Computers and Electrical Engineering 85 (2020) 106640

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Deep Convolutional - Optimized Kernel Extreme Learning Machine Based Classifier for Face Recognition☆

Tripti Goel*, R Murugan

National Institute of Technology Silchar, Assam 788010, India

ARTICLE INFO

Article history:
Received 15 April 2019
Revised 4 January 2020
Accepted 12 March 2020

Keywords:
Deep Learning
Back Propagation
Convolutional Neural Network
Extreme Learning Machine
Kernel Function
Particle Swarm Optimization

ABSTRACT

Face recognition task is an active area of research in recent years in the field of computer vision and biometric. Feature extraction and classification are the most significant steps for accurate face recognition system. Conventionally, eigenface approach or frequency domain features were used for feature extraction, but they are not invariant to outdoor conditions like, lighting, pose, expression and occlusion. In the present work, multiple convolutional and pooling layers of Deep Learning Networks (DLN) will extract efficiently the high level features of the face database. These features are given to the Kernel Extreme Learning Machine (KELM) classifier whose parameters are optimized using Particle Swarm Optimization (PSO). The proposed Deep Convolutional-Optimized Kernel Extreme Learning Machine (DC-OKELM) algorithm leads to better performance results and fast learning speed compared to classification using deep neural networks. The performance of DC-OKELM is evaluated on four standards face database such as AT&T, CMU-PIE, Yale Faces and UMIST. Experimental results are compared with other state of the art classifiers in terms of error rate and network training time which shows the effectiveness of the proposed DC-OKELM classifier.

1. Introduction

Face recognition (FR) has been an progressive research area in the field of biometrics, computer vision, neural networks, image processing, and pattern recognition [1,2]. Now a days, face recognition application has been increasing emphasized for the security purposes like database matching, law enforcement, public security, and identity authentication for voter card, aadhar card or driver license, access control, and intelligence surveillance [3].

Even though, FR has received high level of efficiency up to now, but its accuracy is restricted to only frontal images under normal lighting conditions [4]. It is because of some varying constraint that limits the application area of automatic face recognition system. These variances are illumination variations, facial expressions variances, pose invariance and ageing effects etc. [5–7]. Traditionally, these variations have been normalized using pre-processing techniques and then features are extracted using eigenface or wavelet transform.

However, in recent years, DLN [8] have been recommended and show significant outcomes for image classification. Deep learning network composed of multiple alternate convolutional and pooling layers which can extract much complex and robust features which are invariant to various transformation in images. In deep learning [9], the hidden node's parameters

☆ This paper is for CAEE special section SI-elm. Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Zhihong Man.

* Corresponding author.
E-mail addresses: triptigoel83@gmail.com (T. Goel), murugan.rmn@gmail.com (R. Murugan).

https://doi.org/10.1016/j.compeleceng.2020.106640
0045-7906/

need to be trained using Back Propagation (BP) [10] training. Back propagation training is time consuming and also faces the problem of local minima, overfitting and require large amount of data.

BP uses iterative training to tune the weights and biases of the hidden nodes. Due to iterative training of network parameters, it is computationally complex and time consuming. Because of computational complexity and time consuming nature of classification using back propagation in deep learning, in the proposed work, only features will be extracted from Deep Convolutional Networks (DCN) and then followed by KELM classifier [11]. KELM is a non iterative classifier that does not require training of network parameters. In KELM, kernel function is used at the hidden nodes whose parameters are optimized using PSO. KELM provides better performance results in terms of recognition accuracy and training time for image classification. Therefore, the proposed algorithm DC-OKELM combines the invariant and deep features extraction capability of deep learning and fast training speed of KELM.

Main contributions of the proposed approach are: 1) In the proposed work, feature extraction capability of deep networks and faster speed of KELM classifier is integrated to extract the features which are robust to various outliers like illumination, pose, expression, ageing effects and occlusion and classify them with fast speed. 2) The proposed framework for classification avoids the problem of local minima and over fitting due to gradient descent training of parameters of deep learning. 3) As the kernel parameters play an important role in classification, kernel parameters are optimized using PSO optimization technique and also compared with state of the arts optimization algorithm to achieve best accuracy in minimum time. 4) Comparative analysis have been made with other state of the art deep learning and Extreme Learning Machine (ELM) algorithms, out of which the proposed DC-OKELM offers better classification results with highest accuracy and least training time. Therefore, in the present work, a novel framework is proposed which combines the feature extraction capability of deep networks with KELM classifier whose parameters are optimized for face recognition. Because of better recognition rates and very less training time, this framework can be used for real time face recognition applications.

Rest of the paper is organized as, Section 2 presents a brief review of deep convolutional network and KELM networks. Section 3 introduces the description of the proposed DC-OKELM algorithm for face recognition. Section 4 illustrates the experimental results and discussions in terms of error rate. Finally, the conclusion and future scope of the work has found in Section 5.

2. Related works

Many statistical algorithms have been widely used for successful feature extraction. These methods calculates the set of basis images using training set and then project the test face on these basis images, which reduces the size of database. Thus, such a linear projection selects the most prominent features of the face image in order to distinguish the faces of different persons. Main statistical algorithms for face recognition include Principal Component Analysis (PCA) introduced by Turk and Pentland [12], Linear Discriminant Analysis (LDA) given by Belhumeur [13], Independent Component Analysis (ICA) given by Bartlett [14]. But these methods are sensitive to illumination, expression and pose variations. Other main disadvantage of the statistical model is that they have many assumptions or conditions which should be satisfied for effective classification.

Neural Networks (NN) [15] have arisen as an important tool to overcome the stated problems. NN are inspired by the human brain which has the ability to learn from experience. Neural network is a Multi-Layer Perceptron (MLP) which works as a powerful classifier with minimum pre processing. NN are trained by using BP which requires many arbitrary choices such as number of layers, number and types of nodes, learning rate, number of epochs and stopping criteria.

To overcome the constraints of NN, DLN [16] are used. DLN are the variants of MLP, biologically inspired by cat's visual cortex of simple and complex cells [17]. It is a feed forward network which has the ability of extracting topological properties from unprocessed image that provides partial invariance to translation, rotation, scale and deformation. Therefore, it does not require any pre-processing step to normalize the variations.

DLN are proposed by LeCun et al. [18], that was combined with global training techniques, applied on handwritten character recognition. Lawrence et al. [19] proposed a hybrid NN that combines a Self-Organizing Map (SOM) NN and DLN. Tuning of weights and biases of DLN are done by using standard MLP algorithm. This approach require considerable amount to train SOM and CNN. This algorithm also had high complexity since two different NN were combined to perform the recognition. Simard et al. [20] use elastic distortion and DLN on MNIST database to achieve the good performance results. Requirement of large training size limits the use of this algorithm for recognition. Khalajzadeh et al. [21] proposed a hierarchical structure based CNN which has 4 layers and standard BP for the learning. But, this approach have achieved unsatisfying recognition rate. Zhou et al. [22] proposed stack auto encoders DLN for classification. Gu et al. [23] extracts the variant facial features by combining patch based dictionary learning model and local sparse representation model for single sample per subject. Yu and Li [24] discussed the recent progresses in deep learning like recurrent neural networks and convolutuional networks and stressed on the feature representations. Luo [25], proposed convolutional sparse auto encoder which holds structure of convolutional encoder and sparsify the feature maps using max pooling for feature learning. Shuai et al. [26] proposed Directed Acyclic Graph (DAG) recurrent NN having a more complex architecture having inputs from multiple layers and also outputs to multiple layers. Bertsekas [27] proposed aggregation of feature vectors obtained using NN and deep reinforcement learning. But the main limitation of this algorithm is very time consuming and work well on only small examples.

Features extracted from DLN are classified using ELM [28]. It is a single hidden layer learning algorithm that provides faster learning speed and good generalization performance. In ELM, there is no need of tuning of the weights and biases, they are randomly initialized and generated earlier at the time of training the network. Mapping of input data to feature space in ELM is done by nonlinear piecewise activation function. However, the selection of proper activation function for at the hidden node for a particular dataset is still an unresolved problem. Variants of ELM are: evolutional learning [29], online sequential learning [30], incremental learning [31], ordinal ELM [32], optimized ELM [33], PCA-ELM [34].

Huang [11] used the kernel function for feature mapping of the input data to solve the problem of selection of activation function for hidden nodes with faster learning speed and good recognition results. Because kernel function is used at hidden nodes for feature mapping, this approach is called KELM. Kernel's parameters are also important parameters which should be selected very carefully for efficient face recognition system. Therefore, in the present work, the parameters of kernel function are optimized using PSO to improve the performance results of the proposed classifier.

3. Deep convolutional-optimized kernel extreme learning machine

DLN consists of alternating layers of locally connected convolutional layers, downsampling layers and the fully connected layers for classification. Architecture of deep learning network in shown in Fig. 1. In the proposed work, ResNet-50 [35], [36] pre trained network is used for classification. ResNet-50 is a DLN which is 50 layers deep and can abstract rich feature representations for extensive range of images. Because of so many layers, ResNet-50 can progressively extract more complex and invariant features from the images than other statistical techniques. DCN architecture can achieve invariance to any translations using three concepts, which are: local receptive fields, shared weights and pooling. Each neuron takes its input from a small receptive field from the previous layer and therefore unresponsive to the variations outside of its receptive field. Convolution kernel use the concept of weight sharing at the convolutional layer. Pooling reduces the resolution of the feature map and thus computational complexity for the upper layers.

DLN are hierarchical neural networks that contain alternating convolution and pooling layers. The convolution layer acts as a feature extractor which convolved the fixed size kernels over the outputs from the previous layer to produce its own feature maps. Pooling layer reduces the computational complexity for upper layers and also provides robustness to the variations in translation. These features are then used by optimized-KELM classifier for accurate face recognition. The block diagram of the proposed algorithm is shown in Fig. 2. A brief overview of each layer of DCN is as follows:

Deep Convolutional Neural Networks In convolution layer, kernel filters are applied over an image at all possible offsets. A filter consists of a layer of connecting weights, inputs the rectangular section of the previous layer and outputs the corresponding feature map. Each rectangular section shares the same weights for each neuron and hence reduce the number of learnable parameters. Convolution layer acts as a feature extractor in which previous layer's local receptive fields are convolved with fixed size kernel filters to produce the output. The idea of connecting receptive fields comes from Hubel & Wiesel's discovery of locally sensitive, orientation - selective neurons in the cat's visual system. Once the features have been extracted, its exact location becomes less important, as its position relative to other features is approximately preserved.

Each convolutional layer of the network takes m x n pixel image as an input, convolved with N, k x k size kernel filters, adds one of N scalar bias value followed by an activation function to form the output map. The size of the each output map is given as (m-k+1) x (n-k+1). The main objective of this layer is to perform feature extraction by scanning across the entire input layer, providing some degree of shift invariance. In addition, it also gives advantage of weight sharing by convolving with the same kernel across the input layer.

Convolutional layers are followed by batch normalization layers which normalize the activations and gradients propagating through a network to make training of the network a simple optimization problem. Batch normalization layers are followed by a nonlinear activation function, Rectified Linear Unit (ReLU), which performs the thresholding on each element of the input. During thresholding operation, any value of the input less than zero is set to zero. Each convolution layer is

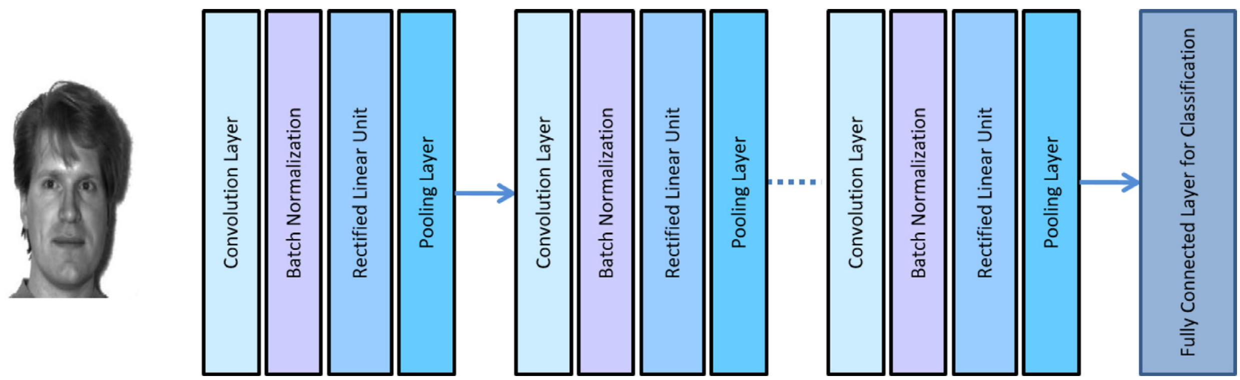

Fig. 1. Architecture of ResNet Deep convolutional network.

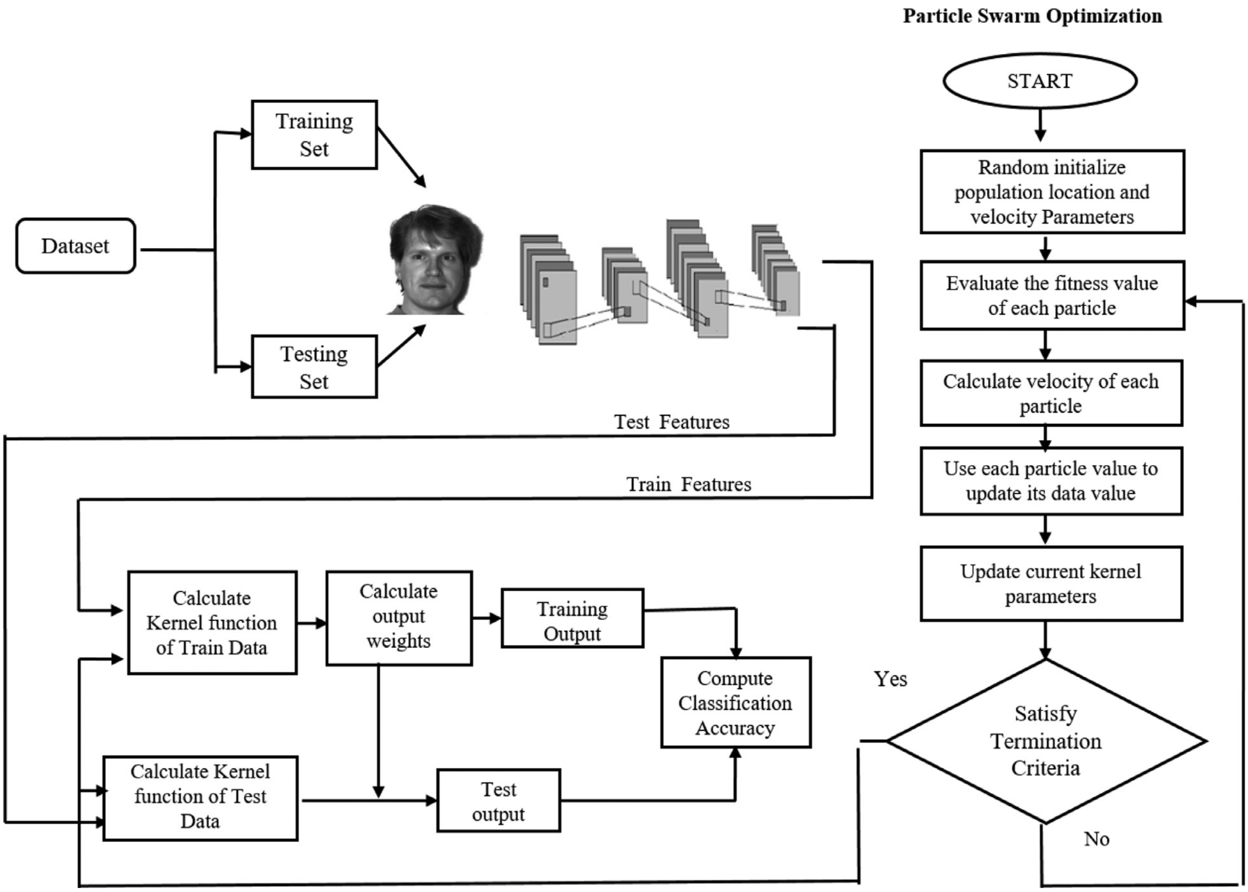

Fig. 2. Block diagram of the proposed DC-OKELM algorithm.

followed by a max pooling layer to reduce the resolution of the feature map. Pooling or down sampling refers to reducing the overall size of the feature map and hence reduce the computational complexity. It also reduces the sensitivity of the feature map's output to shifts and other form of distortions. For N input maps to the pooling layer, there will be exactly N output maps, but size is reduced by a constant factor. Max pooling applies a window function u(n,n) to the input patch, and computes the maximum in the neighborhood. Features extraction is done just before the classification layer of ResNet-50 network, therefore no need of the tuning of the parameters of the DCN. These features are then fed to KELM for classification.

Optimized Kernel Extreme Learning Machine In the present work, the non-iterative learning algorithm, KELM is used for classification. KELM overcomes the limitation of slow learning speed, local minima, large number of hidden neurons, over fitting, of standard BP algorithm. In ELM, the input weights and biases are chosen arbitrarily and weights of the output layer are determined through generalized inverse operation on the output of the hidden layer nodes. This makes the learning very fast and better recognition results as compared to conventional learning algorithms. In KELM, polynomial kernel is used at the hidden layer to calculate the output. Selection of the parameters of polynomial function of KELM is the vital step and therefore PSO is used to select the optimum parameters of KELM.

KELM classifier consists of n input layer's unit, M number of neurons in hidden layer and K output layer's unit. Architecture of single hidden layer KELM is shown in Fig. 3. In KELM, the output weights of the hidden layer after training are given in Eq. (1).

$$H\beta = T \tag{1}$$

Where, β represents the output weights of hidden layer, H denotes the output matrix of hidden layer and T is the desired output for the classifier. From Eq. (1), the output weights of the KELM classifier can be found by using Eq. (2).

$$\beta = H^* T \tag{2}$$

Here H^* denotes the Moore-Penrose inverse of matrix H, which can be calculated using orthogonal projection method, $H^* = H^T(HH^T)^{-1}$. Eq. (2) can be rewritten using Eq. (3) to make the network more stable, by introducing regularization

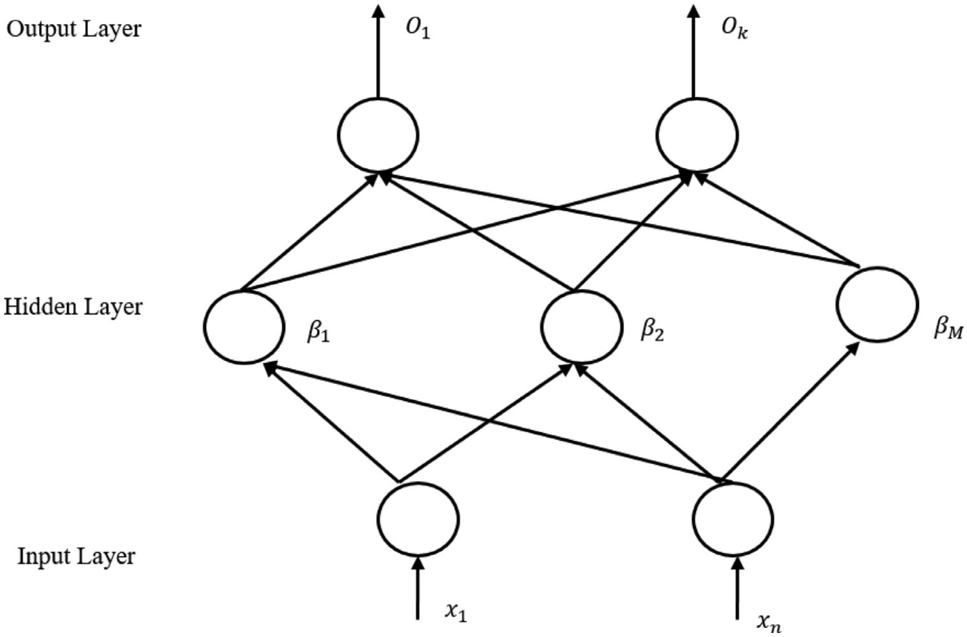

Fig. 3. Architecture of KELM network.

parameter [37] R.

$$\beta = H^T \left(\frac{1}{R} + HH^T \right)^{-1} T \tag{3}$$

The output of ELM for test face is calculated using Eq. (4).

$$y = h(x)\beta = h(x))H^T \left(\frac{1}{R} + HH^T \right)^{-1} T \tag{4}$$

From Eq. (4), based on mercer's condition, kernel matrix is defined as in eq. (5).

$$\varphi_{ELM} = HH^T \qquad \varphi^{ELMi.j} = h(x_i)h(x^j) = k(x_i, x_j) \tag{5}$$

The Eq. (4) and (5) can be rewritten as:

$$y = \begin{bmatrix} k(x, x_1) \\ k(x, x_2) \\ k(x, x_3) \end{bmatrix} \left(\frac{1}{R} + \varphi_{ELM} \right)^{-1} T \tag{6}$$

As interpreted from Eq. (6), output function can be implemented in only single learning step using KELM. Other advantages of KELM are: no need of selecting the hidden layer activation function and number of hidden neurons like in ELM and other neural networks. Additionally, KELM algorithm presents better performance results with faster training speed and more stable network for classification.

In the proposed DC-OKELM classifier, polynomial kernel is used to calculate the output function of the classifier. The expression for polynomial kernel functions is given using Eq. (7)

$$k(x, y) = (x.y^T + a)^b \tag{7}$$

Selection of kernel parameters is the significant part of the KELM classifier. The performance of the KELM depends greatly on the kernel parameters (a, b) of the polynomial kernel function. In the present approach, these kernel parameters are optimized using PSO.

Particle Swarm Optimization PSO [38] is inspired by the group behaviour of bird's flocks or fish schools. In PSO algorithm [39], each individual is termed as particle and entire population is called swarm. Particles are subjected to move in search space according to some simple formulae. These particles have memory which contains their previous state. The movement of particles are guided by their best known position as well as the best position of the any member of the population or swarm. This process is repeated until a satisfactory solution has been reached. The complete optimization procedure using PSO has been found in Fig. 4.

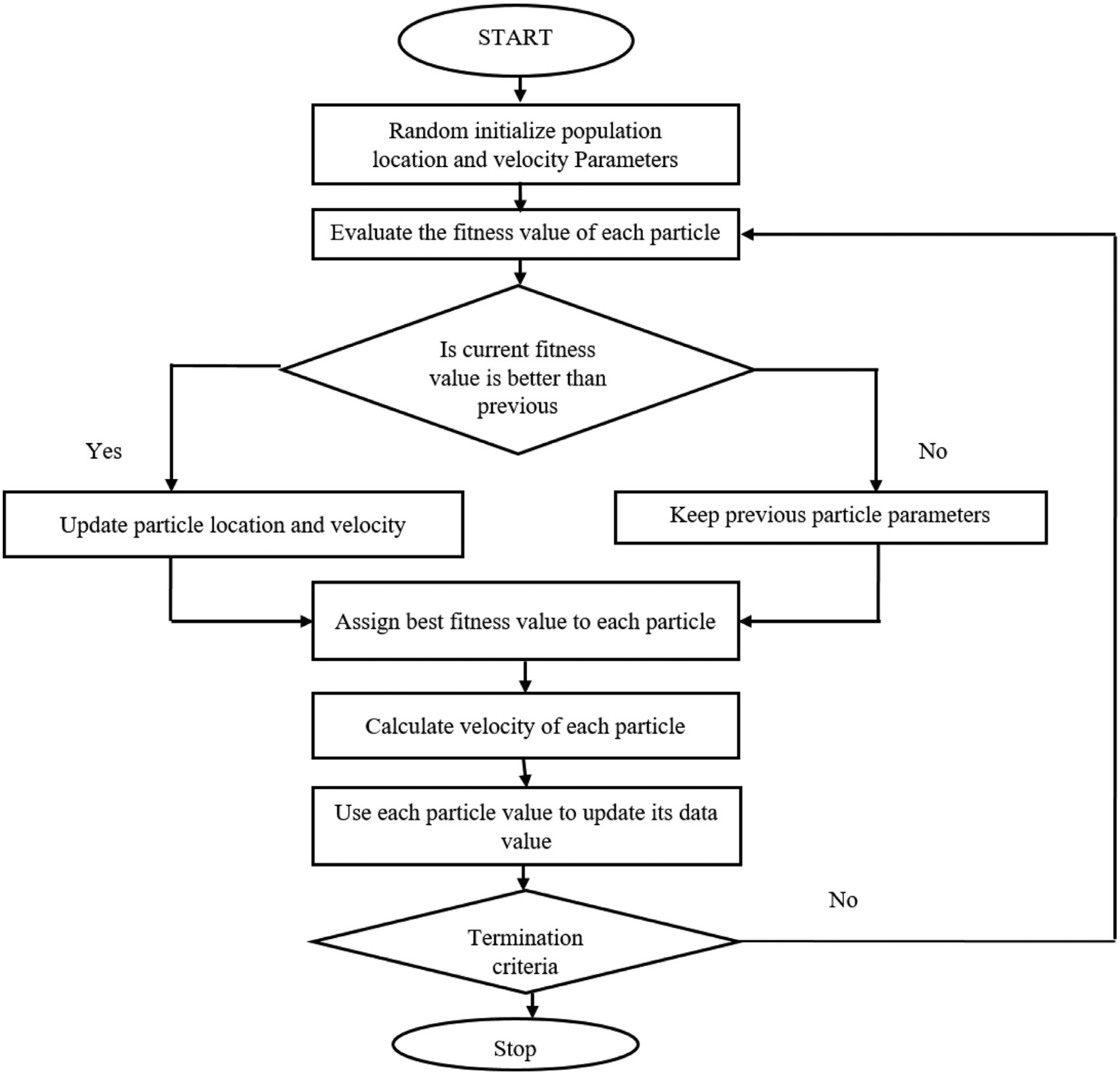

Fig. 4. Flow diagram of optimization using PSO.

4. Experimental results and discussion

This section presents the performance results of the proposed DC-OKELM classifier for face recognition on four benchmark face databases such as AT&T, Yale faces, CMU PIE and UMIST. The present approach is also compared with the other state of art techniques in terms of error rate.

AT&T database AT&T database (https://www.kaggle.com/kasikrit/att-database-of-faces) consists 400 gray-scale images of 40 individuals in.pgm format. Each individual has 10 images. These images includes varying facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). Each image has a resolution of 11292, and 256 gray levels. All the images were taken against a dark homogeneous background, with tilt and rotation up to 20 and scale variation up to 10. Fig. 5 shows images of one subject under all the variations of AT&T database.

For training DC-OKELM classifier, N numbers of images are randomly selected from each individual. There is no any overlapping between training and test sets. As the number of images for each individual increases, error rate will decrease. Table 1 shows the comparison of error rate using different techniques for classification for different number of training images per sample. The proposed classifier for face recognition is compared with ELM classifier on the features extracted using DCN, deep convolution, ELM, KELM, BP, DAG networks, stack encoders and sparse auto encoders. Table 2 shows the

Fig. 5. Sample face images of a same person of AT&T database.

Table 1
Error rate in percentage for AT&T Face database.

Classifier	No. of training images							
	1	2	3	4	5	6	7	8
Proposed DC-OKELM	13.33	0.31	0.35	0.41	0.5	0.62	0.83	0
DC-ELM [40]	16.4	3.4	1.07	2.08	1	0.62	0.83	0
Deep convolution [35]	33.9	27.5	26.07	30	34.5	31.87	15.0	12.5
ELM [28]	29.67	1.75	11.21	10.83	9.1	8.0	6.33	7.25
KELM [11]	32.78	20.62	13.24	12.5	9	8.75	7.5	5.0
BP [10]	31.39	17.18	11.78	12.5	8.5	6.8	10	8.7
DAG [26]	47.7	26.56	19.64	19.58	21.5	21.5	3.3	2.5
Stack Encoder [22]	23.7	23.75	21.78	18.33	12	11.25	12.5	8.75
Sparse Auto encoder [25]	48.05	15.31	9.28	3.75	5.5	2.62	2.5	5.0

Table 2
Comparision of training time in sec. for AT&T Face database.

Classifier	Training time (in Seconds)							
	1	2	3	4	5	6	7	8
Proposed DC-OKELM	38.4	33.48	34.26	34.93	35.22	35.33	35.41	38.0
Deep convolution [35]	231	442	636	822	989	1145	1290	1421
DAG [26]	61.56	104.1	147.5	157.3	157.6	155.4	304.0	309.4
Stack Encoder [22]	1615	1617	1653	1685	1755	1793	1801	1834
Sparse Autoencoder [25]	239.9	455.2	644.4	824.6	992.1	1159	1305	1446

Fig. 6. Sample face images of a same person of Yale Face database.

comparison of training speed of the proposed classifier with classification using DCN, DAG networks, stack encoders and sparse auto encoders.

Yale face Database The Yale database (https://www.kaggle.com/olgabelitskaya/yale-face-database) contains total 165 gray scale images of 15 individuals, 11 images of each in giff format. It does not show any pose variations, but contains illumination variations (e.g. left/right/center light) and facial expressions (smile, sad expression, open/closed mouth). The size of original image is 243/320. Fig. 6 shows the all images of first two subjects under various variations. Table 3 shows the comparison of error rate of state of the art classifiers for different number of training images per sample on Yale Face database. Table 4 shows the comparison in training speed of the proposed method with other deep learning classifiers.

Table 3
Error rate in percentage for Yale Face database.

Classifier	Yale faces							
	1	2	3	4	5	6	7	8
Proposed DC-OKELM	46.07	37.0	9.1	11.42	7.78	8	5	6.67
DC-ELM [40]	46	44.44	21.67	20	11.11	12	11.17	13.3
Deep Convolution [35]	62	68.14	55	39.0476	18.9	32	41.67	17
ELM [28]	40.53	20.29	17.5	13.33	11.778	10.4	7.7	8
KELM [11]	46	28.44	19.47	9.53	11.11	9.33	13.33	11.11
BP [10]	44.67	21.481	19.167	12.38	10	13.3	8.3	6.67
DAG [26]	54	35.5	30.83	26.67	20	26.7	20	15.5
Stack Encoder [22]	55.3	29.6	34.1	28.5	26.7	24.4	24.4	18.4
Sparse AutoEncoder [25]	56.7	40	41.9	28.3	28.3	24.4	18.4	15.5

Table 4
Comparision of training time in sec. for Yale Face database.

Classifier	Training time in sec.							
	1	2	3	4	5	6	7	8
Proposed DC-OKELM	15.5	18.5	13.6	13.6	13.9	14.2	14.0	14.1
Deep Convolution [35]	47.1250	92.72	137.87	184.04	231.39	279.67	325.35	372.26
DAG [26]	38.0	33.74	44.91	52.98	59.6	60.63	76.3	82.8
Stack Encoders [22]	716.5	733.4	743.5	754.3	752.6	774.3	786.8	795.0
Sparse Auto Encoders [25]	54.2	107.1	162.8	219.8	272.9	330.6	384.2	437.0

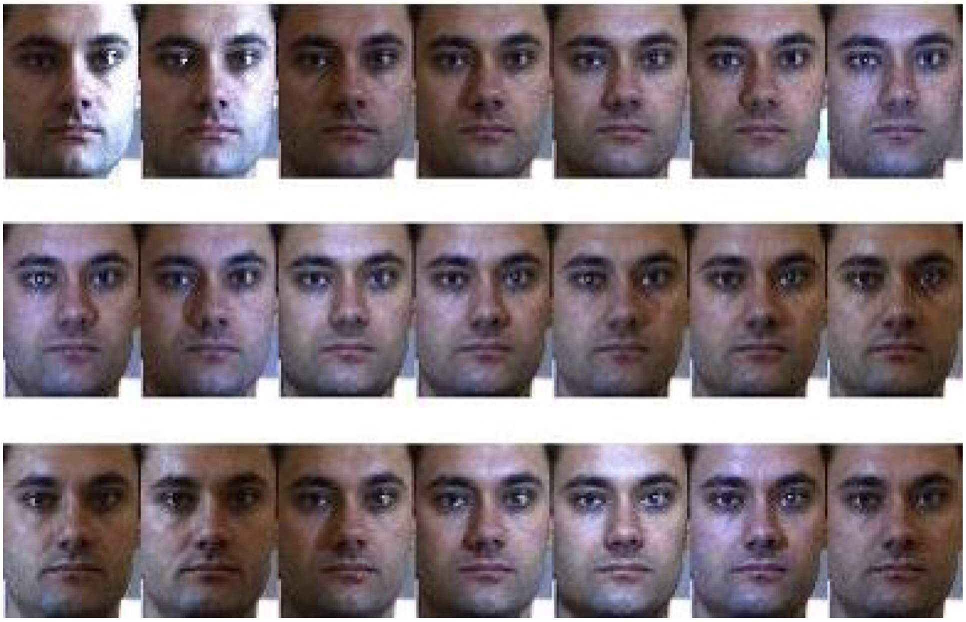

Fig. 7. Sample face images of a same person of CMU PIE database.

CMU PIE database CMU PIE database [41] consists of 68 subjects with total of 41,368 images containing, variations in Pose, Illumination and Expressions (PIE). In the present study, only images with illumination variations are considered. There are 21 images per subject under illumination variations. Fig. 7 shows the example of 21 images of a particular subject of this database. Table 3 shows the comparison of the proposed DC-OKELM classifiers with other classifiers on CMU PIE database. The comparisons are made with respect to number of training image per subject. Table 5 shows the comparison of error rate of different classifier for different number of training images per sample on CMU PIE database. Table 6 shows the comparison in training speed of the proposed method with deep learning classifier.

UMIST Database UMIST database (http://images.ee.umist.ac.uk/danny/database.html) consists of total 564 face images of 20 subjects. The number of images per subject is not fixed, it varies from 19 to 36. The size of each image is 220 x 220 and 256 shades of gray. The files are all in.pgm format. Each subject consists a wide range of poses variations from profile to frontal views. It also contains the variations of race, sex and appearance. Fig. 8 shows all the images of a particular subject

Table 5
Error rate in percentage for CMU PIE Face database.

Classifier	No.of training images							
	1	2	3	4	5	6	7	8
Proposed DC-OKELM	5.9	1.2	0.38	0	0	0	0	0
DC-ELM [40]	8.4	3.98	2.9461	0.0695	0.0947	0	0	0
Deep convolution [35]	16.68	12.74	3.03	0.116	0	0	0	0
ELM [28]	40	32.40	21.70	15.25	7.5	5.9	1.47	0.06
KELM [11]	26.8	10.2	29.2	20.76	1.4	1.5	0	0
BP [10]	50.68	38.43	13.52	12.56	10.70	10.8	4.97	0
DAG [26]	50.6	24	28	4.8	2.9	1.3	0.5	0.23
Stack Encoder [22]	5.5	4.7	4.1	2.4	0.9	0.3	0.11	0
Sparse Auto Encoder [25]	17.8	4.2	0.1	0	0	0	0	0

Table 6
Comparision of training time in sec. for CMU PIE Face database.

Classifier	Training time in sec. on CMU PIE database							
	1	2	3	4	5	6	7	8
Proposed DC-OKELM	141.5	154.4	166.7	134.8	122.5	154.7	156.7	158.0
Deep convolution [35]	901.9	1976.8	3007.1	3976.2	4858.7	5541.3	6109.8	6819.1
DAG [26]	422.2	680.5	642.1	1274	1297	1454	2008	2670
Stack Encoder [22]	631.6	610.4	602.6	614.0	628.1	649.9	685.8	694.7
Sparse Auto Encoder [25]	821.6	1654	2436	3103	3705	4272	4803	5292

Table 7
Error rate in percentage for UMIST Face database.

Classifier	UMIST							
	1	2	3	4	5	6	7	8
Proposed DC-OKELM	33.0	34.7	32.5	22	21	19	17.9	10.9
DC-ELM [40]	45.83	37.05	45	33.33	27.5	29.30	26.25	24.54
Deep convolution [35]	52.35	45.67	33.46	17.27	22.22	16.25	15	11.25
ELM [28]	50.28	49.26	45.62	42.67	44.46	46.73	45.62	42
KELM [11]	43.05	41.47	36.87	37	39.64	41.15	40.41	37.27
BP [10]	48.9	49.70	41.87	40.33	38.5714	38.076	40.167	33.63
DAG [26]	48.9	30.5	22.5	16.3	12.1	7.3	5.8	10
Stack Encoder [22]	53.3	52.6	55	54.6	56.0	50	50	46.8
Sparse Auto Encoder [25]	58.6	61.1	54.3	43	36.4	35.3	31.6	17.2

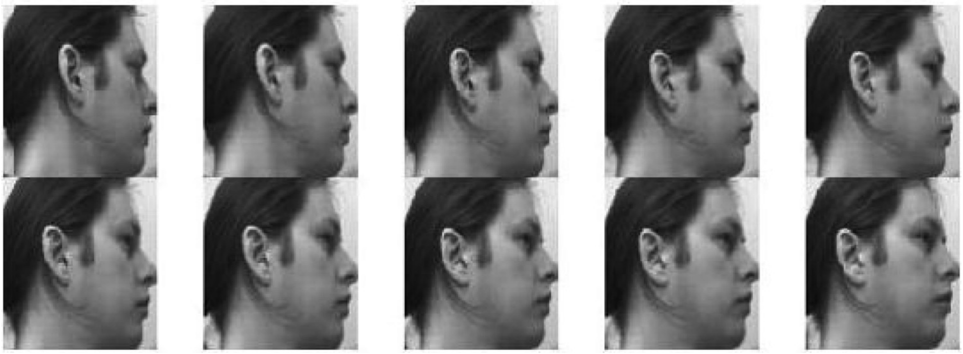

Fig. 8. Sample face images of a same person of UMIST database.

of this database. In the proposed work, first 19 face images per person have been selected to form a new sub-database of 380 images. Table 7 shows the comparative results of error rate the proposed DC-OKELM classifiers with other classifiers on UMIST database and Table 8 shows the comparison in training speed of the proposed method with deep learning classifier.

Parameters of the polynomial function of kernel ELM is optimized by using PSO optimization. PSO is simple, efficient, stochastic and adaptive technique for solving optimization problems. A comparison of PSO optimization with other five benchmark optimization algorithms: genetic algorithm (GA) [42], pattern search [43], grey wolf optimization (GWO) [44], bayesian optimization algorithm (BOA) [45], whale optimization algorithm (WOA) [46] on AT&T face database is shown in

Table 8
Comparison of training time in sec. for UMIST Face database.

Classifier	Training time in sec.							
	1	2	3	4	5	6	7	8
Proposed DC-OKELM	46.7	53.3	50.5	36.9	56.4	51.3	57.8	60.1
Deep convolution [35]	252.7	789.2	759.8	987	1147	1299	1473	1673
DAG [26]	65.4	96.7	135.5	164	197.1	233.3	242.4	253.7
Stack Encoder [22]	1888	1904	1930	1738	1624	1657	1673	1654
Sparse Auto Encoder [25]	248.8	491.5	720.0	925.4	1124	1327	1528	1727

Table 9
Comparison of Diffrent Optimization Techniques with Error Rate and Training Time.

Optimization technique	No. of Training Images	1	2	3	4	5	6	7	8
Ours	Error rate(%)	13.33	0.312	0.3571	0.4167	0.5	0.62	0.83	0
	Training time(in Sec.)	38.31	33.48	34.26	34.93	35.22	35.33	35.41	36.08
BOA	Error rate(%)	13.33	0.3125	0.3571	0.4167	0.5	0.6250	0.833	0
	Training time(in Sec.)	90.90	80.93	86.52	88.37	88.36	71.37	72.72	71.92
Pattern search	Error rate(%)	13.89	0.62	1.071	0.83	0.5	0.62	0.8	0.8
	Training time(in Sec.)	28.76	27.45	28.24	27.97	30.51	28.28	27.59	28.22
GA	Error rate(%)	13.3	0.312	0.357	0.416	0.5	0.82	0.83	0
	Training time(in Sec.)	86.18	83.31	88.53	86.20	88.77	89.22	89.78	91.89
GWO	Error rate(%)	14.44	2.18	1.07	0.83	1.5	2.5	2.5	0
	Training time(in Sec.)	24.60	24.17	25.06	25.43	24.81	24.43	24.43	24.45
WOA	Error rate(%)	14.44	2.18	1.07	0.83	1.5	2.5	2.5	0
	Training time(in Sec.)	24.22	24.20	24.23	24.29	25.09	24.577	24.25	28.25

Table 9. Here, it is shown that optimization using PSO achieves the less error rate in comparison with pattern search, GWO and WOA. Although error rate using GA and BOA is same as PSO optimization but the training time of PSO is better than these optimization algorithm. These results motivate us to use PSO as the optimization algorithm to be used in our proposed classifier as it gives best accuracy with the least time. Dong et al. [47] proposed supervised learning to improve PSO, but implementing supervised learning to improve the parameters of PSO is tedious process, it will take more training time than our method.

Discussion The proposed classifier DC-OKELM, which uses DCN for feature extraction and optimized KELM for classification, is compared with eight state of the art methods. It can be interpreted from Tables 1 to 4, that the proposed method for feature extraction and classifier achieved the best recognition rate among all the classifiers on four standard face databases. Also, the training time required for the proposed method for feature extraction and classification is very much less than required for DLN. These results prove the efficiency of the proposed method for face recognition compared with other conventional methods. During experiments, neither of the pre-processing nor variation normalization techniques has been used. Images are directly given to the DCN. These features undergo several convolution and max pooling operations which extracts high level of features from the face images. These features are fed to KELM classifier whose parameters are also optimized. The advantages of using optimized KELM for classification as compared to back propagation for tuning the parameters of DL are: 1) KELM consists of only single hidden layer and also non iterative training, therefore, training time is very much less than deep classifier. 2) There is no need of random initialization of any parameters selection of hidden neurons, activations function and kernel parameters are optimized using optimization algorithm. 3) It gives better performance results compare to deep convolutional neural networks.

5. Conclusions

A novel and fast approach of feature extraction and face recognition based upon deep convolutional networks & KELM is presented in this paper. Residual network (ResNet) is used to extract robust features from the input image which are invariant to illumination, pose, expression using the set of convolution and pooling layers. It also provides partial invariance to translation, rotation and shifts. These extracted features are learned and classify using polynomial function Kernel ELM whose parameters are also optimized using PSO optimization algorithm. The present method is capable of rapid classification and consistently exhibits better classification performance than the other classification approach. To evaluate the performance of the present work, experiments are performed on AT&T, Yale, CMU PIE and UMIST face databases. With 5 images per person, it gives the error rate of 0.5, 8.89, 0, & 21 respectively on AT&T, Yale, CMU PIE and UMIST databases. These results are achieved without any normalization on these database. It is the lowest error rate compared to state of the art techniques. Training time of the proposed method is also very less compared to deep learning networks because it need not any iterative tuning of the network. Therefore, the proposed algorithm recognizes faces with highest accuracy in less time so that can be used for real time applications. Further work can be done on optimization of weights and biases of the hidden nodes of kernel ELM classifier.

Declaration of Competing Interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

CRediT authorship contribution statement

Tripti Goel: Conceptualization, Formal analysis, Writing - original draft, Writing - review & editing. **R Murugan:** Funding acquisition, Writing - original draft, Writing - review & editing.

Acknowledgements

All persons who,have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given as their written permission tb be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

References

[1] Pentl A, Choudhury T. Face recognition for smart environments. Computer 2000;33(2):50–5.
[2] Jain AK, Li SZ. Handbook of face recognition. Springer; 2011.
[3] Chellappa R, Wilson CL, Sirohey S, et al. Human and machine recognition of faces: A survey. Proceedings of the IEEE 1995;83(5):705–40.
[4] Gross R, Brajovic V. An image preprocessing algorithm for illumination invariant face recognition. In: International Conference on Audio-and Video-Based Biometric Person Authentication. Springer; 2003. p. 10–18.
[5] Georghiades AS, Belhumeur PN, Kriegman DJ. From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis & Machine Intelligence 2001(6):643–60.
[6] Ling H, Soatto S, Ramanathan N, Jacobs DW. A study of face recognition as people age. In: 2007 IEEE 11th International Conference on Computer Vision. IEEE; 2007. p. 1–8.
[7] Park U, Tong Y, Jain AK. Age-invariant face recognition. IEEE transactions on pattern analysis and machine intelligence 2010;32(5):947–54.
[8] Schmidhuber J. Deep learning in neural networks: An overview. Neural networks 2015;61:85–117.
[9] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics; 2010. p. 249–56.
[10] Latha P, Ganesan L, Annadurai S. Face recognition using neural networks. Signal Processing: An International Journal (SPIJ) 2009;3(5):153–60.
[11] Huang G-B. An insight into extreme learning machines: random neurons, random features and kernels. Cognitive Computation 2014;6(3):376–90.
[12] Turk MA, Pentland AP. Face recognition using eigenfaces. In: Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE; 1991. p. 586–91.
[13] Belhumeur PN, Hespanha JP, Kriegman DJ. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis & Machine Intelligence 1997(7):711–20.
[14] Bartlett MS, Movellan JR, Sejnowski TJ. Face recognition by independent component analysis. IEEE Transactions on neural networks 2002;13(6):1450–64.
[15] Er MJ, Wu S, Lu J, Toh HL. Face recognition with radial basis function (rbf) neural networks. IEEE transactions on neural networks 2002;13(3):697–710.
[16] Parkhi OM, Vedaldi A, Zisserman A, et al. Deep face recognition. In: bmvc, 1; 2015. p. 6.
[17] Kim Y. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 2014.
[18] LeCun Y, Bengio Y, Hinton G. Deep learning. nature 2015;521(7553):436.
[19] Bell S, Lawrence Zitnick C, Bala K, Girshick R. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2874–83.
[20] Simard L, Willmer CN, Vogt NP, Sarajedini VL, Phillips AC, Weiner BJ, et al. The deep groth strip survey. ii. hubble space telescope structural parameters of galaxies in the groth strip. The Astrophysical Journal Supplement Series 2002;142(1):1.
[21] Khalajzadeh H, Mansouri M, Teshnehlab M. Face recognition using convolutional neural network and simple logistic classifier. In: Soft Computing in Industrial Applications. Springer; 2014. p. 197–207.
[22] Zhou P, Han J, Cheng G, Zhang B. Learning compact and discriminative stacked autoencoder for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing 2019.
[23] Gu J, Hu H, Li H. Local robust sparse representation for face recognition with single sample per person. IEEE/CAA Journal of Automatica Sinica 2017;5(2):547–54.
[24] Yu D, Li J. Recent progresses in deep learning based acoustic models. IEEE/CAA Journal of Automatica Sinica 2017;4(3):396–409.
[25] Luo W, Li J, Yang J, Xu W, Zhang J. Convolutional sparse autoencoders for image classification. IEEE Transactions on Neural Networks and Learning Systems 2018;29(7):3289–94. doi:10.1109/TNNLS.2017.2712793.
[26] Shuai B, Zuo ZL, Wang B, Wang G. Scene segmentation with dag-recurrent neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018;40:1480–93.
[27] Bertsekas DP. Feature-based aggregation and deep reinforcement learning: A survey and some new implementations. IEEE/CAA Journal of Automatica Sinica 2018;6(1):1–31.
[28] Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: theory and applications. Neurocomputing 2006;70(1-3):489–501.
[29] Zhu Q-Y, Qin AK, Suganthan PN, Huang G-B. Evolutionary extreme learning machine. Pattern recognition 2005;38(10):1759–63.
[30] Liang N-Y, Saratchandran P, Huang G-B, Sundararajan N. Classification of mental tasks from eeg signals using extreme learning machine. International journal of neural systems 2006;16(01):29–38.
[31] Huang G-B, Chen L. Enhanced random search based incremental extreme learning machine. Neurocomputing 2008;71(16-18):3460–8.

[32] Deng W-Y, Zheng Q-H, Lian S, Chen L, Wang X. Ordinal extreme learning machine. Neurocomputing 2010;74(1-3):447–56.
[33] Matias T, Araújo R, Antunes CH, Gabriel D. Genetically optimized extreme learning machine. In: 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA). IEEE; 2013. p. 1–8.
[34] Castaño A, Fernández-Navarro F, Hervás-Martínez C. Pca-elm: a robust and pruned extreme learning machine approach based on principal component analysis. Neural processing letters 2013;37(3):377–92.
[35] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence; 2017.
[36] He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. In: European conference on computer vision. Springer; 2016. p. 630–45.
[37] Toh K-A, Kim J, Lee S. Biometric scores fusion based on total error rate minimization. Pattern Recognition 2008;41(3):1066–82.
[38] Kennedy J. Particle swarm optimization. Encyclopedia of machine learning 2010:760–6.
[39] Shi Y, et al. Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), 1. IEEE; 2001. p. 81–6.
[40] Pang S, Yang X. Deep convolutional extreme learning machine and its application in handwritten digit classification. Computational intelligence and neuroscience 2016;2016.
[41] Sim T, Baker S, Bsat M. The cmu pose, illumination, and expression (pie) database. In: Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition. IEEE; 2002. p. 53–8.
[42] Malhotra R, Singh N, Singh Y. Genetic algorithms: Concepts, design for optimization of process controllers. Computer and Information Science 2011;4(2):39.
[43] Ravindra M, Rao RS. A pattern search algorithm approach for optimal allocation of pmus considering sensitive bus constraints to obtain complete observability. International Journal of Mathematical and Computational Methods 2017;2.
[44] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in engineering software 2014;69:46–61.
[45] Torun HM, Swaminathan M, Kavungal Davis A, Bellaredj MLF. A global bayesian optimization algorithm and its application to integrated system design. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2018;26(4):792–802. doi:10.1109/TVLSI.2017.2784783.
[46] Zhang Q, Liu L. Whale optimization algorithm based on lamarckian learning for global optimization problems. IEEE Access 2019;7:36642–66. doi:10.1109/ACCESS.2019.2905009.
[47] Dong W, Zhou M. A supervised learning and control method to improve particle swarm optimization algorithms. IEEE Transactions on Systems, Man, and Cybernetics: Systems 2016;47(7):1135–48.