Mathematical and Computer Modelling 57 (2013) 848–856

Contents lists available at SciVerse ScienceDirect

Mathematical and Computer Modelling

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

A hybrid neural network Imperialist Competitive Algorithm for skin color segmentation

Navid Razmjooy [a], B. Somayeh Mousavi [b,*], F. Soleymani [c]

[a] Department of Electrical Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

[b] Department of Computer Engineering, Zahedan Branch, Islamic Azad University, Zahedan, Iran

[c] Department of Mathematics, Zahedan Branch, Islamic Azad University, Zahedan, Iran

ARTICLE INFO

Article history:
Received 18 November 2011
Received in revised form 14 September 2012
Accepted 19 September 2012

Keywords:
Artificial neural network
Imperialist competitive algorithm
Skin detection
Morphological processing

ABSTRACT

Skin color detection is a popular and useful technique because of its wide range of utilizations both in human computer interaction and content based analysis. Applications such as detecting and tracking of human body parts, face detection and recognition, naked people detection and people retrieval in multimedia databases all benefit from skin detection. Hence finding a proper method for segment the skin-like pixels can solve the presented problems. The Imperialist Competitive Algorithm (ICA) is a new evolutionary algorithm which was recently introduced and has a good performance in some optimization problems. The ICA was inspired by socio-political processes of imperialistic competition of mankind in the real world. In this paper a hybrid ICA-ANN is proposed for solving skin classification. A new algorithm that combines ICA and ANN to solve skin classification has been proposed, and then a hybrid Neural Network (NN)-Imperialist Competitive Algorithm (ICA) is applied to solve the classification problem. In the proposed algorithm, a multi layer perceptron network (MLP) manages the problem's constraints and an ICA algorithm searches for high quality solutions and minimum cost. The proposed color segmentation algorithm operates directly on RGB color space without the need of color space conversion. Experimental results show that this method can improve the performance of the MLP algorithm significantly.

1. Introduction

Nowadays detecting human skin regions in an image has been of growing interest and plays an important role in a wide range of image processing applications and various human computer interaction domains. Regarding color based skin detection, the advantage of using color over grayscale is due to the extra dimensions of color, i.e., two objects of similar gray tones might be very different in a color space [1]; also, the experience suggests that human skin has a characteristic color, which is easily recognized by humans [2]. A color feature is pixel based requiring no spatial context; therefore it is orientation and size invariant and fast to process. Applications such as detecting and tracking of human body parts [3], face detection [4], naked people detection and people retrieval in multimedia databases [5], all benefit from skin detection [6]. Also, color based skin detection gains attention in contributing to blocking objectionable image or video content on the Internet automatically [7]. Besides its usage in computer related technologies, skin color plays an important role for humans and human–human relations: It can serve as an indication about the well-being of a person, one can trace a person's ethnic origin and age from the skin color and it gives an indication of whether somebody has been exposed to the sun for a longer time [8].

* Corresponding author.
E-mail addresses: navid.razmjooy@hotmail.com (N. Razmjooy), bbmoosavi@gmail.com (B.S. Mousavi), fazl_soley_bsb@yahoo.com (F. Soleymani).

0895-7177/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2012.09.013

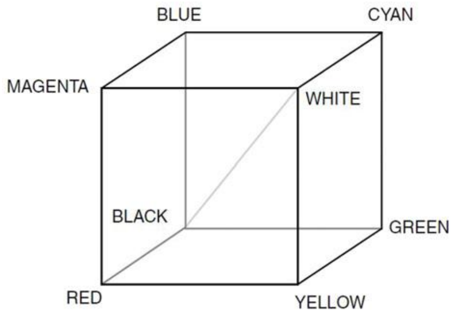

Fig. 1. The RGB color cube.

When building a system to segment skin color, two main steps are necessary: (1) to represent the image pixels in a suitable color space, (2) to select a suitable classifier.

Over the last decades, Artificial Neural Networks have shown a good potency for modeling complicated systems [9]. The classification section is an important step of skin detection which needs a proper insulator between skin and environment. The most widely used neural network model is the multi-layer percepton (MLP), in which the connection weight training is normally completed by a back propagation learning algorithm [10].

As the appointment of network structure and parameters are most significant, some evolutionary algorithms such as Genetic Algorithm (GA) [11], Back Propagation (BP) [12], Pruning Algorithm [13], Simulated Annealing [14] and Particle Swarm Optimization [15] have shown a significant role in this regard.

These evolutionary algorithms can conclude a neural network at different levels such as weight training, architecture adaptation (for determining number of hidden layers, number of hidden neurons and node transfer functions) and learning rules [15].

There are some inevitable drawbacks of the presented approaches, which need high performance algorithms.

In this paper we focused only on the weight optimization of ANN, and propose an evolutionary algorithm to a good classifier for classification in skin purposes; for this case, we employ a new evolutionary algorithm for optimizing the weights of Multilayer Perceptron (MLP) ANNs called the Imperialist Competitive Algorithm. This optimization algorithm is inspired by imperialistic competition which has shown great performance in both convergence rate and better global optima achievement [16–19].

In the present work, we propose ICA for optimizing the initial weights of the feed-forward neural network. The simulation results demonstrate the effectiveness and potential of the new proposed network for skin classification compared with BP neural network using the same observed data.

In order to reduce random variation of the proposed algorithm itself, each experiment has been run 15 times and the mean is presented. The method is tested against ANNs.

2. RGB color space

A color space is an abstract mathematical model representation of a set of colors. All of the color spaces can be derived from the RGB information supplied by devices such as camera and scanners.

A wide variety of them have been applied to the problem of skin color modeling. This paper uses an RGB type of color spaces. This color space because of its simplicity is widely used through computer vision purposes. Red, green and blue are three primary additive colors and are represented by a three-dimensional, Cartesian coordinate system; Fig. 1 shows this representation.

The indicated diagonal of the cube, with equal amounts of each primary component, represents various gray levels. By using this color space, RGB color values of the input image do not need to be converted into another color of coordinates. We have observed an important objectivity, that skin colors in the RGB color space are approximately distributed in a linear fashion [20,21].

In [22] the RGB color space has been used for skin detection. A sample of 3d plot for skin, and non-skin pixels in this space is shown in Fig. 2:

3. Skin supervised classification

Once the RGB color space has been selected as the skin features, it is significant to design a classifier to categorize the skin pixels. Image classification is a useful area in image processing and machine learning, in which attempts to label an image with proper identifiers. The classification system is divided into two main parts: the training and testing stage.

The training sets are chosen manually to demonstrate a common picture set of that class. The classifier method then analyzes the training set, producing a descriptor for that specific class based on the specific features of the training set. This

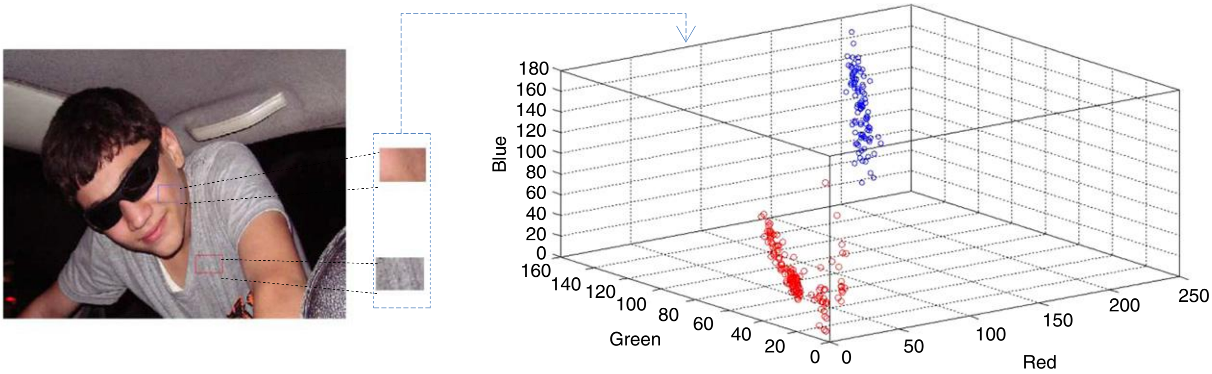

Fig. 2. Analysis of skin colors in the RGB color-space: 3D-plot of skin (blue), 3D-plot of non-skin (red); The skin colors approximately distributed in a linear fashion in the RGB color space. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

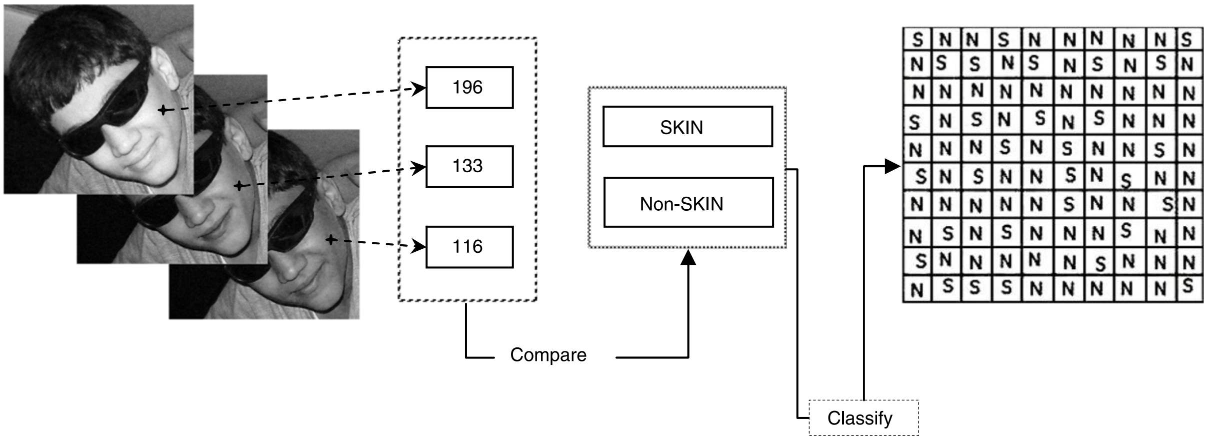

Fig. 3. Steps in supervised classification.

descriptor could then be utilized on other images of the desired database, which specifies if the selected image has similarity by that class [23]. After the training stage, the feature vectors of a test image works as input; this stage has been known as the testing stage. A classifier determines on the bases of learning model, with its own classification rules, as to which class that feature vector belongs. Fig. 3 shows the steps of classification.

4. Artificial neural network

An artificial neural network (ANN) can be briefly represented as the programming equivalent of a brain. It tries to simulate the function of a brain by imitating the layout. An ANN contains several neurons which are interconnected. Each connection between the neurons has a specific weight which efficacy how much the output from the neuron will affect the input to the next neuron. Each neuron typically also has a weight of its own called a bias term which determines the effect of the neuron itself. The information in an ANN is stored in its weights, training an ANN; hence requires the weights to be determined. There are several existing algorithms that deal with this process.

Back propagation is a common method that is used for feed forward networks. It computes the error on each of the training pairs and adjusts the weights to fit the desired output. This is done in several epochs until the total error on the training set become small enough or when the error stops to decline. Indeed, it employs supervised learning in which the network is trained using data for which inputs as well as target outputs are noticed. Once trained, the network weights are vapid and can be used to calculate output values for new given samples.

BP is a gradient descent algorithm on the error space, which most likely gets trapped into a local minimum, the success of which is entirely dependent on initial (weight) settings. This shortcoming can be removed by an exploration searching ability of the evolutionary algorithms such as ICA.

5. Imperialist competitive algorithm

The Imperialist Competitive Algorithm (ICA) is a new evolutionary optimization algorithm which is derived by imperialistic competition. ICA, like other evolutionary algorithms commences with an initial population which is known

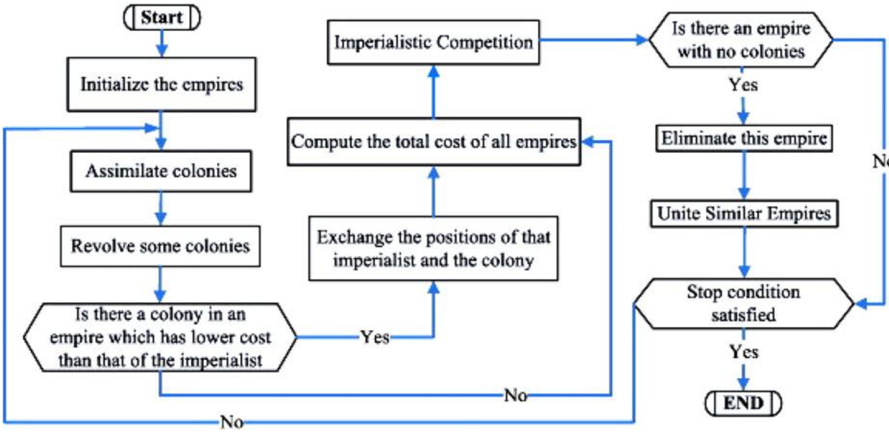

Fig. 4. Flowchart of Imperialist Competitive Algorithm.

as the country; a country contains types of: colonies and imperialists which together form empires [24]. Indeed, imperialist countries try to overcome other countries and turning them to their colonies. Also, imperialist countries compete strongly with each other for taking occupancy of other countries; Imperialistic competition among these empires forms the proposed evolutionary algorithm. During this competition the weakest empires collapse and stronger ones will get more potency [25].

Imperialistic competition converges to a situation in which there exists only one empire and colonies have the same fitness function and power value as the imperialist, as a fraction of colonies can be replaced by the imperialist.

The pseudo code of the Imperialist Competitive Algorithm is introduced as:

1. Select some random points on the function and initialize the empires.
2. Move the colonies toward their relevant imperialist (Assimilation).
3. Randomly change the position of some colonies (Revolution).
4. If there is a colony in an empire which has lower cost than the imperialist, exchange the positions of that colony and the imperialist.
5. Unite the similar empires.
6. Compute the total cost of all empires.
7. Pick the weakest colony (colonies) from the weakest empires and give it (them) to one of the empires (imperialistic competition).
8. Eliminate the powerless empires.
9. If stop conditions are satisfied, stop, if not go to 2.

The flowchart of the Imperialist Competitive Algorithm is shown in Fig. 4 [26].

In ICA, the epithet country indeed is an array of variables that should be optimized by finding the final goal. In such a condition a country is an $1 \times N$ array which is used to optimize an N dimensional problem. The floating point numbers are applied to the variable values of the country.

From the above, the algorithm starts with some initial countries which are randomly distributed in search space. n number of stronger countries (countries with lower cost) are selected to be the imperialists and the others are divided among them based on their power. The initial number of colonies of each imperialist should be directly commensurate to its normalized power.

After dividing all colonies among imperialists and creating the initial empires, these colonies commence moving into their relevant imperialist territory which is based on assimilation policy [25]. Fig. 5 shows the motion of a colony into its relevant imperialist. In this motion, θ and x are random numbers with uniform distribution as demonstrated in formula (1, 2) and d is the distance between the colony and the imperialist.

$$x \sim U(0, \beta \times d), \tag{1}$$

$$\theta \sim U(-\gamma, \gamma) \tag{2}$$

where β is a positive number less than 2, d is the space between the imperialist and its colony and orders the derivation from the original direction; In this paper β and γ are considered as 2 and 0.5 respectively.

Fig. 6 shows that if this motivation causes to find a colony with a better situation (lower cost) rather than its imperialist, the position of the colony and imperialist change together.

The total power of an empire depends on both the power of the imperialist country and the colonies and is defined by:

$$T \cdot C_n = \text{Cost}(\text{imperialist}_n) + \xi \text{mean}\{\text{Cost}(\text{colonies} - \text{of} - \text{empires}_n)\}. \tag{3}$$

In imperialistic competition, all imperialists try to take the assets of other imperialists and develop their own power.

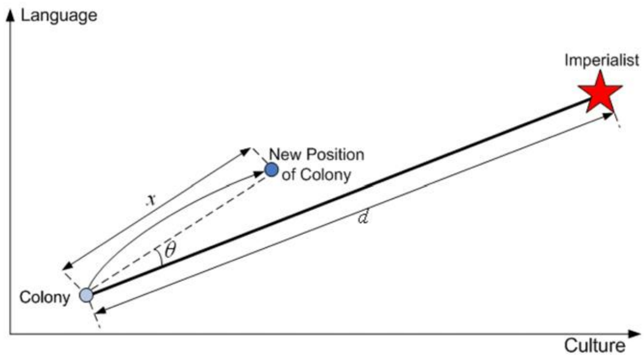

Fig. 5. Colonies' movement into their relevant imperialist.

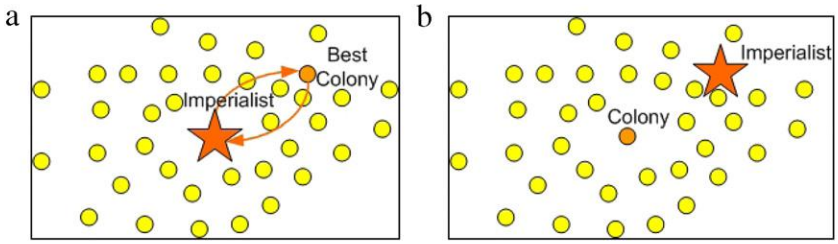

Fig. 6. (a) Imperialist and colony position change, (b) total empire after applying changes.

In ICA, this fact is defined as taking the total power of an empire by the sum of imperialist state power and a percentage of the mean power of its colonies. The total power of an empire is defined as:

$$P_r = P_{\text{im}} + \text{mean}\{\text{power}(\text{colonies})\}, \tag{4}$$

where P_{im} is the power of the imperialist and e is a positive number less than 1.

In such a condition, powerless empires which can't compete, increase their power and those which can't detain diminution of their power will be collapsed.

Imperialistic competition and the motion of colonies into their relevant imperialist will hopefully make a new world with one empire such that all the colonies and even imperialists themselves have the same position and power. In this term the imperialist contains an array of variables which is an optimal resolution of the problem.

6. ANN weights evolution using ICA (HNNICA)

Selecting the optimal value for weights can be formulated as an exploration search problem wherein the architecture of the neural network is reconstructed and fixed during the evolution. The value of the weights may be described as being trained with certain length and the whole network is encoded by interpolation of all weight values of the network in the chromosome. A heuristic concerning the order of the interpolation is to put connection weights in the same node together.

Evolutionary search of weights values can be formulated as follows:

(1) Generate an initial population of N weight chromosomes. Evaluate the fitness of each EANN depending on the problem.
(2) Depending on the fitness and using suitable selection methods reproduce a number of children for each individual in the current generation.
(3) Apply genetic operators to each child individual generated above and obtain the next generation.
(4) Check whether the network has achieved the required error rate or the specified number of generations has been reached then goes to step 2.
(5) End.

In order to express the ANN, consider a two layered network which is formulated as formula (5):

$$F = \sum_{i=1}^{H} w_i \sigma \left(\sum_{j=1}^{d} w_j x_j + b \right) \tag{5}$$

where H denotes the number of neurons in the hidden layer, w denotes the weights of the network, b denotes the bias value and σ denotes the activation function of each neuron which in this case is considered as sigmoid.

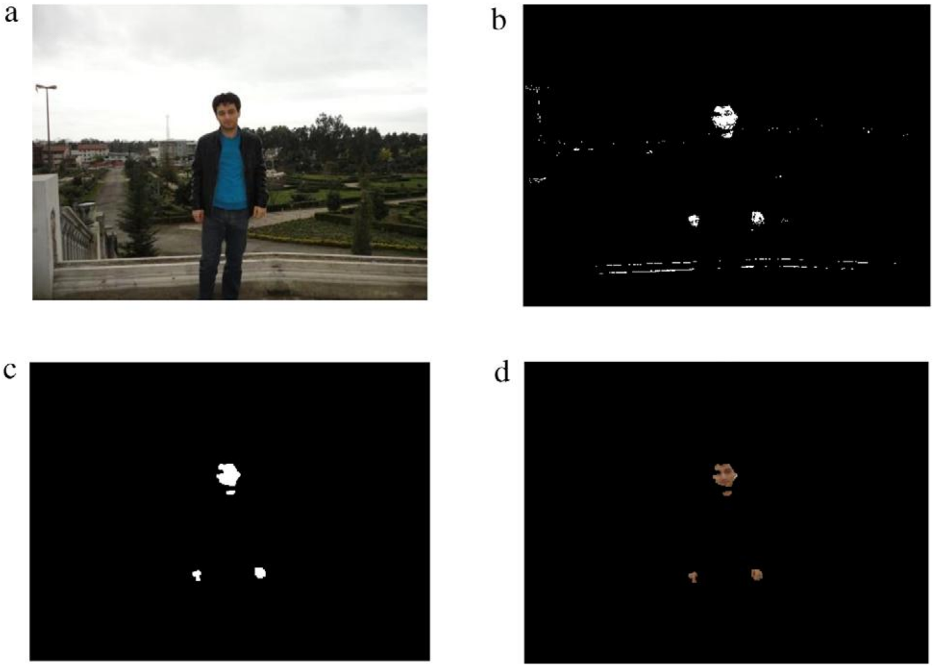

Fig. 7. (a) Original image, (b) skin regions before morphological processing, (c) skin regions after morphological processing, (d) detected skin area.

The network is trained by performing optimization the value of the weights for each node interconnection and bias terms; until the values' outputs at the output layer neurons are as close as possible to the actual outputs. The mean squared error of the network (MSE) can be defined as:

$$\mathrm{MSE} = \frac{1}{2}\sum_{k=1}^{g}\sum_{j=1}^{m}\left(Y_j(k) - T_j(k)\right)^2 \tag{6}$$

where m is the number of output nodes, g is the number of training samples, $Y_j(k)$ is the desired output, and $T_j(k)$ is the real output.

7. Dataset description

We used Bao dataset [27] to show our proposed method; this database includes 370 face images from various races, mostly from Asia, with a wide range of size, lighting and background.

8. Experiments and results

In this study, a two area classification (as skin or non-skin individually) is used for skin detection; *pixel-based* classification is selected to classify each skin and non-skin pixel, independently from its neighbors. In these applications, the search space for objects of interest, such as faces or hands, can be reduced through the detection of skin regions. To this end, skin segmentation is very effective because it usually involves a small amount of computation and can be done regardless of pose. The classifier includes a vector of $3 \times n$ pixel coefficient vectors from each image, either a skin or non-skin image, where n is the number of neurons in the hidden layer: since the transfer function employed is a sigmoid function, the output will be generated between 0 and 255 (uint8 mode). Therefore, the output of the neural classifier needs to be improved so that it is either 0 or 1 [23]. For finally characterizing the skin and non-skin classes, a single threshold is utilized as if the neural network output is higher than the selected threshold point, then the output is 1. Otherwise, the output is 0. The threshold value used in this work is 128 [23]. As the post processing stage, morphological operations contain: filling holes and opening followed by closing is implemented to the result images [28]. Fig. 7 shows the effect of morphological processing.

To evaluate the performance of the proposed algorithm toward gradient descent, three performance metrics are defined. The first metric is the correct detection rate *(CDR)* and is given in Eq. (7). The false acceptance rate *(FAR)* is the percentage of identification moments in which false acceptance occurs. The false rejection rate *(FRR)* is the percentage of identification moments in which false rejection occurs. The FAR and FRR are expressed in Eqs. (8) and (9), respectively:

$$\mathrm{CDR} = \frac{\text{No. of Pixels Correctly Classified}}{\text{Total Pixels in the Test Dataset}} \tag{7}$$

Fig. 8. Test set: (a) skin sets, (b) non-skin sets.

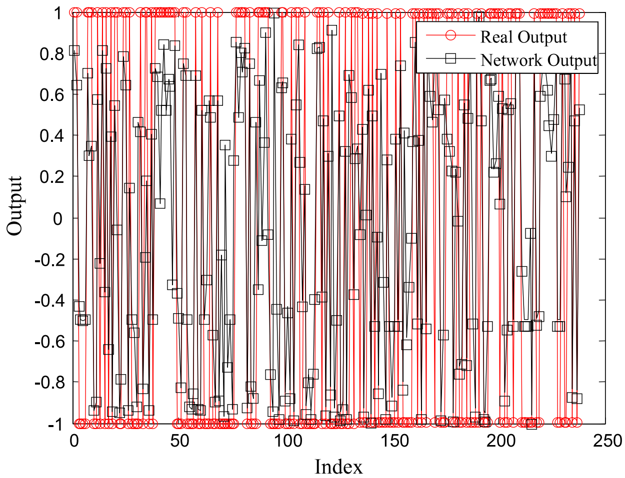

Fig. 9. Real and network output for train data.

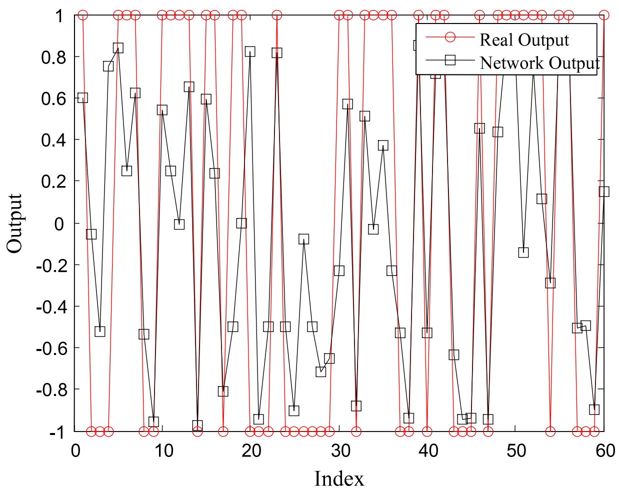

Fig. 10. Real and network output for test data.

$$\mathrm{FAR} = \frac{\text{No. of Non-Skin Pixels Classified as Skin Pixels}}{\text{Total Pixels in the Test Dataset}} \tag{8}$$

$$\mathrm{FRR} = \frac{\text{No. of Skin Pixels Classified as Non-Skin Pixels}}{\text{Total Pixels in the Test Dataset}}. \tag{9}$$

Fig. 8 shows the test set from the faces and non-faces to measure metric performance.

Typically, a generalization of the least mean square algorithm (that improves network weights) to minimize the mean squared error (between the target and actual outputs of the network) is applied to the back propagation algorithm. Table 1 shows obtained results in the Bao dataset [27], which includes 370 face images from various races, mostly from Asia, with a wide range of size, lighting and background. From the experimental results, it can be seen that the HNNICA performed better and gave a higher correct detection rate.

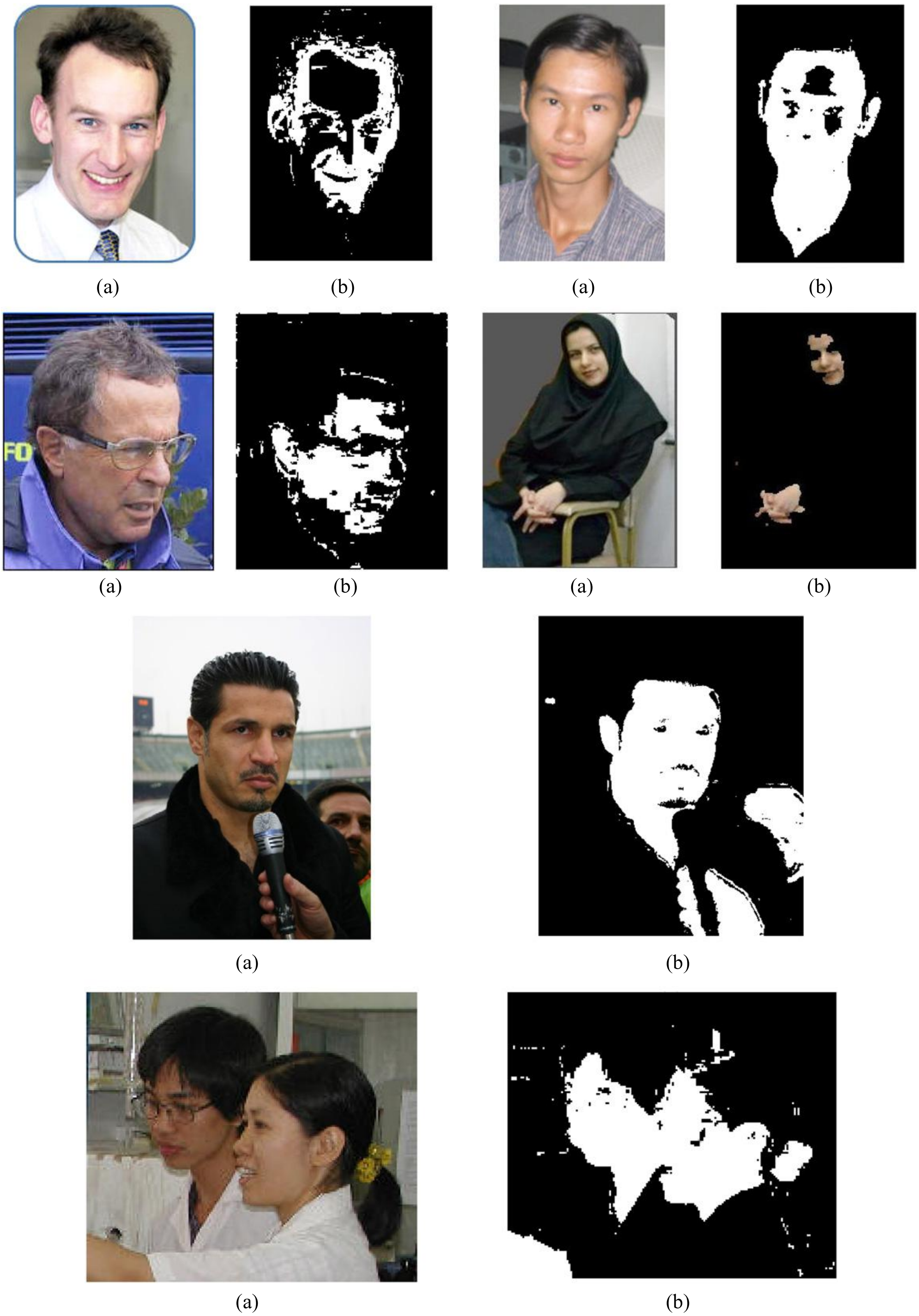

Fig. 11. Sample images from the skin detection database. (a) Original image, (b) skin segmented result.

Table 1
Skin detection performance for HNNICA and MLP.

Algorithm	CDR	FRR	FAR
HNNICA	**70.84**	**4.16**	**25**
MLP	68.42	5.2	26.38

Figs. 9 and 10 show the extent of the match between the measured and predicted skin rate of the train set and validation phases by ICA–ANN in terms of a scatter diagram.

Fig. 11 shows sample outputs of the HNNICA classifier with the pixel based technique. The figure shows that the HNNICA classifier can successfully identify exposed skin regions including face, hands, and neck.

9. Conclusion

An evolutionary approach based on skin detection was presented. The Imperialist Competitive Algorithm employed to decrease the mean square error (HNNICA) instead of using gradient descent in neural networks. We used RGB color space to decrease computations; this algorithm reliably reduced the number of false detections toward the gradient descent algorithm.

The skin detection method based on HNNICA is a promising approach given its satisfactory results. However, more features with a sophisticated algorithm need to be added in order to use it for more general applications.

References

[1] E. Hjelmas, B.K. Low, Face detection, a survey, Computer Vision and Image Understanding (2001) 236–274.
[2] R.-L. Hsu, M. Abdel-Mottaleb, A.K. Jain, Face detection in color images, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 696–707.
[3] Antonis A. Argyros, Manolis I.A. Lourakis, Real-time tracking of multiple skin colored objects with a possibly moving camera, in: ECCV, 2004, pp. 368–379.
[4] J. Cai, A. Goshtasby, Detecting human faces in color images, Image and Vision Computing 18 (1999) 63–75.
[5] Silvio Borer, Sabine Suesstrunk, Opponent color space motivated by retinal processing, in: International Conference on Color in Graphics, Imaging and Vision, CGIV, 2002, pp. 187–189.
[6] P. Kakumanu, S. Makrogiannis, N. Bourbakis, A survey of skin-color modeling and detection methods, Pattern Recognition 40 (3) (2007) 1106–1122.
[7] Julian Stottinger, Allan Hanbury, Christian Liensberger, Rehanullah Khan, Skin paths for contextual aging adult videos, in: International Symposium on Visual Computing, 2009, pp. 303–314.
[8] Moritz Storring, Computer vision and human skin color, Ph.D. Thesis, Alborg University, Denmark, 2004.
[9] D. Guyer, X. Yang, Use of genetic artificial neural networks and spectral imaging for defect detection on cherries, Computers and Electronics in Agriculture 29 (2000) 179–194.
[10] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by backpropagating errors, Nature 323 (1986) 533–536.
[11] X. Qu, J. Feng, W. Sun, Parallel genetic algorithm model based on AHP and neural networks for enterprise comprehensive business, in: IEEE Intl. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, 2008, pp. 897–900.
[12] P. Tang, Z. Xi, The research on BP neural network model based on guaranteed convergence particle swarm optimization, in: Second Intl. Symp. on Intelligent Information Technology Application, IITA'08, vol. 2, December 2008, pp. 13–16.
[13] R. Reed, Pruning algorithms—a survey, IEEE Transactions on Neural Networks 4 (1993) 740–747.
[14] M.C.P. de Souto, A. Yamazaki, T.B. Ludernir, Optimization of neural network weights and architecture for odor recognition using simulated annealing, in: Proc. 2002. Intl. Joint Conf. on Neural Networks, vol. 1, 2002, pp. 547–552.
[15] C. Zhang, H. Shao, Y. Li, Particle swarm optimization for evolving artificial neural network, in: 2000 IEEE Intl. Conf. on Systems, Man, and Cybernetics, vol. 4, October 2000, pp. 2487–2490.
[16] A. Biabangard-Oskouyi, E. Atashpaz-Gargari, N. Soltani, C. Lucas, Application of imperialist competitive algorithm for materials property characterization from sharp indentation test, The International Journal of Engineering Simulation (2008) (in press).
[17] R. Rajabioun, F. Hashemzadeh, E. Atashpaz-Gargari, B. Mesgari, F. Rajaei Salmasi, Identification of a MIMO evaporator and its decentralized PID controller tuning using Colonial Competitive Algorithm, Presented in IFAC World Congress, 2008 (in press).
[18] E. Atashpaz-Gargari, F. Hashemzadeh, R. Rajabioun, C. Lucas, Colonial competitive algorithm, a novel approach for PID controller design in MIMO distillation column process, International Journal of Intelligent Computing and Cybernetics 1 (3) (2008) 337–355.
[19] H. Sepehri Rad, C. Lucas, Application of imperialistic competition algorithm in recommender systems, in: 13th Int'l CSI Computer Conference, CSICC'08, Kish Island, Iran, 2008.
[20] Leonid Sigal, Stan Sclaroff, Vassilis Athitsos, Estimation and prediction of evolving color distributions for skin segmentation under varying illumination, in: Proceedings of IEEE Conference on Computer Society Conference on Computer Vision and Pattern Recognition, 2000, pp. 152–1594.
[21] Rogerio Schmidt Feris, Teofilo Emidio de Campos, Roberto Marcondes Cesar Jr., Detection and Tracking of Facial Features in Video Sequences, Springer-Verlag Press, 2000, pp. 197–206, MICAI-2000, 1793.
[22] J. Brand, J. Mason, A comparative assessment of three approaches to pixel level human skin-detection, in: Proc. of the International Conference on Pattern Recognition, vol. 1, 2000, pp. 1056–1059.
[23] N. Razmjooy, B.S. Mousavi, F. Soleymani, A real-time mathematical computer method for potato inspection using machine vision, Computers and Mathematics with Applications 63 (2012) 268–279.
[24] E. Atashpaz Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspires by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, Singapore, 2007.
[25] A.M. Jasour, E. Atashpaz, C. Lucas, Vehicle fuzzy controller design using imperialist competitive algorithm, in: Second First Iranian Joint Congress on Fuzzy and Intelligent Systems, Tehran, Iran, 2008.
[26] H. Duan, Chunfang Xu, Senqi Liu, Shan Shao, Template matching using chaotic imperialist competitive algorithm, Pattern Recognition Letters (2009) http://dx.doi.org/10.1016/j.patrec.2009.12.005.
[27] R. Frischholz, Bao face database at the face detection homepage. Available: http://www.facedetection.com.
[28] R. Davies, Machine Vision, Morgan Kaufman, 2004.