: Proceedings of international conference on CVPR, pp 2634–2641

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.