Int. J. Nonlinear Anal. Appl. 11 (2020) No. 1, 301-319
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2020.4296

Comparing performance of metaheuristic algorithms for finding the optimum structure of CNN for face recognition

Arash Rikhtegar[a], Mohammad Pooyan[b]*, Mohammad Taghi Manzuri[c]

[a] *Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.*
[b] *Engineering faculty, Shahed University, Tehran, Iran.*
[c] *Computer Engineering Department, Sharif University of Technology, Tehran, Iran.*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Local and global based methods are two main trends for face recognition. Local approaches extract salient features by processing different parts of the image whereas global approaches find a general template for face of each person. Unfortunately, most global approaches work under controlled environments and they are sensitive to changes in the illumination. On the other hand, local approaches are more robust but finding their optimal parameters is a challenging task. This work proposes a new local-based approach that automatically tunes its parameters. The proposed method incorporates different techniques. In the first step, convolutional neural network (CNN) is employed as a trainable feature extraction procedure. In the second step, different metaheuristic methods are merged with CNN so that its best structure is found automatically. Finally, in the last step the decision is made by employing proper multi-class support vector machine (SVM). In this fashion a fully automated system is developed that is self-tuned and do not need manual adjustments. Simulation results demonstrate efficacy of the proposed method.

Keywords: Face Recognition; Convolutional Neural Network; Support Vector Machine; Multi-Class Classification; Metaheuristic Algorithm;
2010 MSC: 68T10,68T50

*Mohammad Pooyan
Email address: a.rikhtegar@srbiau.ac.ir, pooyan@shahed.ac.ir, manzuri@sharif.edu (Arash Rikhtegar[a], Mohammad Pooyan[b]*, Mohammad Taghi Manzuri[c])

Received: December 2019 *Revised:* February 2020

1. Introduction

Recently, human computer interaction (HCI) has found a lot of attention. This interesting branch of science focuses on finding and developing more elegant methods for interaction between human and computers. For example, some commercial products have introduced more natural and friendlier interfaces for interaction between human and computers. Such solutions rely on listening to the user on the microphone and watching him/her on a camera. Apparently, face recognition is among the primary techniques that makes such solutions possible. Additionally, face recognition systems can be integrated with identification, authentication, access control, and surveillance systems [1]. Although, effective identification systems based on fingerprints or scan of iris exist, but they need participant's cooperation and participant should be aware of the interaction. On the other hand, some applications require a face recognition method that can work in the absence of the participant's cooperation and his knowledge [2]. Reliable face recognition systems could address such situations.

1.1. Related works

Most face recognition methods can be classified either as global-based approach or local-based approach. Global-based schemes learn a general template for face of each person in the training phase. Usually, this task is performed by employing a linear or a non-linear projection mapping. Then, in the recognition phase the target image is compared with template of all individuals and the best match is selected. On the other hand, in the local-based approaches different parts of image are processed separately and a set of features are extracted from each image. Then, these features are used to find discriminative patterns between different persons.

Principal component analysis (PCA) [3], linear discriminate analysis (LDA) [4], and independent component analysis (ICA) [5] are among the most well-known linear projection schemes and all of them use the same concept. These methods employ a linear projection technique and map the high-dimensional space of images into a low-dimensional space of features. The main advantage of these methods is their low complexities. However, effects of variation in illumination, facial expression, and pose on the image are not linear. Consequently, these linear methods cannot represent faces with a lot of variations. To address this, non-linear projection methods have been proposed. Non-linear methods are based on kernel technique. That is, images are first mapped into a higher-dimensional space in which images may constitute a linear space. Then, traditional linear methods are used for converting back into a low-dimensional space of features. These methods include kernel PCA [6], kernel ICA [7], and generalized LDA [8]. Previous works have shown that performances of kernel methods highly rely on choice of kernel and their parameters also influence the results considerably [9]. Therefore, non-linear global methods may not produce very good results.

In the local-based approaches each part of image is processed separately; therefore, they are more robust to variations such as changes in the facial expression, partial occlusion, misalignment of image, and even non-linear effects such as variations in illumination. Different local-based approaches have been proposed in the literature. Local binary pattern (LBP) describes change of adjacent pixels in terms of their central pixel [10]. In this fashion local patterns are described by normalizing the neighboring pixels. This method is robust to small illumination variations and monotonic intensity transformations. Later, enhanced LBP method was introduced where image is divided into a set of sub-images and then LBP is performed on each of sub-images. In [11] different aspects of LBP and enhanced LBP were studied, and it was shown that enhanced LBP outperforms other LBP-based methods. Wen et al. used an improved discriminative common vector for feature extraction [12]. Then, these features were used to train a support vector machine (SVM). Another work proposed the idea of multi linear neighbor preserving projection [13]. This approach preserves the geometry

and local structure of neighbor pixels. Abusham et al. used local graph structure for learning the spatial information of neighboring pixels [14].

Recently, local approaches based on deep learning structures have been used for face recognition. Convolutional neural network (CNN) with bottleneck structure was used for learning proper mapping of images into a low dimension space [15]. Idea of adaptive CNN was proposed in [16] while another method used bottleneck structure without weight sharing technique [17]. This method resulted in a very complex network with eight different layers and over 120 million free parameters. Authors of [18] investigated properties of CNN and compared its performance with commercial solutions for face recognition. They argued that bottleneck structure is not very efficient and instead they proposed a CNN structure that directly optimizes the Euclidian space and showed that this technique decreases within class distances [19].

1.2. Our contributions

Global approach fails to address effects of illumination or partial occlusion efficiently. On the other hand, local approach has lots of parameters that should be determined. Unfortunately, determining these parameters is not a trivial task [15]. Recently, approaches based on deep learning has gained a lot of attention. But, there are some shortcomings for the existing methods. For example, some of them involve lots of parameters [17]. Determining such a huge number of parameters make such systems computationally inefficient. Other works based on deep learning have used the bottleneck structure [15, 18]. In the bottleneck structure a non-linear map from images into a low dimension space is learned, and then the original image is reconstructed from this compressed representation. Recent studies have suggested this structure is not efficient and it may diminish the results [19]. Another problem with CNN is determining its optimum structure. Determining structure of CNN involves lots of parameters and hence unlike ordinary neural networks, in CNN an exhaustive search over all possible structures is not possible. This work proposes solutions for these problems.

Previous works have metaheuristic algorithms are very powerful tools for solving complex optimization problems [20, 21]. This work exploits potency of metaheuristic algorithms for finding the optimum structure of CNN for face recognition tasks. In this regard three popular methods of genetic algorithm (GA), simulated annealing (SA), and imperialist competitive algorithm (IC) are implemented and their performances are compared.

In the traditional CNNs the last layer is a perceptron layer. On the other hand, previous works have shown that SVM outperforms neural networks [22]. Therefore, in this work CNN structure is modified and the last layer is replaced with SVM. In this fashion a hybrid system is developed that uses potency of both neural networks and SVM. That is, the neural network part of CNN is retained as a powerful trainable feature extractor while the final decision is made by the SVM.

To construct a multi class SVM, different strategies are possible. Also, in the face recognition tasks the number of classes may be high, so the proper method should be selected. To address this issue, performances of different SVM ensembles for face recognition task are compared.

The rest of this paper is organized as follows. Structure of the proposed method is presented in the section 2. Section 3 is devoted to the evaluation of the proposed method. Finally, the paper is concluded in the section 4.

2. Face Recognition with CNN

The proposed method has three main components. In the first step, CNN is used for extracting salient features. In the second step, potency of different metaheuristic algorithms for finding the optimum structure of CNN are compared. Finally, in the last step different structures of multi class SVM are compared for making the final decision.

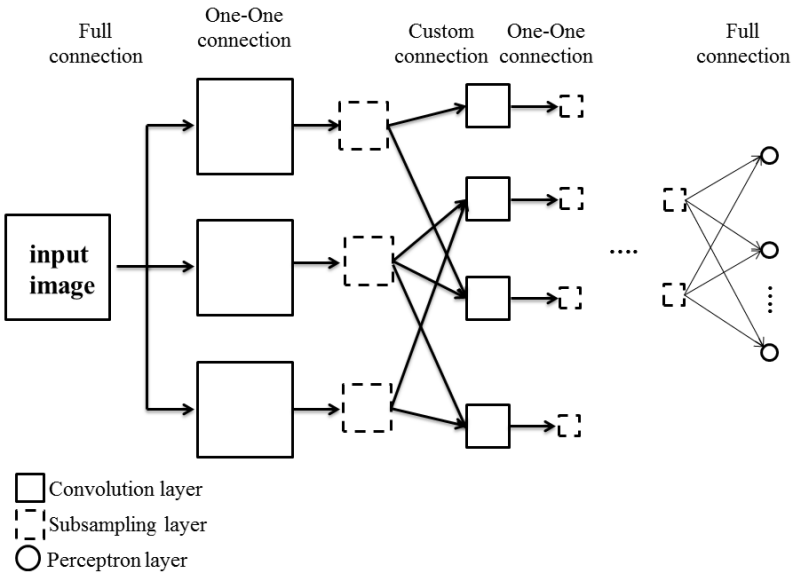

Figure 1: General structure of CNN

2.1. Feature extraction and CNN

The first step of the proposed method is extracting salient features from images. A set of suitable features have small within-class distances. This characteristic is very crucial, and it makes the system robust to changes in the facial expression and illumination. In this fashion correct recognition of a person is guaranteed. On the other hand, it is desired that features have large between class distances. This characteristic governs that a person is not recognized as somebody else. To extract features, usually a series of signal processing techniques are applied. The main objective of this phase is determining suitable processes such that extracted features have small within-class distance and large between-class distance. Determining such processes is challenging in many pattern recognition problems. CNNs are a special type of neural networks that can find the proper processes themselves. Therefore, they have the potency to address this problem.

Idea of CNN was motivated by the evidence of locally sensitive, orientation-selective neurons in the visual cortex of cats [23]. CNNs are hierarchical networks that consist of multiple layers of neurons. These networks are an interleaved arrangement of convolutional and subsampling layers. Each convolutional layer has a set of neurons called receptive fields where their inputs are convolved with these trainable kernels. Thus, these receptive fields work like fixed feature extractors and locally process results of previous layer of network. In most cases each layer has more than one receptive field, so that different features are extracted from each layer. In contrast, subsampling layers reduce resolution of their inputs by perform an averaging operation.

Three main ideas of CNN are local receptive fields, spatial subsampling, and weight sharing. These concepts help the system to be robust to shifts and deformations in the image [24]. Weight sharing is the property that allows each neuron to use the same weight for all pixels. Therefore, it drastically reduces the number of free parameters and improves generalization property of the network. Like many other neural network, CNNs are usually trained with back propagation algorithm [25]. It is noteworthy that both convolutional layers and subsampling layers are followed by an activation function which performs a non-liner operation. Fig. 1, shows the general structure of a CNN.

2.2. Optimizing structure of CNN

Like other neural networks, proper parameters of CNN should be determined. These parameters include number of layers, number of receptive fields in each layer, size of receptive fields in each layer, connection scheme between subsampling and convolutional layers, and activation functions of each layer. Apparently, the search space is very big and exhaustive search is not practical. On the other hand, if the proper structure is not found, performance of the system would diminish considerably. Based on previous studies, metaheuristic methods are efficient ways for solving complex optimization problems. This work investigates performance of three different metaheuristic algorithms for determining the optimum structure of CNN.

1. Genetic algorithm (GA)

   GA is a population based metaheuristic algorithm which mimics biological evolution and it is based on the concept of "survival of the fittest". The algorithm starts with a set of random candidate solutions and as it proceeds through different generations, the promising traits are selected and passed down to the next generation. This is achieved by only allowing the best individuals from each generation to participate in the breeding of the next generation.

2. Simulated annealing (SA)

   Annealing is a heat treatment technique in the metallurgy where a metal is heated to very high temperatures and then it is cooled down very slowly. This process allows atoms of the metal to form a crystal structure with minimum level of energy. SA mimics this process for solving complex optimization problems. It starts with a very high temperature and a random solution. Then, the solution undergoes a small modification and the new value of objective function is calculated. If the new solution has a lower cost, then the new solution is accepted, otherwise, this "worse" solution is accepted with the probability of:

   $$p{=}e^{-\Delta E/t} \tag{2.1}$$

   where ΔE and t are differences between the two cost functions and current temperature, respectively. According to Eq. (2.1), at high temperatures more "worse" solutions are accepted, but at low temperatures this probability decreases. This mechanism allows algorithm to escape from local optima and to converge to global optima of the problem.

3. Imperialist competitive algorithm (IC)

   IC algorithm is another population-based metaheuristic algorithm that was proposed recently [29] and it has been applied for solving numerous problems [30]. It is based on the socio-political behaviors and interactions between countries. The algorithm starts with a set of random solutions to the problem which are called countries. Then, some of the best solutions act as imperialists and compete to gain control of the remaining countries. In the assimilation phase, imperialists influence their colonies and change some of their properties. During different runs of the algorithm, some empires grew stronger and some grew weaker. Consequently, the weakest empire loses some of its colonies and eventually it may collapse. Also, it is possible that some colonies experience a revolution which is a sudden change in their properties. Revolution operation prevents the premature convergence of the algorithm and it helps the algorithm to perform a better search.

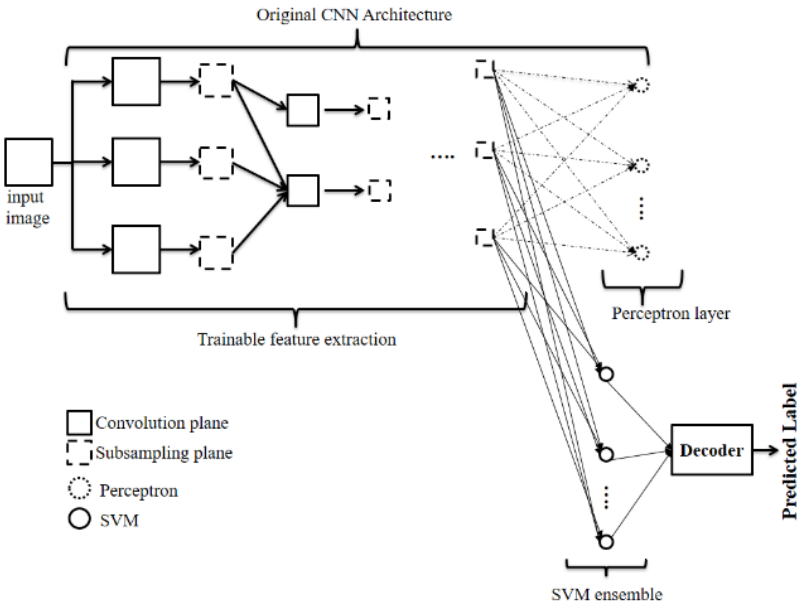

Figure 2: Hybrid SVM-CNN structure

2.3. Classification

After determining optimum structure of CNN, features are extracted and then proper decision boundaries are defined. In the standard architecture of CNN, a single layer of neuron (also known as perceptron) implements this part. But, it is quite possible to use other machine learning techniques instead of this layer. For example, previous works have compared SVM with neural networks and have pointed out to some advantages of SVM. For example, perceptron is trained with back propagation algorithm which converges to local minimum. On the other hand, due to convexity of non-linear SVM, global minimum is found [31]. Also, besides having a much stronger theoretical foundation [32], SVM does not have the problem of determining its optimum structure. In other words, unlike neural network where its structure should be determined, SVM automatically selects the support vectors and finds the optimum model [33].

Based on these reasons and previous results [34], we have replaced the last layer of CNN with SVM to achieve better performance. For this purpose, original CNN network with perceptron in its last layer was trained until its parameters were converged. Then, last layer of trained CNN was removed and the remaining structure was retained as feature extraction procedure. Finally, the recognition was achieved by a multi-class SVM. Fig. 2, depicts this.

2.3.1. Binary SVM classifier

The process of distinguishing between different classes needs a classifier to define suitable decision boundaries. SVM is based on Vapnik's statistical learning theory where a hyper plane with maximum-margin is created to distinguish between different classes [35]. SVM achieves great performance by maximizing this margin. In this fashion the probabilistic test error bound is minimized.

Considering training vectors $x_k \in R^n, k = 1, 2, \ldots, m$ for two classes, and corresponding labels vector$y_k \in \{1, -1\}$, SVM solves a quadratic optimization problem as:

$$\arg\min_{\omega,b,\xi} \left\{ \frac{1}{2}\omega^T\omega + C\sum_{k=1}^{m} \xi_k \right\} \tag{2.2}$$

Subject to

$$\begin{aligned} &y_k(\omega^T\phi(x_k)+b) \geq 1-\xi_k, \\ &\xi_k \geq 0, k=1,...,m, \end{aligned} \tag{2.3}$$

Furthermore, if different classes are not linearly separable it is possible to map the original feature-space into a new space with much higher dimension where features may be linearly separable. Linear classification in this new space is equivalent to non-linear classification in the original feature-space. This is done using kernel trick, where the dot product of SVM cost is replaced with:

$$K\left(x_i, x_j\right) \equiv \phi\left(x_i\right)^T . \phi\left(x_j\right) \tag{2.4}$$

where, ϕ denote a kernel function. If radial basis function (RBF) is used as the kernel, Eq. (2.4) reduces to:

$$K\left(x_i, x_j\right) = \exp\left(-\gamma||x_i - x_j||^2\right), \gamma > 0 \tag{2.5}$$

After training the SVM, test samples are classified according to:

$$f(x) = \omega^T\phi(x) + b \tag{2.6}$$

While $|f(x)|$ provides a measure of confidence, usually a hard threshold is applied on the $f(x)$ to produce the predicted label:

$$\tilde{y} = sgn(f(x)) \tag{2.7}$$

In this work, LIBSVM [36] with RBF kernel is used.

2.3.2. Multiple class case

Face recognition is an instance of multi-class problem; therefore, the classifier should support multi-class decision. Given the binary classifier $f: x \longrightarrow \pm 1$, we can extend it to a multi-class paradigm by employing an ensemble of such classifiers. Target ensemble consists an encoding function, n binary classifiers ($[f_1, f_2, \ldots, f_n]$), and a decoding function. Different types of ensembles can be constructed based on the encoding function.

- Distributed Output Coding (DOC)

In this method encoding function creates a binary codebook with the size of $k \times n$, where k is the number of classes and$n = \lceil log_2(k) \rceil$. Every row of the codebook is assigned to a class and every column of the codebook is used for one of binary classifiers. Apparently, columns of the codebook consist of a series of 1 and -1. Thus, in the training phase, a binary classifier is constructed for every column of the codebook. In the testing phase all the binary classifiers are evaluated and the corresponding code word is constructed:

$$\tilde{D} = [sgn(f_1(x)), sgn(f_2(x)), \ldots, sgn(f_n(x))] \tag{2.8}$$

Then, the predicted label is calculated as:

$$\tilde{y} = \underset{i}{arg\,min} \left\{ H\left(\tilde{D}, R_i\right) \right\} \tag{2.9}$$

Where, H is the hamming distance and R_i is row i of the codebook.

Figure 3: Sample images from ORL and Yale databases

- One Against All (One-All)

In this method encoding function creates a diagonal binary codebook with the size of $k \times k$. This scheme trains a single classifier per each class, with the samples of that class as positive instances and all other samples as negative instances. Therefore, this method consists of k binary classifier. In the testing phase all classifiers are applied on the sample and the label is predicted as the index of classifier with the highest value of confidence interval [37].

$$\tilde{y} = \underset{i}{arg\,max} \{|f_i(x)|\} \tag{2.10}$$

- One Against One (One-One)

This ensemble needs *k×(k-1)/2* number of binary classifiers. In this fashion a classifier is trained for distinguishing between every possible pairs of classes. In the testing phase, all classifiers are applied on the sample. Then, the outcomes are combined based on voting or decision directed acyclic graphs [38, 39].

3. Experimental Results

To evaluate performance of the proposed methods three publicly available databases of ORL, Yale, and AR were employed. ORL database has a total number of 400 images taken from 40 distinct individuals subjected to different lightings, facial expressions (open/closed eyes, smiling/not smiling), and facial details (glasses/no glasses). The faces are in an upright, frontal position against a dark homogenous background. Yale database has a total number of 165 different images, taken from 15 different persons. Pictures of each person are taken under eleven variations of facial expressions and configurations including, center-light, with glasses, happy, left-light, with no glasses, normal, right-light, sad, sleepy, surprised, and wink. Fig. 3, shows samples of both ORL and Yale databases.

AR database is a color image database and it includes images from 126 different people. There are 26 different frontal views of each person with different facial expressions, lighting conditions, and occlusions. Each person was photographed on two different sessions (14 days apart) and in each session 13 images were taken. In our experiments with AR database, it was split into three different subsets. The first part is denoted by AR-Expression and it covers different facial expressions and illumination conditions. This part contains 14 images per individual. The second part is denoted by AR-Sunglasses and it contains 6 images per individual with sunglasses occlusions. The last part is denoted by AR-Scarf and it contains 6 images per individual with scarf occlusions. Sample images from each of these three parts are presented in Fig. 4.
Table 1 shows details of the employed databases.

It is noteworthy that for all experiments images were cropped from their original sizes into images with 64×64 pixels. Also, all samples from AR database were converted into gray scale with rgb2gray() function within Matlab environment. Finally, all simulations were carried out in Matlab environment and SVM was imported as a Mex file from libsvm library.

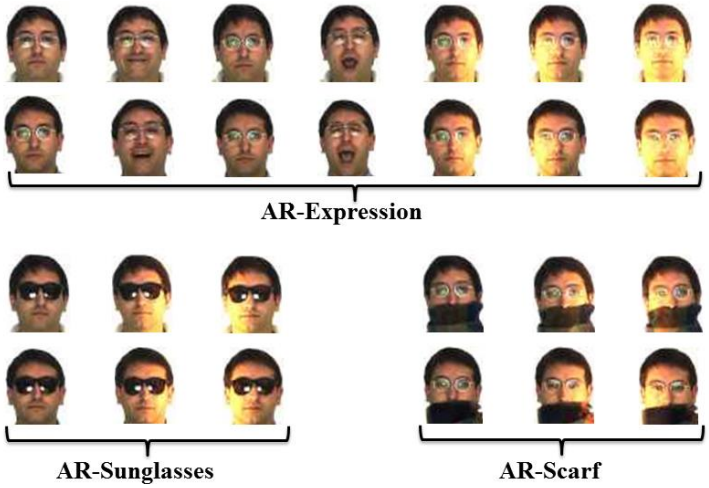

Figure 4: Sample images AR database

Table 1: Details of the employed databases

Database		No. of individuals	No. of image per individual
ORL		40	10
Yale		15	11
AR	Expression	126	15
	Sunglasses	126	6
	Scarf	126	6

3.1. Performance of different metaheuristic algorithms

CNN has a very complex structure with lots of parameters to be tuned. These parameters include, number of layers (N), activation function of each layer (ξ_i), number of receptive fields in convolutional layers (η_i), size of receptive fields in convolutional layers (S_i), and custom connection between subsampling and convolutional layers (χ_i). In our simulation we used four different activation functions, therefore we had 4^N different possibilities. These activation functions are defined in the Eq. (3.1) to (3.4).

$$ltanh\,(x) \;= 1.7159\; tanh(\frac{2x}{3}) \tag{3.1}$$

$$tansig\,(x) = \frac{2}{1+e^{-2x}} - 1 \tag{3.2}$$

$$linear\,(x) = x \tag{3.3}$$

$$logsig\,(x) = \frac{1}{1+e^{-x}} \tag{3.4}$$

We allowed $\eta_i \in [2, 15]$. Also, the number of receptive fields in the subsampling layer is always equal to its preceding convolutional layer and last layer is a single perceptron; therefore, we had $14^{\lfloor N/2 \rfloor}$ different possibilities for number of receptive fields.
We performed averaging by 2 in the subsampling layer. Consequently, only size of receptive fields in the convolutional layers should be determined. We allowed $S_i \in [3, 10] \times [3, 10]$. Apparently, both length and width of receptive field could vary; therefore, we had $64^{\lfloor N-1/2 \rfloor}$ different possibilities for size of receptive fields. It is noteworthy that if network has even number of layers, size of last convolutional layer is not a free parameter.
There is always a one-one connection between subsampling layer and its preceding convolutional layer. Also, the last layer always has a full connection. Therefore, connection between subsampling layer and its next convolutional layer is the only free parameter. Let η_i and η_{i+1}denote number of receptive fields in a subsampling layer and its next convolutional layer, respectively. Then, the connection matrix $(C_{i,i+1})$ is a binary matrix with η_i rows and η_{i+1} column. In this manner if $c_{k,l} = 1$, then there is a connection between receptive field k from subsampling layer i to receptive field l in the convolutional layer *i+1*. If connection matrix is constructed randomly we would have $2^{\eta_i \times \eta_{i+1}}$ different possibilities. η_i and η_{i+1} are generated independently from uniform distributions; therefore, its expected is $E[\eta_i \times \eta_{i+1}] = 72.25$. Furthermore, there are only $\alpha = \lfloor N - 1/2 \rfloor$ succession of subsampling and convolutional layer; therefore, on average there are $2^{72.25 \times \alpha}$ possibility for generating all connection matrices.

To put these numbers into perspective, if N=6, then there are 4096 different possibilities for activation functions, 2744 different possibilities for number of receptive fields, 4096 different possibilities for size of receptive fields, and more than 3×10^{43} different possibilities for generating connection matrices. Apparently, searching this space is not practical. Therefore, metaheuristic algorithms are good candidates.

In this section performances of different metaheuristic algorithms for searching this space are compared. Objective function of all methods was defined as mean square error (MSE) of target CNN structure after 2000 epochs in the training phase of CNN on the ORL database. Also, to improve the search only connections matrices were used that had at least one value of 1 in each row and each column. Other parameters of metaheuristic algorithms are described in the following paragraphs.

Our implementation of GA had 100 individuals with two-point cross over [26] and tournaments selection [27]. Additionally, elitism with rate of 1% was incorporated into our algorithm and for mutation 1% of individuals were randomly selected and one of their parameters was replaced with a random value.

Cooling schedule is among the most important parameters of SA. We used linear cooling schedule with rate of 0.99. Initial temperature is another important parameter of the SA algorithm. To find the proper initial temperature the algorithm proposed in [28] was used for achieving the acceptance ratio of 0.8.

We used 100 countries with 8 initial imperialists for IC algorithm. In the assimilation process some properties of each colony were selected randomly and they were replaced with their imperialist counterparts. Finally, revolution with probability of 1% was implemented as randomly selecting a property of the country and changing it with a new random value.

Metaheuristic algorithms are stochastic in their nature. That is, different runs of algorithms results in different outcomes. Therefore, to achieve more reliable results, each experiment was repeated for 10 times. Values of mean and standard deviation of MSE of each method are reported in table 2.

Referring to results of table 2, IC algorithm archives the lowest value of mean and std of error in the training phase. Achieving the lowest value of mean of error shows that the structures found by this

Table 2: Mean and std of MSE in training phase of different CNN structures

Algorithm	Mean	std
IC	0.273	0.037
SA	0.336	0.048
GA	0.471	0.095

algorithm are better candidates. On the other hand, achieving a low value of std of error shows that different runs of the algorithm results in structures with comparable performances. Based on these arguments, if an algorithm achieves a low value of mean and std of error, we can be more confident that the structure is optimum. Consequently, we conclude that IC algorithm is more powerful for determining the optimum structure of CNN for face recognition. To further compare performance of different structures, MSE of training phase of the best structure found by each algorithm is plotted in Fig. 5.

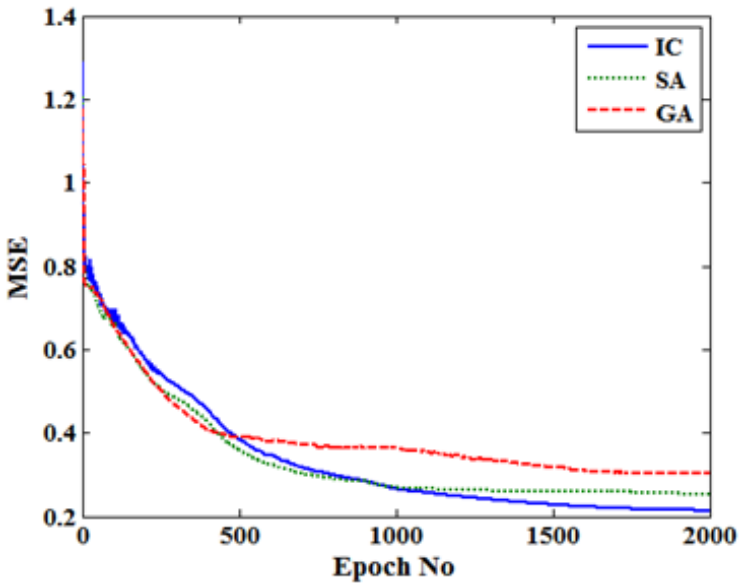

Figure 5: MSE of best CNN structures found by different metaheuristic algorithms

Based on Fig. 5, the structure found by IC has the lowest value of MSE in the training phase and hence it has better performances. Table 3 presents parameters of this 6-layer CNN structure. Furthermore, the custom connections between layer 2-3 ($C_{2,3}$) and layer 4-5 ($C_{4,5}$) are presented in tables 4 and 5, respectively.

3.2. Performance of different ensembles of SVMs

To investigate recognition rate of the proposed method a set of experiments was carried out. Each experiment consists of three phases. First, the best CNN structure found from IC algorithm was trained using training samples. Then, the updated network was applied on each train sample and the inputs of the last layer were saved as feature vectors. Finally, these features were used to train the SVM ensemble.

Table 3: Parametrs of the best CNN structure, parametrs that are not free are denoted by a (-) mark

Layer No.	ξ_i	η_i	S_i
1 (convolutional)	linear	4	[9×8]
2 (subsampling)	logsig	-	-
3 (convolutional)	ltanh	7	[5×4]
4 (subsampling)	ltanh	-	-
5 (convolutional)	tansig	14	-
6 (perceptron)	ltanh	-	-

Table 4: Costum connection between receptive fields of second and third layers

		Third Layer						
Second	Layer	1	0	1	1	0	0	0
		1	0	0	1	1	0	1
		0	1	1	1	0	1	0
		1	1	0	0	0	1	1

For each test on ORL and Yale databases 5 images of each person were selected randomly for training and the remaining samples were used for testing. This procedure was repeated for ten times and then average values of recognition rates were calculated. For each test on AR-Expression one of the sessions was used for training (7 images) and images from the other session were used for testing.

CNNs have very complex structures and therefore their training is usually very time consuming. In all of simulations that follows we used 4000 epochs for training of CNN. Table 6, presents training time of the best structure from IC algorithm on each database. The time was measured in seconds and on a system with i5-6200 processor.

Going back to section 2.3.2, different ensemble methods are possible. We define complexity of each method as the number of binary classifiers it requires. As we mentioned earlier, DOC needs $\lceil log_2(k) \rceil$ binary classifier, One-All needs k binary classifiers and One-One needs $k \times (k\text{-}1)/2$ number of binary classifiers, where k is the number of classes. Table 7, compares complexity of different SVM ensembles.

Now, performance of different SVM ensembles are compared. Referring to table 7, complexity of one-one method for AR database is very high; therefore, AR database was only tested in DOC and one-all settings. Average values of recognition rates of all methods are reported in table 8.

Comparing results of table 8, shows that DOC achieves the lowest recognition rate, since DOC

Table 5: Costum connection between receptive fields of fourth and fifth layers

	Fifth layer													
Forth Layer	0	0	1	1	1	0	0	0	1	0	1	1	0	1
	1	1	1	0	1	0	1	0	1	1	0	1	0	0
	0	1	1	0	1	1	0	0	0	1	1	0	1	1
	0	0	0	0	1	0	1	1	0	0	0	1	0	1
	1	0	1	1	0	1	1	0	1	0	1	0	1	0
	0	1	0	0	1	0	1	0	0	1	1	0	0	0
	0	0	1	0	0	0	0	1	0	1	0	1	0	1

Table 6: training time of best CNN structure on different databases

Database	Yale	ORL	AR-Expression
Time (s.)	4078	9997	45663

Table 7: Complexity of different ensembles

		No. of Binary Classifiers		
		ORL	Yale	AR-Expression
Ensemble method	DOC	6	4	7
	One-All	40	15	126
	One-One	780	105	7875

method is not resilient to classification errors and even a single error will result in a wrong final decision. In contrast, one-one has the highest level of resiliency and therefore it has achieved the best value of recognition rates. But, the high level of complexity of one-one method makes it impractical for large databases. Consequently, for the rest of simulations one-all setting was adopted.

Table 8: The average recognition rates of different SVM ensembles

		Databases		
		ORL	Yale	AR-Expression
Ensemble method	DOC	63.4	74.1	46.2
	One-All	96.7	97.6	95.3
	One-One	98	98.1	--

3.3. Resiliency and performance of different CNN structures

After deciding about the structure of multi-class SVM (one-all strategy), we can investigate and compare performances of different structures found by different metaheuristic algorithms. This investigation could help us to learn more about merits and demerits of each method. To that end, performance of the best structure found by GA, SA, and IC on all databases were evaluated. Table 9, shows results of this simulation.

Table 9: The average recognition rates of different CNN structures

		Databases		
		ORL	Yale	AR- Expression
Metaheuristic algorithms	GA	93.5	94.7	90.4
	SA	95.8	96.9	93
	IC	96.7	97.6	95.3

Based on results of table 9, the structure found by IC algorithm archives the highest value of recognition rates, so again we conclude that IC method is more suitable for this task.

Usually databases are compiled in controlled environments. On the other hand, in practical situations images may suffer from different factors including different qualities or partial occlusion. Consequently, systems should be evaluated in such situations. To investigate resiliency of structures found by different methods to these factors a set of experiments was carried out. In the first experiment the effect of partial occlusion on ORL database was evaluated. To that end, different portion of images were covered with a black rectangle and then performance of each structure was evaluated. Fig. 6, shows results of this experiment.

In practical situations images may have different levels of noise. To that end, in the second experiment robustness of different structures of CNN to Gaussian noise was evaluated on ORL database. Results of this simulation are shown in Fig. 7.

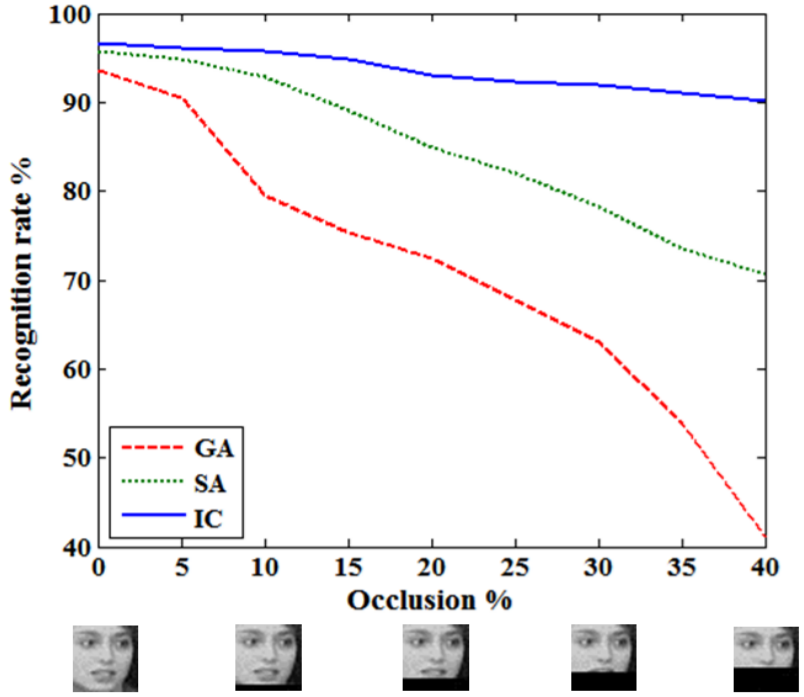

Figure 6: Effect of occlusion on recognition rate of different CNN structures

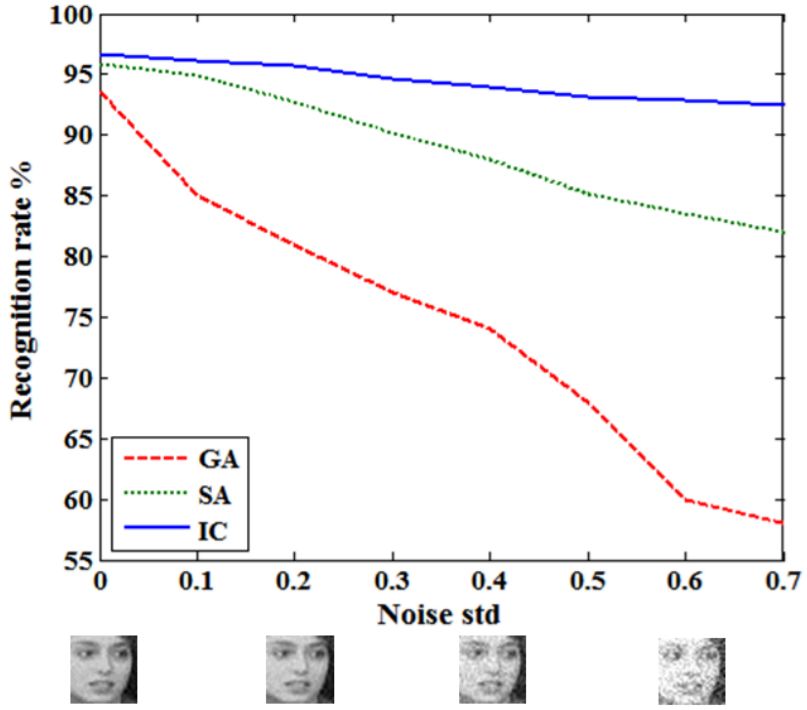

Figure 7: Effect of Guassian noise on recognition rate of different CNN structures

Comparing results of Fig. 6 and Fig. 7, shows that occlusion degrades performance of systems more severely. Also, the structure found by IC algorithm is more robust to occlusion and presence of noise.

In the first experiment effect of occlusion was simulated by covering different bottom portions

of images. To achieve more reliable results another experiment was conducted on images with real disguise. Different structures of CNN were trained with AR-Expression and then systems were tested separately on AR-Sunglasses and AR-Scarf. Results of these experiments are reflected in table 10.

Table 10: The average recognition rates of different CNN structure in presence of real disguises

		Databases	
		AR- Sunglasses	AR- Scarf
Metaheuristic algorithms	GA	73.2	52.5
	SA	82.6	68.1
	IC	90.3	85.8

Comparing results of Fig. 6 and Fig. 7, and table 10 with results of table 9 leads to an interesting conclusion. According to table 9 all structures have comparable performances on ORL, Yale, and AR-Expression databases. On the other hand, results of Fig. 6, Fig. 7 and table 10, shows that very large deviations in performances of these systems in presence of noise and occlusions exist. Based on these results we conclude that IC algorithm is a better algorithm for finding the optimum structure of CNN for face recognition.

A comparison between our proposed method and some of previous works is provided in table 11. For this aim, the proposed method was compared with methods of eigenface [3] linear discriminant regression classification (LDRC) [40], CNN with bottleneck structure [15], adaptive CNN (ACNN) [16], weighted two phase test sample sparse representation (WTPTSSR) [41], and linear collaborative discriminant regression classification (LCDRC) [42].

4. Conclusion

This work was motivated by the idea of developing a smart recognition system that is fully automated and does not need any manual tuning. To that end, convolutional neural network was employed. CNNs have the desired property of being smart. That is, CNN is capable of learning proper processes for extracting high quality features. In this fashion, manual determination of the processes was eliminated. This property makes the system more adaptable to new conditions and real-life applications that may differ considerably from academic databases of face recognition. Furthermore, to solve the problem of "optimal CNN structure", different metaheuristic algorithms were investigated. Our simulation results showed that not only IC algorithm found more optimum structures of CNN in terms of recognition rate, but it also resulted in structures which were more resilient to partial occlusion and presence of noise.

Table 11: Comparing performance of the proposed method with previous works

Method	ORL	Yale	AR- Expression
Eigenfaces	88.0	87.8	66.9
LDRC	92.3	68.0	86.0
CNN	92.6	93.3	---
ACNN	90.4	---	---
WTPTSSR	93.5	100	78.2
LCDRC	96.5	78.0	95.0
Proposed method	96.7	97.6	95.3

References

[1] Senior, A.W. and R.M. Bolle, Face Recognition and its Application, in Biometric Solutions. 2002, Springer. p. 83-97.
[2] Zhao, W., et al., Face recognition: A literature survey. Acm Computing Surveys (CSUR), 2003. 35(4): p. 399-458.
[3] Turk, M. and A. Pentland, Eigenfaces for recognition. Journal of cognitive neuroscience, 1991. 3(1): p. 71-86.
[4] Belhumeur, P.N., J.P. Hespanha, and D.J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on pattern analysis and machine intelligence, 1997. 19(7): p. 711-720.
[5] Bartlett, M.S., J.R. Movellan, and T.J. Sejnowski, Face recognition by independent component analysis. IEEE Transactions on neural networks, 2002. 13(6): p. 1450-1464.
[6] Kim, K.I., K. Jung, and H.J. Kim, Face recognition using kernel principal component analysis. IEEE signal processing letters, 2002. 9(2): p. 40-42.
[7] Bach, F.R. and M.I. Jordan, Kernel independent component analysis. Journal of machine learning research, 2002. 3(Jul): p. 1-48.
[8] Ji, S. and J. Ye, Generalized linear discriminant analysis: a unified framework and efficient model selection. IEEE Transactions on Neural Networks, 2008. 19(10): p. 1768-1782.
[9] Alam, M.A., Kernel Choice for Unsupervised Kernel Methods. 2014, The Institute of Statistical Mathematics.
[10] Suruliandi, A., K. Meena, and R.R. Rose, Local binary pattern and its derivatives for face recognition. IET computer vision, 2012. 6(5): p. 480-488.
[11] Yang, B. and S. Chen, A comparative study on local binary pattern (LBP) based face recogni-

tion: LBP histogram versus LBP image. Neurocomputing, 2013. 120: p. 365-379.
[12] Wen, Y., An improved discriminative common vectors and support vector machine based face recognition approach. Expert Systems with Applications, 2012. 39(4): p. 4628-4632.
[13] Al-Shiha, A.A.M., W.L. Woo, and S.S. Dlay, Multi-linear neighborhood preserving projection for face recognition. Pattern Recognition, 2014. 47(2): p. 544-555.
[14] Abusham, E.A. Face verification using Local Graph Stucture (LGS). in Biometrics and Security Technologies (ISBAST), 2014 International Symposium on. 2014. IEEE.
[15] Duffner, S. and C. Garcia. Face recognition using non-linear image reconstruction. in Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on. 2007. IEEE.
[16] Zhang, Y., et al., Adaptive Convolutional Neural Network and Its Application in Face Recognition. Neural Processing Letters, 2016. 43(2): p. 389-399.
[17] Taigman, Y., et al. Deepface: Closing the gap to human-level performance in face verification. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.
[18] Taigman, Y., et al. Web-scale training for face identification. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.
[19] Schroff, F., D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.
[20] Ghasemzadeh, H., M.T. Khass, and M.K. Arjmandi, Audio steganalysis based on reversed psychoacoustic model of human hearing. Digital Signal Processing, 2016. 51: p. 133-141.
[21] Glover, F.W. and G.A. Kochenberger, Handbook of metaheuristics. Vol. 57. 2006: Springer Science & Business Media.
[22] Ghasemzadeh, H., et al., Detection of vocal disorders based on phase space parameters and Lyapunov spectrum. Biomedical Signal Processing and Control, 2015. 22: p. 135-145.
[23] LeCun, Y. and Y. Bengio, Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 1995. 3361: p. 310.
[24] Firouzian, I. and Firouzian, N., 2020. Face Recognition by Cognitive Discriminant Features. International Journal of Nonlinear Analysis and Applications, 11(1), pp.7-20.
[25] Le Cun, B.B., et al. Handwritten digit recognition with a back-propagation network. in Advances in neural information processing systems. 1990. Citeseer.
[26] Ghasemzadeh, H. A metaheuristic approach for solving jigsaw puzzles. in Intelligent Systems (ICIS), 2014 Iranian Conference on. 2014. IEEE.
[27] Goldberg, D.E. and K. Deb, A comparative analysis of selection schemes used in genetic algorithms. Foundations of genetic algorithms, 1991. 1: p. 69-93.
[28] Ben-Ameur, W., Computing the initial temperature of simulated annealing. Computational Optimization and Applications, 2004. 29(3): p. 369-385.
[29] Atashpaz-Gargari, E. and C. Lucas. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. in Evolutionary computation, 2007. CEC 2007. IEEE Congress on. 2007. IEEE.
[30] Hosseini, S. and A. Al Khaled, A survey on the imperialist competitive algorithm metaheuristic: Implementation in engineering domain and directions for future research. Applied Soft Computing, 2014. 24: p. 1078-1094.
[31] Suykens, J.A., T. Van Gestel, and J. De Brabanter, Least squares support vector machines. 2002: World Scientific.
[32] Wang, L., Support vector machines: theory and applications. Vol. 177. 2005: Springer Science & Business Media.
[33] Olson, D.L. and D. Delen, Advanced data mining techniques. 2008: Springer Science & Business

Media.
[34] Niu, X.-X. and C.Y. Suen, A novel hybrid CNN–SVM classifier for recognizing handwritten digits. Pattern Recognition, 2012. 45(4): p. 1318-1325.
[35] Cortes, C. and V. Vapnik, Support-vector networks. Machine learning, 1995. 20(3): p. 273-297.
[36] Chang, C.-C. and C.-J. Lin, LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2011. 2(3): p. 27.
[37] Polat, K. and S. Güneş, A novel hybrid intelligent method based on C4. 5 decision tree classifier and one-against-all approach for multi-class classification problems. Expert Systems with Applications, 2009. 36(2): p. 1587-1592.
[38] Abe, S., Analysis of multiclass support vector machines. Thyroid, 2003. 21(3): p. 3772.
[39] Platt, J.C., N. Cristianini, and J. Shawe-Taylor. Large Margin DAGs for Multiclass Classification. in nips. 1999.
[40] Huang, S.-M. and J.-F. Yang, Linear discriminant regression classification for face recognition. IEEE Signal Processing Letters, 2013. 20(1): p. 91-94.
[41] Liu, Z., et al., Face recognition via weighted two phase test sample sparse representation. Neural Processing Letters, 2015. 41(1): p. 43-53.
[42] Qu, X., et al., Linear collaborative discriminant regression classification for face recognition. Journal of Visual Communication and Image Representation, 2015. 31: p. 312-319.