Received: March 22, 2020. Revised: April 20, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Recognition of Facial Emotions Relying on Deep Belief Networks and Quantum Particle Swarm Optimization

Kamal A. El Dahshan[1] Eman K. Elsayed[2] Ashraf Aboshoha[3] Ebeid A. Ebeid[1]*

[1]*Department of Mathematics, Faculty of science, Al-Azhar University, Cairo, Egypt*
[2]*Department of Mathematics, Faculty of science (Girls), Al-Azhar University, Cairo, Egypt*
[3]*National Centre for Radiation Research and Technology, Atomic Energy Authority, Nasr City, Cairo, Egypt*
* Corresponding author's Email: ebeidali78@yahoo.com

Abstract: Classification of human face images into emotion categories is a hard challenge. Deep learning is mostly effective for this task. In this paper, we propose a facial emotions recognition system based on Deep Belief Networks (DBNs) and Quantum Particle Swarm Optimization (QPSO). The proposed methodology is composed of four phases: first, the input image is preprocessed by cropping the Region Of Interest (ROI) in order to get the desired region and discard non-significant parts. Second, the ROI is divided into many blocks and then the integral image is utilized to determine the superior (most efficient) blocks. Third, image down sampling algorithm is adapted to reduce the size of the new sub image in order to improve the system performance. Fourth, the emotion's class is identified using the DBN. Instead of adapting DBN parameters manually, QPSO is used to automatically optimize DBN parameters' values. The proposed algorithm has been applied to the Japanese female facial expression (JAFFE) and FER-2013 datasets. By applying the proposed algorithm, the computation time is reduced effectively by about 62% on the whole JAFFE and 82% on the whole FER-2013 datasets. Whereas, the accuracy is retained at 97.2% and 68.1% on JAFFE and FER-2013 datasets respectively.

Keywords: Emotion recognition, Deep learning, Deep belief networks (DBNs), Integral image, Quantum particle swarm optimization (QPSO).

1. Introduction

Two decades ago, the facial emotion (expression) recognition (FER) has become one of the hottest disciplines in computer science. Many FER based applications have risen. They include - but are not limited to- human-computer interaction (HCI), animation, computer vision and pattern recognition [1-3]. Indeed, the recognition of facial expressions is a difficult problem for machine learning techniques as long as people don't show their expressions in the same way. Deep learning is a new area of research under the machine learning umbrella which can classify human faces' emotions into many classes effectively. The research in facial expression analysis is basically relaying upon the detection of basic emotions [13]; happiness, anger, disgust, sadness, fear and surprise. Any FER system may be composed of three phases: First, facial emotions detection; this includes detecting facial components (e.g., eyes, mouth and nose) or some key points from the facial image. Second, facial expressions extraction: variety of methods are used to extract temporal or spatial features such that: Principal Component Analysis(PCA), Local Binary Pattern(LBP), Scal-invariant feature transform (SIFT), etc. Third, emotion classification: FER is accomplished by applying an expression classifier such as support vector machine (SVM), K-Nearest Neighbour (K-NN) and AdaBoost to identify the expression's class [3]. Most of these works focus on the accuracy and ignore the computation time which is an important factor especially when we handle large dataset. Thus, many algorithms consume large time through the training process. This drawback was the motivation to propose an accurate and fast algorithm. The major contributions of this work are: 1) Reducing the computation time. This is done by selecting the blocks in the emotion image based on the integral image and then applying image down sampling algorithm to reduce image size. 2) Adapting DBN structure parameters automatically by employing quantum particle swarm optimization to ameliorate the results. This paper is organized as follows: in the next section, related works are shown. Section 3 provides details of Deep Belief Networks and Quantum Particle Swarm Optimization. The methodology of this work is described in section 4. Section 5 shows the experimental results and finally section 6 presents the paper's conclusion.

2. Literature survey

Many approaches have been proposed to overcome the different challenges in Facial emotions recognition (FER) systems. These approaches may be categorized into two groups, conventional FER and deep learning approaches. Conventional approaches are mainly depending on the manual feature engineering. After image pre-processing, the procedure in such approaches extracts the facial features then classifies the test image into the corresponding label. The feature extraction approaches that are frequently used in FER systems focus on extracting appearance features, geometric features, or a hybrid of geometric and appearance features. These approaches include Local Binary Pattern (LBP) [56], Haar-like feature [63], Directional Ternary Pattern (DTP) [6], Local Phase Quantization (LPQ) [7], Gabor descriptor [15, 16] and Principal Component Analysis (PCA) [2]. After feature extraction, the facial expression can be classified to the correct label by applying the proper classifier. Support Vector Machine (SVM) [2,48], KNN algorithm [14, 15, 16], Naive Bayes classifier [50,51], An AdaBoost [1,52,53] and Sparse Representation-based Classifier(SCR) [7,54] are the common classifiers used with the conventional approaches. Deep learning techniques perform better than conventional ones in many machine learning tasks including [49] target detection, identification and classification. In addition, they are convenient to handle large scale data in effective manner [53]. Deep learning approaches composed of: Convolutional Neural Network (CNN), Deep Belief Network (DBN), Recurrent neural network (RNN) and Generative Adversarial Network (GAN). Convolutional Neural Network (CNN) is simply neural network that employs convolution in place of general matrix multiplication in one or more of their layers. CNNs convert data pattern into more complex patterns using smaller and simpler patterns. CNNs based algorithm has been proposed in [11] for FER. The architecture of the network is fine-tuned with Visual Geometry Group model (VGG) to improve results. A deep learning approach based on attentional convolutional network was proposed. The approach focuses on important parts of the face by using a visualization technique [8]. Authors in [12] employ CNN to recognize Facial action units (AUs) to produce the seven basic emotion states. The work [55] introduced two versions of ACNN (Convolution Neutral Network with Attention mechanism) to combine local patches with global images. Deep Belief Network (DBN) is a probabilistic generative model which is composed of multiple layers of stochastic and latent variables [32]. It is based on Restricted Boltzmann Machine (RBM) and its feature extraction of input signal is unsupervised and abstract. By combining DBN with other components, it has proved to be an effective FER approach. Yang [37] proposed a deep FER system. The proposed system trains the Restricted Boltzmann Machine (RBM) network using the training sample set to obtain its probability distribution model. A Boosted Deep Belief Network (BDBN) has been proposed in [29] to perform FER iteratively. A set of effective features can be learned and selected to form a boosted strong classifier in a statistical way. The LBP [36], local ternary pattern (LTP) [9] and The Local directional position pattern (LDPP) [30] algorithms have been utilized to extract features for FER systems. Then, the extracted features have been trained using DBN. Recurrent neural network (RNN) is a type of neural networks where the output from previous step are fed as input to the current step. Training of RNN is very difficult task. But it can be used with convolutional layers to extend the effective pixel neighborhood. In addition, it remembers each information through time which useful in time series prediction, this is called Long Short Term Memory (LSTM). The LSTM has been used with modified versions of CNNs to learn and classify the features for FER systems [4, 24, 57]. Generative Adversarial Network (GAN) has been proposed as a form of generative model for an unsupervised learning composed of a generative and a discriminative Miscellaneous application [59]. A multi-task GAN-based learning approach has been proposed for multi-view FER [60]. The research [61] proposed an end-to-end GAN-based model. the proposed model can automatically generate face images with arbitrary expressions and head- poses. GANs have been utilized with CNN to train the

Received: March 22, 2020. Revised: April 20, 2020.

generator of six basic expressions from the facial image in [62].

3. The basic components in the proposed model

The proposed model is relying on DBN and QPSO algorithms. The structure of such algorithms has been explained as follows:

3.1 Deep belief networks (DBNs)

Hinton proposed a fast and greedy algorithm that can learn deep, directed belief networks; one layer at a time. The top two layers form an undirected associative memory [31]. DBN is a probabilistic generative model composed of multiple layers of stochastic and latent variables [32]. The two most significant properties of deep belief nets are implementing an efficient, layer-by-layer learning procedure and the inference required for forming a percept is both fast and accurate. Figure 1shows generative model of DBN with one visible layer and three hidden layers' network. The generative network generates candidates from input data while the discriminative network evaluates them. It can be adopted to generate unique and realistic facial images and other.

Learning the probability distribution of the input data distinguishes DBNs. In DBNs, the Restricted Boltzmann Machine (RBM) has the ability to represent data features, thus it is used to build their basic structure. RBM has two layers; a visible layer and a hidden one. Figure 2 shows the basic RBM model.

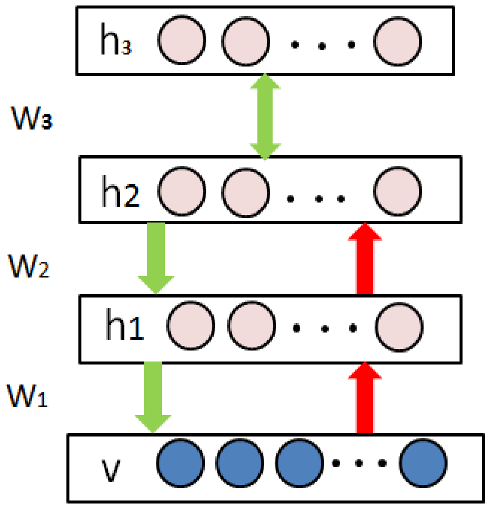

Figure. 1 Generative model of DBN with 1 visible and 3 hidden layers

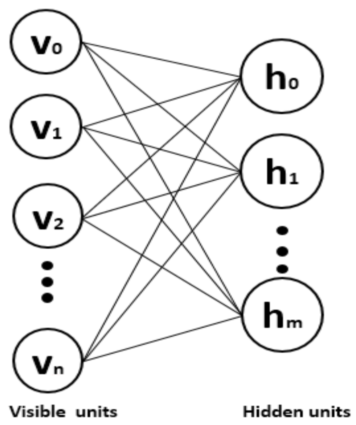

Figure. 2 Basic model of Restricted Boltzmann machine

In RBM with one hidden layer, consider n units in the visible layer v and m units in the hidden layer h; the energy function is given by [33]:

$$E(v.h) = -\sum_{i.j} v_i h_j w_{ij} - \sum_i a_i v_i - \sum_j b_j h_j \quad (1)$$

where w_{ij} is the connection weight and a_i, b_j are the biases of two layers. $i=1, 2, ...n, j=1, 2,..., m.$ According to the energy function, the joint probability $p(v, h)$ of visible and hidden variables v, h is assigned by the following equation:

$$p(v.h) = \frac{1}{z}\sum_h e^{-E(v.h)} \quad (2)$$

Where $z = \sum_{v.h} e^{-E(v.h)}$ refers to the partition function. Among all hidden units, the activation status is conditionally independent. Thus, the activation probability of the j-th hidden unit may be expressed as:

$$p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}) \quad (3)$$

In the same time, the activation probability of the i-th visible unit is:

$$p(v_i = 1|h) = \sigma(a_i + \sum_j w_{ij} h_j) \quad (4)$$

σ is the sigmoid function. According to [39], the sigmoid has the form:

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (5)$$

Learning RBM parameters can be accomplished efficiently by maximizing the log-likelihood for the training data using the gradient ascent [39]. Gibbs

Received: March 22, 2020. Revised: April 20, 2020.

sampling approach was first used to achieve maximum likelihood estimation. It should be used iteratively in order to achieve a better approximation especially when it is used to learn a huge amount of objects. In the training procedure, Gibbs sampling method utilizes gradient ascent. Gradient updating follows the formula:

$$\frac{\partial \log p(v)}{\partial w_{ij}} = < v_i h_j >_{data} - < v_i h_j >_{reconst} \quad (6)$$

Where $< v_i h_j >_{data}$ is the obtained distribution by input data vector and $< v_i h_j >_{reconst}$ represents the expected distribution specified by RBM model [34, 36]. *Contrast divergence (CD)* has been proposed by Hinton in order to increase learning speed of RBM. The *CD* algorithm reconstructs the training sample distribution by replacing an approximate model of the original Gibbs sampling. It improves the RBM model training efficiency. The *CD* Parameters may be updated as follows [37]:

$$\Delta w_{ij} = \gamma(< v_i h_j >_{data} - < v_i h_j >_{reconst}) \quad (7)$$

$$\Delta a_i = \gamma(< v_i >_{data} - < v_i >_{reconst}) \quad (8)$$

$$\Delta b_j = \gamma(< h_j >_{data} - < h_j >_{reconst}) \quad (9)$$

Where γ is the learning rate [35, 39]. The updated weights are computed according to:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} \quad (10)$$

In order to speed up training, Fine-tuning DBNs (refers to transfer learning) will be applied. In transfer learning, knowledge gain during training in a task is utilized to train another related one. During transfer learning, last few layers of the trained network can be replaced with fresh layers for targeted process. The fine-tuned learning is more accurate and less costly than learning from scratch [38]. A single layer of a back Propagation (BP) network is sufficient for fine-tuning.

3.2 Quantum particle swarm optimization

Particle Swarm Optimization (PSO) is a stochastic optimization; it can simulate the behavior of some animal societies such as: Schools of fishes and flocks of birds [20]. The QPSO algorithm forces the particles to mimic the behavior of particles in the quantum mechanics space [19]. The advantage of QPSO is that it has a smaller number of parameters in the adaptation process than the PSO [21]. The particle in the quantum space uses the wave function Ψ. The square of the wave function $|\Psi|^2$ represents the probability density of particles to appear in some position. The probability density function satisfies the following normalization condition:

$$\int_{-\infty}^{+\infty} |\Psi|^2 \, dxdydz = 1 \quad (11)$$

Assume that each particle has D dimensions in quantum space which consists of n particles (population). Thus the position of any particle is $x = (x_1 . x_2 . \cdots . x_D)$; By applying stochastic simulation of Monte Carlo standards on Eq. (11), the position of the particle can be obtained as follows:

$$x = q + \frac{L}{2} \cdot \ln(\frac{1}{u}) \quad (12)$$

Whereas q refers to the local attractor. The parameter u is a random number uniformly distributed between 0 and 1 as in [22, 23]. In the PSO algorithm, each particle has its own local attractor which has components in D dimensions $q = (q_{1,} q_{2,\ldots,} q_D)$. The local attractor guarantees the convergence of the PSO to the correct position. It can be described for the *i*th particle by:

$$q_{id} = \alpha \cdot pbest_{id} + (1-\alpha) \cdot gbest_d \quad (13)$$

Where $d=1, 2, \ldots, D$ and $i=1, 2, \ldots, n$. Both *pbest* and *gbest* are the best position (in the current iteration) and global best positions (the best position in all previous iterations) [58]. α is a random number in the interval [0, 1]. The variable L in Eq. (12) will be computed as the formula $L=2\beta \cdot |mbest_{id} - x_{id}|$. Where the parameter β is called the "*Contraction-Expansion Coefficient*" [20] and *mbest* is the mean value of the best positions of all particles. It was proposed to avoid the premature convergence. Hence, the position of each particle can be updated according the equation:

$$x_{id}(t+1) = q_{id} + \beta \cdot |mbest_d - x_{id}(t)| \cdot \ln(\frac{1}{u}) \quad (14)$$

According to Eq. (14), the particle's position in the next iteration depends on the local attractor, average (mean) positions and particle's position in the current iteration. The fitness function is used to choose the optimum values (which may be *pbest* or *gbest*). The form of fitness function depends on the problem that QPSO solves.

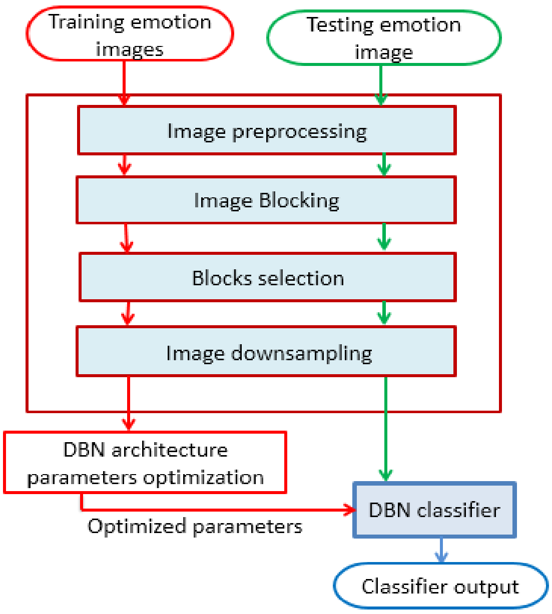

Figure. 3 Proposed algorithm for emotion recognition

4. Methodology

The different steps of the proposed algorithm are image preprocessing, image blocking, blocks selection, image down sampling, and finally emotion classification, see Fig. 3.

The details of the proposed algorithm are described hereafter:

- Image preprocessing: in this step, Viola and Jones algorithm [5] is adapted to detect the ROI then the redundant regions at the image boundary are removed.
- Image blocking: the ROI is divided into a number of blocks (sub regions). All blocks have the same size and dimensions.
- Blocks selection: in order to select the superior blocks, the integral image [17, 18] is computed independently for each block to produce a matrix *G*. Then, for every block, the last value *V* in the matrix *G* is used to decide whether that block will be selected or rejected. Therefore, the blocks that have a small value of *V* are the best. The integral image is fast and efficient for generating the sum of values in a rectangular subset of a grid.
- Image down sampling: in order to speed up the computations and enhance the accuracy, down sampling technique is used to reduce the size of the image. Computations speed depends on number of nodes in the visible layer which depends on the number of pixels in the input image.
- DBN based classification: classification process is accomplished by using the DBNs classifier. The input image' values are received by the first layer of RBM and each value represents a node in the visible layer. During the DBN learning process, many parameters of the DBN architecture require the setting up and configuration. The number of the *hidden units n*, the *weight decay* λ, the *learning rate* γ and the *momentum* φ are the important parameters. Instead of adapting these parameters manually, QPSO can effectively perform this task. The general procedure of QPSO is as follows: 1) Parameters' values are initialized. 2) The fitness function is calculated for each particle. 3) Local and global bests are updated. 4) Parameters' values are updated according to Eq. (14) above. 5) Repeat steps from 2 to 4 until stopping condition is met. In order to speed up the evaluation process, many methods were proposed [40, 41, and 42]. A hybrid algorithm has been constructed as follows: first, a partial dataset is selected randomly to perform parameters optimization. Second, the stability *(S)* of candidate solutions (candidate parameters' values) is computed according to the following equation:

$$S = {}^{\mu}/_{\sigma}, \tag{15}$$

Where $\mu = \frac{\sum_i^N C_i^k}{N}$ and $\sigma = \sqrt{\frac{\sum_i^N (C_i^k - \mu)^2}{N}}$

C is the accuracy classification obtained by the candidate parameters, while *N* is the number of solutions. Third, if the solutions' performances are relatively stable, then there is no need to continue the training process and thus cut the current epoch. The selected set of parameters is sent to the DBN classifier to classify the images of the test set.

5. Experimental results

The proposed system's training and testing has been accomplished on two popular datasets. The first is the Japanese female facial expression (JAFFE) dataset. The JAFFE dataset had been developed by Kyushu University (Japan) in 1997. It consists of 213 images picked up for 10 Japanese female subjects. Each female has 7 facial expressions which are happiness, sadness, anger, disgust, fear, surprise and neutral [25, 26]. Every expression is implemented by three samples for all subjects (total 10 subjects). Thus, 210 images have been used from the dataset. Fig. 4 (a) shows some samples from the JAFFE dataset. The second dataset is FER-2013 which was collected using the Google

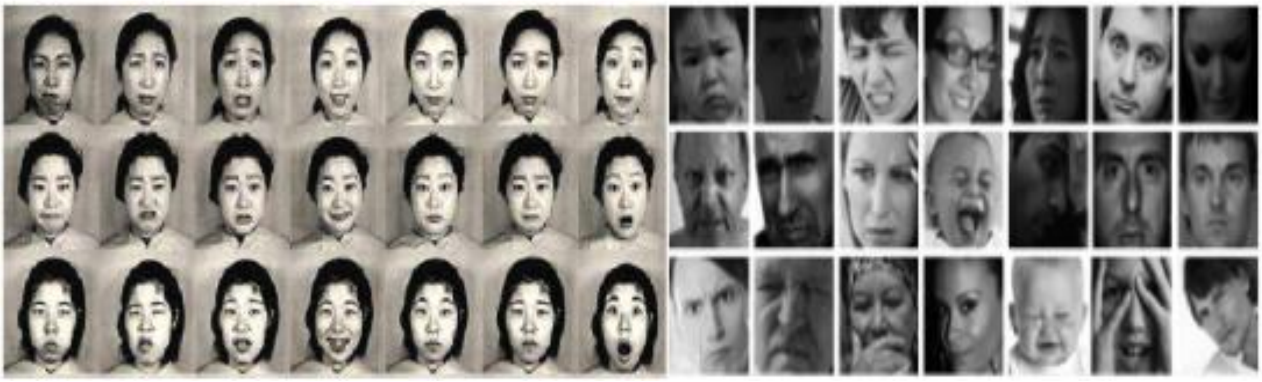

Figure. 4 Dataset: (a) samples from JAFFE dataset [43] and (b) samples from FER-2013 dataset [47]

Table 1. The accuracy and the time for different layers on JAFFE dataset

Alg. \ layer	2		3		4		5		6		7	
	Acc.	T	Acc.	T	Acc.	T	Acc.	T	Acc.	T	Acc.	T
Nearst	90	2.15	93.0	2.3	97	2.6	93.2	2.9	93	3.1	93.0	3.4
Bilinear	93	2.20	**97.0**	**2.27**	94	2.7	94.3	2.8	93	3.0	93.0	3.5
Bicubic	93	2.21	94.0	2.4	94	2.7	96.0	3.0	96	3.3	94.0	3.6
Lanczos2	90	2.15	95.0	2.6	92	2.8	94.6	3.0	92	3.2	93.0	3.5
DWT	93	2.20	92.7	2.4	93	2.7	93.0	3.0	93	3.2	93.4	3.4

image search API research project. This dataset consists of 35,887 gray scale images of 48 × 48 pixels [43]. Faces registration has been done automatically so that many faces were not centered and occupy different positions in the image. They might have low contrast or face occlusion as in Fig. 4 (b).

The original scale of any image in the JAFFE dataset is 256 × 256 pixels. Table 1 shows the accuracy and computation time after applying different down sampling algorithms. The results of using nearest, bilinear, bicubic, Lanczos2 interpolation and DWT have been compared. The time in all the tables indicates the time required for image processing till DBN training process. We introduce a new parameter "performance" defined as follows:

$$Performance = (accuracy / time) \times bf \qquad (17)$$

Where bf is a balance factor and $bf \in \{\ldots; 0.4; 0.6; 0.8; 1; 1.2; 1.4; 1.6; \ldots\}$.

The training set consists of 28,709 images, while each of the validation and test set includes 3,589 images. In this paper, the accuracy is computed according to the following formula [27]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (16)$$

Where TP, TN, FP and FN are true positive, true negative, false positive and false negative respectively.

In Table 1, The nearest interpolation at 4 layers DBN and bilinear interpolation at 3 layers DBN have the best accuracy. The bf is chosen to equal 1 and the resized images are 40 × 60 pixels for all down sampling algorithms. In order to differentiate between down sampling algorithms, the accuracy and computation time are considered to compute the performance as in Eq. (17). Thus, the performance of Bilinear interpolation is 42.7 while it equals 37 in the case of using Nearest interpolation as reported in Table 2. Therefore, the Bilinear interpolation is chosen to perform image down sampling in JAFFE dataset. The performance of different down sampling algorithms is shown in Fig. 5.

Table 2. The performance at different number of layers in JAFFE dataset

layers \ Alg.	2	3	4	5	6	7
Nearst	41.8	40.4	37	32.1	30	27.3
Bilinear	42.2	**42.7**	39	34.0	31	26.5
Bicubic	42.0	39.1	35	33.3	29	26.0
Lanczos2	41.8	36.5	33	31.5	29	26.6
DWT	42.3	38.5	34	31.0	29	27.5

Table 4. The accuracy and the time at different number of layers in FER-2013

Alg. \ layers	2		3		4		5		6		7	
	Acc.	T	Acc.	T	Acc.	T	Acc.	T	Acc.	T	Acc.	T
Nearst	65.5	77	**68**	**71**	67	75	63	76	63	77	63	79
Bilinear	66	79	66.5	78	65	79	64	80	66	82	65	84
Bicubic	63	81	66	78	65	81	65	82	65	83	65	85
Lanczos2	64	80	65	78	66	80	65	81	63	83	65	84
DWT	64.5	78	63	71	65	75	63.3	75	62	77	63	79

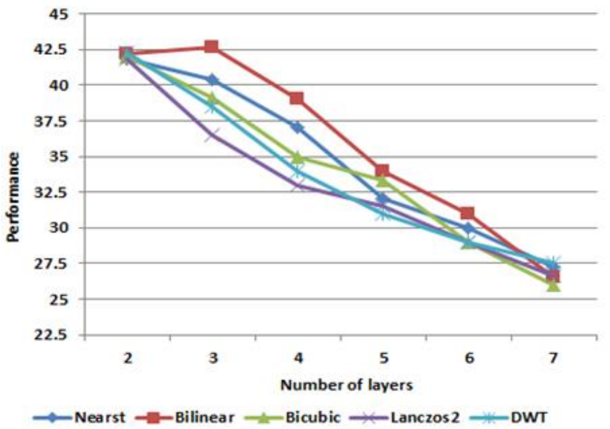

Figure. 5 The performance of down sampling alg. relying on number of layers on JAFFE dataset

Table 3. The accuracy, time and performance at different dimensions on JAFFE dataset

Dimenions	Acc. (%)	Time (sec)	Perforance
71 × 121	96	4.35	22.0
40 × 60	**97**	**2.27**	**51.2**
30 × 30	94	1.90	29.7
20 × 20	94	1.50	37.6
10 × 10	92	1.37	33.5

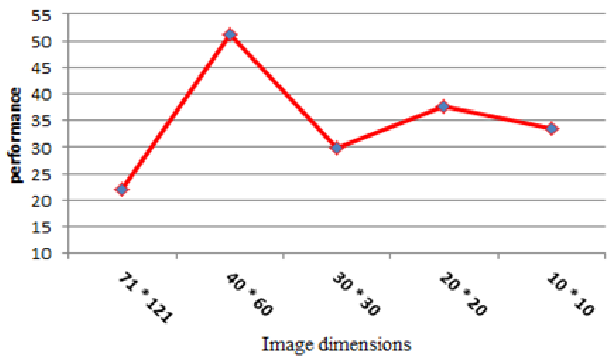

Figure. 6 The performance at different image dimensions on JAFFE dataset

According to table 3, the best image size is 40× 60 pixels at 3 layers DBN based on bilinear down sampling. The accuracy at size 71 × 121 is considered as the base and any increasing in the

Table 5. The performance at different number of layers in FER-2013 dataset

Layers \ Alg	2	3	4	5	6	7
Nearst	.85	**.96**	.89	.83	.81	.80
Bilinear	.84	.85	.73	.80	.80	.77
Bicubic	.78	.75	.80	.79	.78	.76
Lanczos2	.80	.83	.82	.80	.76	.77
DWT	.82	.89	.76	.84	.80	.8

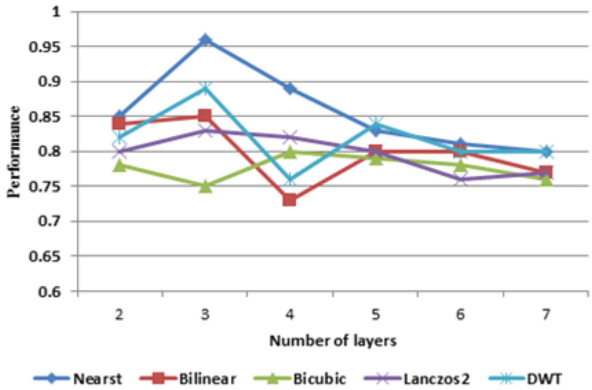

Figure. 7 The performance of down sampling relying on number of layers on FER-2013 dataset

accuracy will increase *bf* by 0.2 and vice versa. The results have been shown graphically in Fig. 6.

Table 4 presents the accuracy and the time of the proposed algorithm on FER-2013 dataset. The nearest interpolation has the best accuracy 68.1% through 71 seconds. All down sampling algorithms have been tested at 20× 20 image size with 3 layer DBN. 10000 images from FER-2013 were utilized in order to compare the performance of the previous algorithms.

The performance was measured as in Table 5. Fig. 7 shows the performance at different layers for image resize algorithms. Increasing the number of DBN layers may slightly increase the accuracy but it consumes more time. Therefore, the performance may decrease despite increasing number of layers.

Table 6. The performance at different dimensions in FER2013 dataset

Dimenion	Acc.	Time (sec)	Performancece
28 48	67.0	1290	.05
30 30	67.0	898	.07
20 20	**68.1**	**490**	**.17**
15 15	66.0	330	.16
10 10	64.0	203	.13

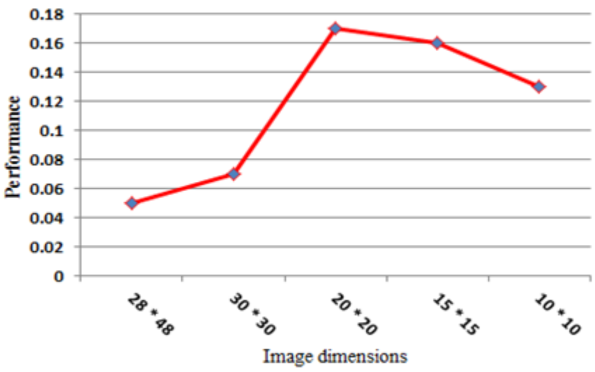

Figure. 8 The performance at different image dimensions on FER- 2013 dataset

Table 6 includes the performance of our system at different dimensions of the image. At the size 20 × 20, the performance has the best value based on nearest interpolation. Fig. 8 shows the results graphically.

QPSO algorithm is utilized to adapt the DBN parameters because it has the advantage over the traditional PSO algorithm especially in computation time [10]. The QPSO parameters are adapted as follows: *Population size* = 20, *Maximum iterations* = 100 and β = 0.75. In the above mentioned DBN experiments, the stopping criterion is either reaching maximum iteration or achieving stability condition as in Eq. (15). The number of neurons (units) in the visible layer must be equal to the number of values (pixels) in the input image. whereas the number of neurons in the next hidden layers is adapted by QPSO to be 1000, 100 and 7. In all experiments just three hidden layers of DBN were constructed. The scope of DBN parameters is obtained as follows: *Number of hidden units* ∈ [100, 2000], *Weight decay* ∈ [0.00001,0.01], *learning rate* ∈ [0.1, 0.9] and *momentum* ∈ [0, 1]. As mentioned before, the proposed algorithm rejects the non-significant blocks and construct the ROI from the selected ones. This forms a new sub image. Since the new sub image is smaller than the original one and its size is reduced, it will consume less time for training. Table

Table 7. The accuracy and time before and after selecting blocks and reducing the image size

Dataset	Before		After	
	Acc.	**Time(s)**	**Acc.**	**Time(s)**
JAFFE	96.0	5.9	97.2	2.27
FER-2013	66.3	2.8×10^3	68.1	490

Table 8. Some of previous works in emotion analysis based on JAFFE dataset

Literature	**Method**	**Classifier**	**Acc.**
[1]	t-SNE	AdaBoost	94.50
[28]	LFDA	KNN	94.30
[9]	LDN	SVM	83.30
[13]	LBP + LDA	SVM	92.22
[29]	BDBN	DBN	93.00
[36]	LBP+ DBN	DBN	87.60
[37]	DBN	DBN	98.75
[6]	DTP	SVM	92.45
Proposed	DBN+QPSO	DBN	97.20

Table 9. Some of previous works in emotion analysis based on FER-2013 dataset

Literature	**Method**	**Classifier**	**Acc.**
[44]	GoogleNet	CNN	65.2
[45]	VGG+SVM	SVM	66.3
[46]	CNN	CNN	66.4
Proposed	DBN+ QPSO	DBN	68.1

7 summarizes the results before and after reducing the size of image.

In JAFFE, 7 images (one image from each emotion for one person) were used for testing and 63 images (one image from each emotion for nine persons) were used for training. Thus, the computed time is the total time required for processing 70 images. It is obvious, the time has been decreased by 62% based on bilinear down sampling and 3 layers DBN. While in FER-2013 dataset, the time refers to the time processing for 32298 images (28709 for training and 3589 for testing). The percentage of decreased time in this case is 82% based on nearest down sampling and 3 layers DBN. Tables 8 and 9 summarize some of the previous works which were implemented on JAFFE and FER-2013 datasets. The major comparison criterion is the accuracy which doesn't depend on the machine specifications. The results in tables 8 and 9 show that the proposed algorithm outperforms the other algorithms with respect to the accuracy. In addition, most of these algorithms don't care about computation time which is important factor. Some of these papers reduce the image size (e.g. paper [6] reduces the size to 150 × 110 in JAFFE) which is slightly speed up the computations. Paper [37] employ DBN with 9 hidden layers which consumes

Received: March 22, 2020. Revised: April 20, 2020.

more time. In our work, the image size is reduced effectively which make the algorithm faster. Also, the DBN parameters are prepared manually in the previous works. Whereas, the parameters are prepared automatically by QPSO in our work. In addition, the proposed evaluation procedure has been designed to cut the current epoch and thus the training time is reduced.

All methods' names in the tables are abbreviated whereas all details have been explained in their original papers. Our algorithm is implemented on HP laptop device with processor Intel ® core(TM) i5-3320M CPU @ 2.60 GHz and RAM 8.00 GB.

6. Conclusion

This paper proposed a facial emotion recognition system based on deep belief networks and quantum particle swarm optimization. The input image is preprocessed then ROI was divided into many blocks. The integral image has been utilized to determine the superior blocks then image down sampling algorithm has been adapted to reduce the size of the new sub image. Bilinear and nearest interpolation have worked well on JAFFE and FER-2013 respectively. Classification of the test emotion image is performed by applying 3 layer DBN to identify the emotion's class. The parameters of DBN have been determined by QPSO during training phase. Experiments and results show that the proposed algorithm possesses a high accuracy level and reduces effectively the required time for computations by about 62% on JAFFE and 82% on FER-2013 datasets. In the future work, we plan to merge another approach to improve the system accuracy. Ontology based approach is an expected one that can add a power in this work. Our system can be extended to work with human computer interaction systems. The extended platform will be implemented on large scale dataset in order to evaluate the system performance.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization of this paper, Kamal, Eman and Aboshosha; methodology, Kamal, Eman, Aboshosha, and Ebeid; the software, Ebeid; writing (original draft), Ebeid; review and editing, Kamal and Ebeid.

References

[1] J. Yi, X. Mao, Y. Xue, and A. Compare, “Facial expression recognition based on t-SNE and adaboostM2”, In: *Proc. of IEEE Conf. on Green Computing and Communications* (*GreenCom*), *Internet of Things* (*iThings/CPSCom*), *Physical and Social Computing*, pp. 1744-1749, 2013.

[2] R. Saini and N. Rana, “A Hybrid Framework of Facial Expression Recognition using SVD & PCA”, *International Journal of Computer Science and Information Technologies*, Vol. 5, No. 5, 2014.

[3] B. C. Ko, “A Brief Review of Facial Emotion Recognition Based on Visual Information”, *Sensors*, Vol. 18, No. 2, p. 401, 2018.

[4] K. Zhang, Y. Huang, Y. Du, and L. Wang, “Facial expression recognition based on deep evolutional spatial-temporal networks”, *IEEE Transactions on Image Processing*, Vol. 26, No. 9, pp. 4193-4203, 2017.

[5] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features”, In: *Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 1, pp. 511-518, 2001.

[6] F. Ahmed and M.H. Kabir, “Facial expression recognition under difficult conditions: A comprehensive study on digital directional texture patterns”, *International Journal of Applied Math and Computer Science*, Vol. 28, No. 2, pp. 399-409, 2018.

[7] Z. Wang and Z. Ying, “Expression recognition based on local phase quantization and sparse representation”, In: *Proc. of the Eighth IEEE International Conf. on Natural Computation* (*ICNC*), pp. 222-225, 2012.

[8] S. Minaee and A. Abdolrashidi, “Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network”, *arXiv preprint arXiv*:1902.01019, Vol.2019, 2019.

[9] D. Kakkar, “Facial Expression Recognition with LDPP & LTP using Deep Belief Network”, In: *Proc. of the 5th IEEE International Conf. on Signal Processing and Integrated Networks* (*SPIN*), pp. 503-508, 2018.

[10] K. A. El Dahshan., E. K. Elsayed, A. Aboshosha, and E. A. Ebeid, “Feature selection for face authentication systems: feature space reductionism and QPSO”, *International Journal of Biometrics*, Vol. 11, No. 4, pp. 328-341, 2019.

[11] A. Fathallah, L. Abdi, and A. Douik, "Facial expression recognition via deep learning", In: *Proc. of the 14th IEEE/ACS International Conf. on Computer Systems and Applications* (*AICCSA*), pp. 745-750, 2017.

[12] M. Mohammadpour, H. Khaliliardali, S. M. R. Hashemi, and M. M. AlyanNezhadi, "Facial emotion recognition using deep convolutional networks", In: *Proc. of the 4th IEEE International Conf. on Knowledge-Based Engineering and Innovation* (*KBEI*), pp. 0017-0021, 2017.

[13] S. L. Happy and A. Routray, "Automatic facial expression recognition using features of salient facial patches", *IEEE transactions on Affective Computing*, Vol. 6, No. 1, pp. 1-12, 2015.

[14] A. K. Dauda and N. Bhoi, "Natural Facial Expression Recognition Using HMM and KNN", *IOSR Journal of Electronics and Communication Engineering*, Vol. 2, no. 16, pp. 61-66, 2016.

[15] S. Zhong, Y. Chen, and S. Liu, "Facial Expression Recognition Using Local Feature Selection and the Extended Nearest Neighbor Algorithm", In: *Proc. of the Seventh IEEE International Symposium on Computational Intelligence and Design* (*ISCID*), Vol. 1, No. 1, pp. 328-331, 2014.

[16] R. Vedantham, L. Settipalli, and E. S. Reddy, "Real Time Facial Expression Recognition in Video Using Nearest Neighbor Classifier", *International Journal of Pure and Applied Mathematics*, Vol. 18, No. 9, pp. 849-854, 2018.

[17] F. C. Crow, "Summed-area tables for texture mapping", *ACM SIGGRAPH Computer Graphics*, Vol. 18, No. 3, pp. 207-212, 1984.

[18] R. R. Damanik, D. Sitanggang, H. Pasaribu, H. Siagian, and F. Gulo, "An application of viola jones method for face recognition for absence process efficiency", In: *Proc. of Journal of Physics Conf.*, Vol. 1007, No. 1, p. 012013, 2018.

[19] D. Yumin and Z. Li, "Quantum behaved particle swarm optimization algorithm based on artificial fish swarm", *Mathematical Problems in Engineering*, Vol. 2014, 2014.

[20] J. Sun, W. Xu, and W. Fang, "Quantum-behaved particle swarm optimization algorithm with controlled diversity", *Computational Science–ICCS*, pp. 847-854, 2006.

[21] S. Li, R. Wang, W. Hu, and J. Sun, "A new QPSO based BP neural network for face detection", *Fuzzy Information and Engineering*, Vol. 40, pp. 355-363, 2007.

[22] Z. Huang, "Improved quantum particle swarm optimization for mangroves classification", *Journal of Sensors*, Vol. 2016, Article ID 9264690, 8 pages, 2016.

[23] R. Misra and K. S. Ray, "Object Tracking based on Quantum Particle Swarm Optimization", In: *Proc. of the 9th* IEEE *International Conf. on Advances in Pattern Recognition (ICAPR)*, pp. 1-6, 2017.

[24] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll, "LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework", *Image and Vision Computing*, Vol. 31, No. 2, pp.153-163, 2013.

[25] P. Li and J. Li, "A facial expression recognition method based on quantum neural networks", In: *Proc. of International Conf. on Intelligent Systems and Knowledge Engineering*, pp. 74-78, 2007.

[26] W. Zheng and C. Liu, "Facial expression recognition based on texture and shape", In: *Proc. of the 25th Wireless and Optical Communication Conf.* (*WOCC*), pp.1-5, 2016.

[27] D. Kaya, "Optimization of SVM Parameters with Hybrid CS-PSO Algorithms for Parkinson's Disease in LabVIEW Environment", *Parkinson's Disease*, Vol. 2019, 2019.

[28] Y. Rahulamathavan, R. C. W. Phan, J. A. Chambers, and D. J. Parish, "Facial expression recognition in the encrypted domain based on local fisher discriminant analysis", *Cognition*, Vol. 4, p. 6, 2013.

[29] P. Liu, S. Han, Z. Meng, and Y. Tong, "Facial expression recognition via a boosted deep belief network", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1805-1812, 2014.

[30] M. Z. Uddin, M. M. Hassan, A. Almogren, A. Alamri, M. Alrubaian, and G. Fortino, "Facial expression recognition utilizing local direction-based robust features and deep belief network", *IEEE Access*, Vol. 5, pp. 4525-4536, 2017.

[31] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets", *Neural Computation*, Vol. 18, No. 7, pp. 1527-1554, 2006.

[32] G. E. Hinton, "Deep belief networks" *Scholarpedia*, Vol. 4, No. 5, p. 5947, 2009.

[33] S. Kamada and T. Ichimura, "An adaptive learning method of Deep Belief Network by layer generation algorithm", In: *Proc. of 10*

IEEE Region Conf. (*TENCON*), pp. 2967-2970, 2016.

[34] M. Fatahi, M. Ahmadi, A. Ahmadi, M. Shahsavari, and P. Devienne, "Towards an spiking deep belief network for face recognition application", In: *Proc. of the 6th International Conf. on Computer and Knowledge Engineering* (*ICCKE*), pp. 153-158, 2016.

[35] P. Zhang, S. Li, and Y. Zhou, "An algorithm of quantum restricted boltzmann machine network based on quantum gates and its application", *Shock and Vibration*, Vol. 2015, 2015.

[36] Y. Wu and W. Qiu, "Facial expression recognition based on improved deep belief networks", In: *Proc. of AIP Conf.*, Vol. 1864, No. 1, p. 020130, 2017.

[37] Y. Yang, D. Fang, and D. Zhu, "Facial expression recognition using deep belief network", *Rev. Tec. Ing. Univ. Zulia*, Vol. 39, No. 2, pp. 384-392, 2016.

[38] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification", *Computers and Electronics in Agriculture*, Vol. 161, pp. 272-279, 2019.

[39] P. Lu, S. Guo, H. Zhang, Q. Li, and Y. Wang, "Research on Improved Depth Belief Network-Based Prediction of Cardiovascular Diseases", *Journal of Healthcare Engineering*, Vol. 2018, 2018.

[40] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification", *IEEE Congress on Evolutionary Computation* (*CEC*), pp.1-8, 2018.

[41] Y. Wang, H. Zhang, and G. Zhang, "cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. Swarm and Evolutionary Computation", *Swarm and Evolutionary Computation*, Vol. 49, pp. 114-123. 2019.

[42] T. Yamasaki, T. Honma, and K. Aizawa, "Efficient optimization of convolutional neural networks using particle swarm optimization", In: *Proc. of the third IEEE International Conf. on Multimedia Big Data* (*BigMM*), pp. 70-73, 2017.

[43] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D. H. Lee, and Y. Zhou, "Challenges in representation learning: A report on three machine learning contests", In: *Proc. of International Conf. on Neural Information Processing*, pp. 117-124, 2013.

[44] P. Giannopoulos, I. Perikos, and I. Hatzilygeroudis, "Deep learning approaches for facial emotion recognition: A case study on FER-2013", *In Advances in Hybridization of Intelligent Methods*, pp.1-16, 2018.

[45] M. I. Georgescu, R. T. Ionescu, and M. Popescu, "Local learning with deep and handcrafted features for facial expression recognition", *IEEE Access*, Vol. 7, pp. 64827-64836. 2019

[46] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks", In: *Proc. of 2016 IEEE Winter Conf. on Applications of Computer Vision* (*WACV*), pp. 1-10, 2016.

[47] Z. Liu, M. Wu, W. Cao, L. Chen, J. Xu, R. Zhang, M. Zhou, and J. Mao, "A facial expression emotion recognition based human-robot interaction system", *IEEE/CAA Journal of Automatica Sinica*, Vol.4, No.4, pp. 668-676, 2017.

[48] H. H. Tsai and Y. C. Chang, "Facial expression recognition using a combination of multiple facial features and support vector machine", *Soft Computing*, Vol. 22, No. 13, pp. 4389-4405, 2018.

[49] Y. Huang, F. Chen, S. Lv, and X. Wang, "Facial Expression Recognition: A Survey", *Symmetry*, Vol. 11, No. 10, p. 1189, 2019.

[50] C. Wang, S. Wang, and G. Liang, "Identity-and Pose-Robust Facial Expression Recognition through Adversarial Feature Learning", In: *Proc. of the 27th ACM International Conf. on Multimedia*, pp. 238-246, 2019.

[51] L. Surace, M. Patacchiola, E. B. Sönmez, W. Spataro, and A. Cangelosi, "Emotion recognition in the wild using deep neural networks and Bayesian classifiers", In: *Proc. of the 19th ACM International Conf. on Multimodal Interaction*, pp. 593-597, 2017.

[52] V. K. Gudipati, O. R. Barman, M. Gaffoor, and A. Abuzneid, "Efficient facial expression recognition using adaboost and haar cascade classifiers", In: *Proc. of 2016 Annual Connecticut IEEE Conf. on Industrial Electronics, Technology & Automation* (*CT-IETA*), pp. 1-4, 2016.

[53] S. Zhang, B. Hu, T. Li, and X. Zheng, "A Study on Emotion Recognition Based on Hierarchical Adaboost Multi-class Algorithm", In: *Proc. Of International Conf. on Algorithms and*

Architectures for Parallel Processing, pp. 105-113, 2018.
[54] Z. Wang and Z. Ying, “Facial expression recognition based on local phase quantization and sparse representation”, In: *Proc. of the 8th IEEE International Conf. on Natural Computation*, pp. 222-225, 2012.
[55] Y. Li, J. Zeng, S. Shan, and X. Chen, “Occlusion aware facial expression recognition using CNN with attention mechanism”, *IEEE Transactions on Image Processing*, Vol. 28, No. 5, pp. 2439-2450, 2018.
[56] W. F. Liu, S. J. Li, and Y. J. Wang, “Automatic facial expression recognition based on local binary patterns of local areas”, In: *Proc. of IEEE WASE International Conf. on Information Engineering*, Vol. 1, pp. 197-200, 2009.
[57] D. H. Kim, W. J. Baddar, J. Jang, and Y. M. Ro, “Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition”, *IEEE Transactions on Affective Computing*, Vol. 10, No. 2, pp. 223-236, 2017.
[58] S. Chen, “Quantum-Behaved Particle Swarm Optimization with Weighted Mean Personal Best Position and Adaptive Local Attractor”, *Information*, Vol. 10, No. 1, p. 22, 2019.
[59] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial nets”, *Advances in Neural Information Processing Systems* , Vol. 2014, pp. 2672-2680, 2014.
[60] Y. H. Lai and S. H. Lai “Emotion-preserving representation learning via generative adversarial network for multi-view facial expression recognition”, In: *Proc. of the 13th IEEE International Conf. on Automatic Face & Gesture Recognition* (*FG 2018*) , pp. 263-270, 2018.
[61] F. Zhang, T. Zhang, Q. Mao, and C. Xu, “Joint pose and expression modeling for facial expression recognition”, In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3359-3368, 2018.
[62] H. Yang, Z. Zhang, and L. Yin “Identity-adaptive facial expression recognition through expression regeneration using conditional generative adversarial networks”, In: *Proc. of the 13th IEEE International Conf. on Automatic Face & Gesture Recognition* (*FG 2018*), pp. 294-301, 2018.
[63] P. Yang, Q. Liu, and D. N. Metaxas, “Boosting encoded dynamic features for facial expression recognition”, *Pattern Recognition Letters*, Vol. 30, No. 2, pp. 132-139, 2009.