

Skin Color Segmentation Based on Artificial Neural Network Improved by a Modified Grasshopper Optimization Algorithm



Navid Razmjoooy , Saeid Razmjoooy , Zahra Vahedi, Vania V. Estrela , and Gabriel Gomes de Oliveira

Abstract One of the applications of image processing and computer vision is to detect the skin regions for a wide range of human–computer interaction and content based utilizations. Detecting nude parts in the movies, face detection, tracking of human body parts, and people recognizing in multimedia databases are a small part of its applications. Therefore, designing an efficient technique for skin area detection can help a lot to the determined applications. The grasshopper optimization algorithm is a new optimization algorithm evolutionary algorithm which is recently introduced to solve optimization problems. The main purpose of this paper is to propose a newly developed version of this algorithm to optimize the weights of the backpropagation neural network to design a good segmentation tool for skin area segmentation. The method has been compared with the traditional multi-layer perception and the ICA-MLP to declare the proposed method’s efficiency.

Keywords Skin segmentation · Artificial neural network · Grasshopper optimization algorithm · Modified · Mathematical morphology

N. Razmjoooy (✉)

Department of Electrical and Control Engineering, Tafresh University, 39518 79611 Tafresh, Iran
e-mail: navid.razmjoooy@hotmail.com

S. Razmjoooy

Department of Biosystems Engineering, Faculty of Agriculture Technology and Natural Resources, University of Mohaghegh Ardabili, Ardabil, Iran
e-mail: saeidrazmjoooy@gmail.com

Z. Vahedi

Department of Electrical Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran
e-mail: zahra23vahedi@gmail.com

V. V. Estrela

Department of Telecommunications, Fluminense Federal University (UFF), Rio de Janeiro, Brazil
e-mail: vania.estrela.phd@ieee.org

G. G. de Oliveira

State University of Campinas, UNICAMP, São Paulo, Brazil
e-mail: oliveiragomesgabriel@ieee.org

© Springer Nature Switzerland AG 2021

N. Razmjoooy et al. (eds.), *Metaheuristics and Optimization in Computer and Electrical Engineering*, Lecture Notes in Electrical Engineering 696,
https://doi.org/10.1007/978-3-030-56689-0_9

1 Introduction

The science of image processing is one of the most useful sciences in engineering applications and has been extensively studied and researched in the field for many years. The speed of these advances has been so rapid, that now, after a short period of time, the impact of image processing can be clearly seen in many sciences and industries. While some of these applications are dependent on image processing, they cannot be used without it. As the science of image processing is integrated and specialized in today's world, it is becoming more and more important. Recently, human skin area detection has found a wide range of applications in image processing. This field has several utilizations in the human–computer interaction domains [1]. Applications such as face detection [2], detecting and tracking of human body parts [3], naked people detection [4], 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 images or video content on the Internet automatically [7].

Using the color-based skin detection has an advantage toward grayscale-based one due to the extra dimensions of color such that it may happen a situation in which two objects have similar gray tones but different color space textures. This shows that human skin has a feature for easily recognizing the humans [2]. The color-space features are pixel-based characteristics that need no spatial context which improves its orientation and size invariant and the processing period.

In the recent decade, the applications of artificial neural networks have been increasing exponentially due to its potency in modeling complicated systems [6–8]. Using an artificial neural network (ANN) classifiers for skin-like pixels segmentation is a principal step of skin detection that requires a suitable insulator between the skin and the environment. Among different types of ANNs, a multi-layer perceptron (MLP) network have a wide application due to its simplicity and good efficiency. The connection weights and biases of the MLP networks have been trained normally by a backpropagation learning algorithm [9].

The backpropagation learning algorithm is a classic learning algorithm which is so complicated for big networks. It also has a big problem with trapping in the local minimum to find the minimum error. Several approaches have been used in solving this problem. One of the useful approaches is to use meta-heuristic algorithms. There are different types of meta-heuristics such as Genetic Algorithm (GA) [10], world cup optimization algorithm (WCO) [11], variance reduction of Gaussian distribution (VRGD) algorithm [12], humpback whale optimization (HWO) algorithm [13], Emperor penguin optimizer (EPO) [14]. Meta-heuristic algorithms can improve the ANN efficiency by applying to the different levels such as weight training, learning rules, architecture adaptation (for determining the number of hidden layers, and a number of node transfer functions and hidden neurons).

This study focuses on the weight optimization of ANN to propose an optimized algorithm for better classification of the skin pixels detection. To do so, a new improved meta-heuristic technique called modified grasshopper optimization

algorithm has been utilized for optimizing the weights of Multilayer Perceptron (MLP).

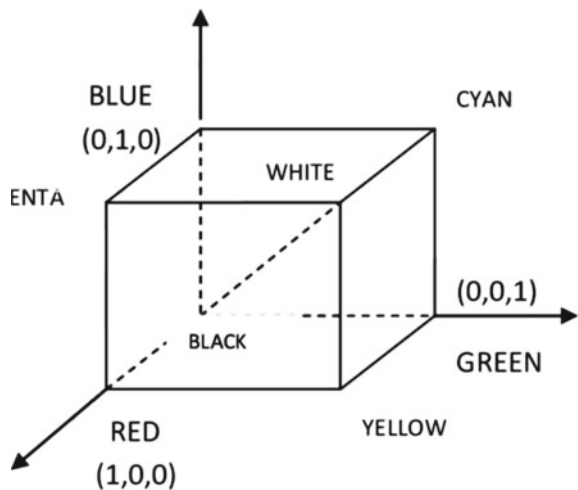
2 RGB Color Space

The purpose of selecting a color model is to facilitate the identification of colors in a standard. In essence, the color model defines a three-dimensional and sub-spatial coordinate system within that system in which each color system is expressed by a single point. Most color models now in use tend toward hardware such as monitors and color printers or applications that aim to work with color, such as producing color graphics for animation. RGB model (blue, red, green) is one of the most common hardware-oriented models for color screens. This model is based on the Cartesian coordinate system. Because of the cube's favorite color space is the image below. In the RGB model, the gray range from black to white is along the line and origin of these two points, and the other colors are points on or inside the cube defined by vectors passing through the origin. To simplify the model, it is assumed that all color values are aligned such that the image cube is below the unit cube, meaning that all values of G, R and B are in the range of [0, 255].

Each image in the RGB color model has three separate pages, each page being the primary color. When these three pages are given to the RGB display, they are combined on the phosphor screen to produce a color image.

So when the images themselves are naturally expressed in terms of three color pages, it makes sense to use the RGB model for image processing. Also, most color cameras use this format for digital imaging, which makes the RGB model an important model in image processing. As can be seen in Fig. 1, the RGB color space is a

Fig. 1 The RGB color cube



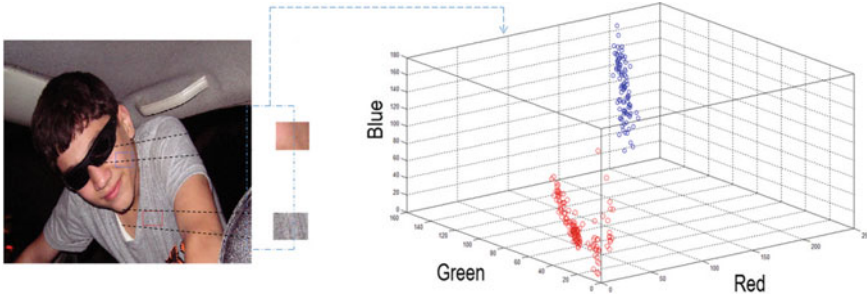


Fig. 2 Analysis of skin colors in the RGB color-space: 3D-plot of skin (blue), 3D-plot of non-skin (red); the skin colors approximately distributed in a linear fashion in the RGB color space

single cube and each axis represents one of the primary colors. The coordinate origin is where the cube lacks three primary colors and this dot represents black, while the opposite vertex is the combination of three primary colors, and this dot in the cube represents white.

The other vertices represent the cyan, purple and yellow secondary colors, each of which comes from a combination of two primary colors. In this model, all other colors within the cube are specified by three components, each point specifying the number of primary colors needed to create a particular color. Each of these components is usually specified by one byte. For example, light red has the value $[255, 0, 0]$ and light yellow the value $[255, 255, 0]$. Since red, blue and green can be independently assigned, 2563 colors can be produced with three main colors. In the image file, three pages, or rather three matrices, each page is the same color, is used to store each image. Since each pixel is indicated by three bytes, the image depth is 24 bits, and the size of each color image is three times the size of a gray-level image. RGB color images are also known as true color or 24-bit images. Figure 1 shows the RGB Color Cube.

A sample of the 3D plot for skin and non-skin pixels in this space is shown in Fig. 2.

3 Skin Color Feature Extraction

The main application of machine learning algorithms is to extract or learn from existing data. Like a child who learns by observing people and parental recommendations. For example, he or she can figure out whether the heater is hot or should not go down the street alone. In fact, machine learning algorithms work to find a summarize of data or to actually create a model of data in any way. As you know, a model always contains a summary of the data. The World Map, for example, is a model of the world that models countries and roads without much detail (for example, what shop is on the street).

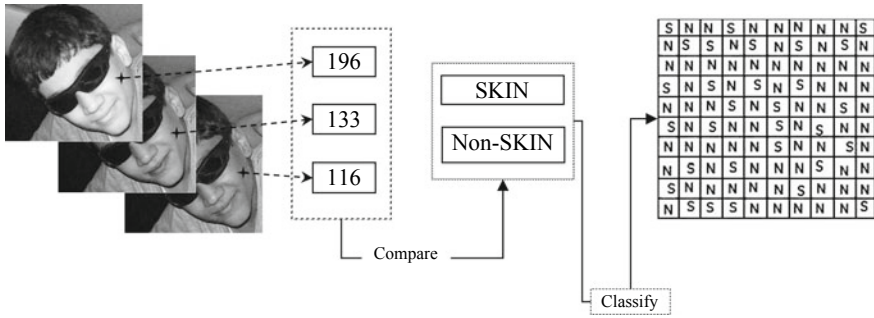


Fig. 3 Steps in supervised classification

Supervised classification is a type of machine learning in which input and output are specified and there is a so-called supervisor that provides information to the learner, thereby trying to learn a function from input to output. After extracting the RGB color information from the input image for skin region detection, it should be used as input data for the classifier to the classification. Image classification is a principal tool in image processing with different applications. This tool can be also utilized for image segmentation which is our purpose in this study. The classifier system divides the input data into two parts of the training and testing categories. In this study, 60% of data has been utilized for training and the 40% remained data have been selected as the testing set. Figure 3 shows the steps of classification.

4 The Improved Grasshopper Optimization Algorithm

4.1 The Standard Grasshopper Optimization Algorithm

The grasshopper optimization algorithm (GOA) is a new swarm-based optimization algorithm that is derived from grasshopper insect behavior [15].

In this algorithm, the population includes a collective of grasshoppers which is called a swarm. Each member of the swarm is a probable solution to the problem. The first step is started by generating a random swarm as the initial solution to the problem [16]. Then, the cost of each grasshopper is determined by obtaining the value of the cost function.

The process is continuous by absorbing the swarm via considered grasshoppers into their location to attract the grasshoppers to move into the considered grasshopper. Here, two main behaviors of the grasshoppers are considered: the tiny and slow movement of the larval grasshoppers in order to the long-range and no sequence movement drive adults and food searching process which is divided into two separate parts of exploitation and exploration.

The development of the i th grasshopper next to the target grasshopper is defined by P_i and is obtained as below:

$$P_i = R_1 SA_i + R_2 GF_i + R_3 WA_i \quad (1)$$

where, GF_i describes the gravity force on the i th grasshopper, WA_i is the wind advection, and SA_i describes the social interaction, and R_1 , R_2 , and R_3 are random constants in the interval $[0, 1]$.

The social interaction of the i th grasshopper (SA_i) related to the social forces between two grasshoppers and is a repulsion force to stop collisions and an attraction force purpose over a small length scale.

$$SA_k = \sum_{\substack{l=1 \\ l \neq i}}^N SA(D_{lk}) \hat{D}_{lk} \quad (2)$$

$$D_{lk} = X_k - X_l \quad (3)$$

$$\hat{D}_{lk} = \frac{X_k - X_l}{D_{lk}} \quad (4)$$

where, D_{lk} describes the length of the Euclidian of the k th with the l th position grasshopper, and \hat{D}_{lk} represents the present unit vector between the l th grasshopper and the k th grasshopper.

The intensity attraction strength is modeled by the following formula:

$$X_k^d = c \left(\sum_{\substack{l=1 \\ l \neq i}}^N c \frac{U_B^d - L_B^d}{2} SF(|X_l^d - X_k^d|) \frac{X_l - X_k}{D_{lk}} \right) + \hat{T}_d \quad (5)$$

where, SF determines the strong point of the social forces that are evaluated by the following:

$$SF(R) = f_i e^{-R/L} - e^{-R} \quad (6)$$

where, L describes the length scale of attraction, and f_i is the intensity force of attraction.

GF is another parameter of the algorithm that is achieved as follows:

$$GF_i = -GF \hat{e}_g \quad (7)$$

where, G_i describes a constant for the gravity force, and \hat{e}_g is a direction for unity vector alongside the wind.

The wind advection model (WA_i) is finally obtained as follows:

$$WA_i = U\hat{e}_g \tag{8}$$

where, U describes the drift constant.

The gravity assumptions and wind effects can be formulated as follows:

$$x_l^d = c \left(\sum_{\substack{l=1 \\ l \neq 1}}^N c \frac{UB^d - LB^d}{2} SF(|X_k^d - X_l^d|) \frac{X_l - X_k}{D_{lk}} \right) + \hat{T}_d \tag{9}$$

$$c = c_{\max} - l \frac{c_{\max} - c_{\min}}{L} \tag{10}$$

where, N is the grasshoppers quantity, LB_d and UB_d are the lower and the upper limitations in the d th dimension, respectively, \hat{T}_d acts objective magnitude of d th dimension by the objective grasshopper, c describes the reducing factor to the proper area of the repulsion region and attraction region, c_{\max} and c_{\min} are the highest value and the lowest value of the factor c , respectively, and finally, l and L are the current iteration and the total number of iteration, respectively.

4.2 Improved Grasshopper Optimization Algorithm Based on Chaos Theory

In this subsection, a technique is proposed for modifying the essential parameters of the Grasshopper optimization algorithm to improve its convergence speed. The keys parameters of Grasshopper optimization algorithm convergence are R_1 , R_2 , R_3 , and U . the modification is applied based on chaos theory.

Chaos theory is the study of unpredictable and random processes. The main idea of the chaos theory is to study the highly sensitive dynamic systems that can be affected by any small variations.

By the explanations above, a large diversity can be made for the swarm generation in GOA to improve the diversity of the algorithm.

This part improves the GOA algorithm capability from the point of the convergence speed and also for escaping from falling into the local optimal point [17, 18]. A general form for the chaos theory is formulated below:

$$CM_{i+1}^j = f(CM_i^j) \tag{11}$$

$$j = 1, 2, \dots, k$$

where, k describes the map dimension, and $f(CM_i^j)$ is the chaotic model generator function.

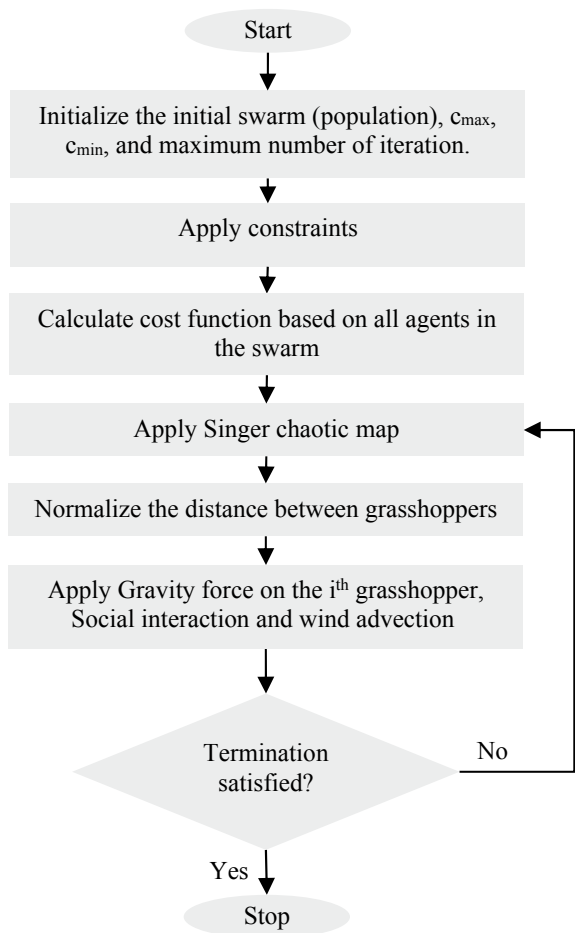
In the presented GOA, the parameters R_1, R_2, R_3 , and U are modeled based on the Singer mechanism chaotic map as follows:

$$\begin{aligned}
 R_{(j,i+1)} &= 1.07 \times (7.9R_{(j,i)} - 23.3R_{(j,i)}^2 + 28.7R_{(j,i)}^3 - 13.3R_{(j,i)}^4) \\
 U_{(i+1)} &= 1.07 \times (7.9R_i - 23.3R_i^2 + 28.7R_i^3 - 13.3R_i^4) \\
 R_{(j,0)} &= rand(), \quad j = 1, 2, 3, \\
 U_0 &= rand()
 \end{aligned}
 \tag{12}$$

where, k is the number of iteration.

Figure 4 shows the flowchart diagram of the presented CGOA.

Fig. 4 The diagram flowchart of the CGOA



4.3 Validation of the CGOA

For performance analysis of the presented method, 4 standard test functions have been studied. Experimental simulations of the presented CGOA are compared with some different popular and newly introduced algorithms including genetic algorithm (GA) [10], shark smell optimization (SSO) algorithm [19], particle swarm optimization algorithm (PSO) [20], world cup optimization algorithm (WCO) [11], and the original grasshopper optimization algorithm (GOA) [16].

The simulations are established based on MATLAB R2017b platform on a laptop computer with processor Intel® Core™ i7-4720 HQ CPU@2.60 GHz with 16 GB RAM. Table 1 demonstrates the results of the simulation.

The results of the mean deviation (MD) and the standard deviation (SD) values of the methods for the analyzed benchmarks are illustrated in Table 2.

Table 1 The utilized benchmarks for efficiency analysis

| Benchmark | Formula | Constraints | Dimension |
|------------|--|-----------------|-----------|
| Rastrigin | $f_1(x) = 10D + \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i))$ | [-512, 512] | 30-50 |
| Rosenbrock | $f_2(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1}) + (x_i - 1)^2)$ | [-2.045, 2.045] | 30-50 |
| Ackley | $f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$ | [-10, 10] | 30-50 |
| Sphere | $f_4(x) = \sum_{i=1}^D x_i^2$ | [-512, 512] | 30-50 |

Table 2 The results of the efficiency analysis by considering 30-dimensions

| Benchmark | | CGOA | GA [10] | PSO [20] | WCO [11] | GOA |
|-----------|----|------|---------|----------|----------|----------|
| f_1 | MD | 0.00 | 70.61 | 74.24 | 2.19 | 3.53 |
| | SD | 0.00 | 1.66 | 8.96 | 4.35 | 3.47 |
| f_2 | MD | 9.53 | 35.41 | 200.1 | 13.16 | 9.71 |
| | SD | 2.87 | 27.15 | 59.00 | 4.62 | 3.86 |
| f_3 | MD | 0.00 | 3.19e-2 | 8.26 | 3.14e-3 | 5.19e-17 |
| | SD | 0.00 | 2.14e-2 | 1.19 | 1.12e-3 | 0.00 |
| f_4 | MD | 0.00 | 1.15e-4 | 8.27e-4 | 6.19e-9 | 0.00 |
| | SD | 0.00 | 3.14e-5 | 5.12e-4 | 3.28e-9 | 0.00 |

The results from Table 2 show that in all four benchmarks, the proposed CFSSO algorithm gives satisfying results toward the other methods, especially the original FSO algorithm.

5 Artificial Neural Network

The very high speed and processing power of the human brain go back to the very large interconnections that exist between the brain-forming cells, and basically, without these communication links, the human brain would be reduced to a normal system and certainly not have the current capabilities. After all, the brain's excellent performance in solving a variety of problems and its high performance has made brain simulation and its capabilities the most important goal of hardware and software architects. In fact, if there is a day (but apparently not too far away) that we can build a computer with the same capabilities of the human brain, there will surely be a major revolution in science, industry, and of course, human life. Since the decades that computers have enabled computational algorithms to simulate human computational behavior, much research has been undertaken by computer scientists, engineers, and mathematicians, whose results are in the field of artificial intelligence and it is classified under the subcategory of computational intelligence as the topic of "Artificial Neural Networks". In the field of artificial neural networks, numerous mathematical and software models have been proposed to inspire the human brain which is applied to solve a wide range of scientific, engineering and practical problems in various fields.

Several types of computational models have been introduced as generic artificial neural networks, each of which can be used for a variety of applications, each inspired by a particular aspect of the capabilities and properties of the human brain. In all of these models, a mathematical structure is assumed to be graphically presentable and has a set of adjustable parameters. This general structure is adjusted and optimized by a training algorithm so that it can exhibit good behavior. A look at the learning process in the human brain also shows that we actually experience a similar process in the brain and all of our skills, knowledge, and memories are shaped by the weakening or strengthening of the communication between neurons. This reinforcement and weakening in mathematical language model and describe itself as a parameter (called Weight). One of the most basic neural models available is the Multi-Layer Perceptron (MLP) model that simulates the transient function of the human brain. In this type of neural network, most of the network behavior of the human brain and its signal propagation has been taken into account, and hence are sometimes referred to as Feedforward Networks. Each of the neurons in the human brain, called neurons, processes it after receiving input (from one neuronal or non-neuronal cell) and transmits the result to another cell (neuronal or non-neuronal). This behavior continues until a definite result is reached, which may eventually lead to a decision, process, thought, or move. The most popular method in the feed-forward networks is backpropagation (BP). This method evaluates the error on all of the training pairs and adjusts the weights to fit the desired output. To do so, several epochs have been

considered and the process continues until achieving the minimum value for the total error on the training set or until the termination criteria are reached.

This method adopts supervised learning to train the network using the data for which inputs, as well as target outputs, are noticed. After training, the weights of the networks are fixed and can be employed for evaluating the output values of the new given samples.

BP is a method based on gradient descent algorithm on the error space that sometimes gets trapped into the local minimum, succeeded it entirely dependent on initial (weight) settings. This drawback can be covered by an exploration searching capability of the evolutionary algorithms.

6 ANN Weights Evolution Using CGOA

Due to modeling by artificial neural networks, they can obtain the desired precision to determine the optimal network architecture by receiving feedback from the network and the test process and time consuming, but this increases the hidden layers and the complexity of the neural network architecture. CGOA can be used as a method to find optimal values of different neural network parameters. This algorithm starts the search with a primitive set of random solutions called the initial population and can perform multilaterally and user searches on a population of variables that increase the likelihood of finding the global optimal point. This algorithm can then simultaneously determine the optimal network structure and weights using a function that is a measure of the optimal performance quality of the response produced. The method of optimal parameter selection of the weights based on the proposed grasshopper optimization algorithm can be formulated as follows:

1. Start algorithm
2. Generate N number of initial random grasshoppers as the weight of the network
3. Calculate the cost of the network based on each grasshopper
4. Apply the CGOA operators to each of the grasshoppers to generate the next generation
5. If the termination condition is reached, go (5), else go (2)
6. End of algorithm.

Consider a simple multi-layer perceptron network with its weights (w) and biases (b):

$$\sum_{i=1}^H w_i f \left(\sum_{j=1}^d w_j x_j + b \right) \quad (13)$$

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

The network can be optimized by optimal selection of the weights and biases based on minimizing the mean squared error of the network (MSE) as follows:

$$MSE = \frac{1}{2} \sum_{k=1}^g \sum_{j=1}^m (Y_j(k) - T_j(k))^2 \quad (14)$$

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

7 Dataset Description

We used the Bao dataset to show our proposed method; this database includes 370 face images from various races, with 221 images with multiple people and other 149 images of one person mostly from Asia, with a wide range of size, lighting and background. In this paper, Bao Face Database has been utilized. This database is introduced by Frischholz that includes 370 face images from different races, mostly from Asia, with a wide range of size, lighting and background [21]. The database has no using or distribution policies or license.

8 Simulation Results

This study considers two categories including skin and non-skin classes. The method is based on pixels information that uses this information for classifying the skin-like and non-skin-like pixels one by one. This method reduces the image information into objects of interest, like faces or hands during skin pixels detection.

The input of the proposed classifier is a matrix of $3 \times n$ pixel coefficients from the images either skin or non-skin image which n determines the number of neurons in the hidden layer. Since the employed transfer function is sigmoid, the output image gives different values in the range 0 and 255 (uint8 mode).

Since we need two labels in the final output (skin or non-skin), the output of the neural classifier should be thresholded so that it is either 0 or 1. In this study, Otsu thresholding has been used for this purpose, such that the output values of the neural network have been classified into two categories of 0 and 1 [22].

After thresholding the output, mathematical morphology has been applied to remove the extra parts. In this study, three operators have been utilized including

opening, closing and filing holes [23]. Figure 5 shows the impact of the mathematical morphology on the images.

For validating the efficiency in the proposed system, three performance indices have been employed. The first metric is the false acceptance rate (**FAR**) that defines the ratio of the identification moments in which false acceptance happens. The second index is the false rejection rate (**FRR**) which is the percentage of the identification moments in which false rejection happens and the final index is the correct detection rate (**CDR**) which shows the correctly classified pixels to the total pixels.

$$CDR = \frac{\text{No. of Pixels Correctly Classified}}{\text{Total Pixels in the Test Dataset}} \tag{15}$$

$$FAR = \frac{\text{No. of non - skin Pixels Classified as skin Pixels Classified}}{\text{Total Pixels in the Test Dataset}} \tag{16}$$

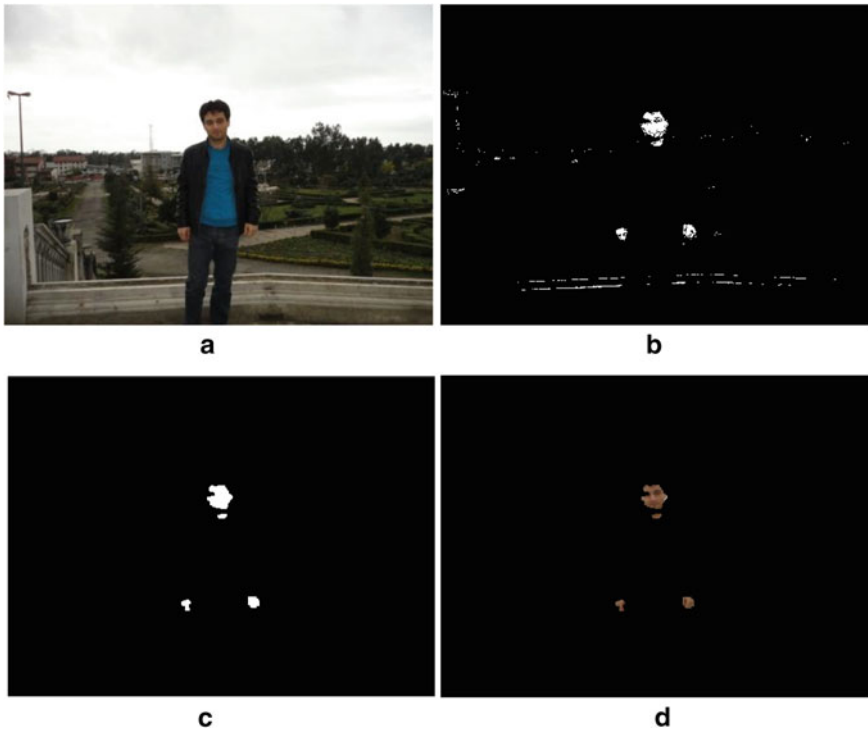


Fig. 5 **a** Original image, **b** skin region segmentation and thresholding, **c** applying mathematical morphology, **d** skin area detection

Fig. 6 Test set: **A** skin sets, **B** non-skin sets



Table 3 The performance of the skin region detection for the methods

| Algorithm | CDR | FRR | FAR |
|-----------|-------|------|-------|
| CGOA-ANN | 82.15 | 7.45 | 10.4 |
| HNN-ICA | 70.84 | 4.16 | 25 |
| MLP | 68.42 | 5.2 | 26.38 |

$$FRR = \frac{\text{No. of skin Pixels Classified as non - skin Pixels Classified}}{\text{Total Pixels in the Test Dataset}} \quad (17)$$

Figure 6 shows the utilized test set from the skin and the non-skin to measure the metric efficiency.

Here, the least mean square error minimization has been utilized for improving the network weights between the desired and the real outputs of the network. Comparison results of the proposed method with basic MLP and HNN-ICA [6] are tabulated in Table 3. As can be observed, the proposed optimized network performs better and gives a higher correct detection rate.

Figures 7 and 8 show the extent of the match between the measured and predicted skin rate of the train set and validation phases by CGOA-ANN in terms of a scatter diagram.

For more clarification, some sample outputs have been shown in Fig. 9. As can be observed, using the proposed CGOA-ANN classifier gives satisfying results for skin regions detection including face, hands, and neck.

9 Conclusion

This paper proposed a new optimized method for skin area classification. A new improved version of the grasshopper optimization algorithm was adopted to decrease the mean square error of the neural network to escape from the local minimum. Here, RGB color space was utilized for decreasing the computational cost. The proposed algorithm reduces the number of false detection ratio toward a gradient descent algorithm. The skin region detection method based on CGOA-ANN is compared with some methods to validate and analyze the efficiency.

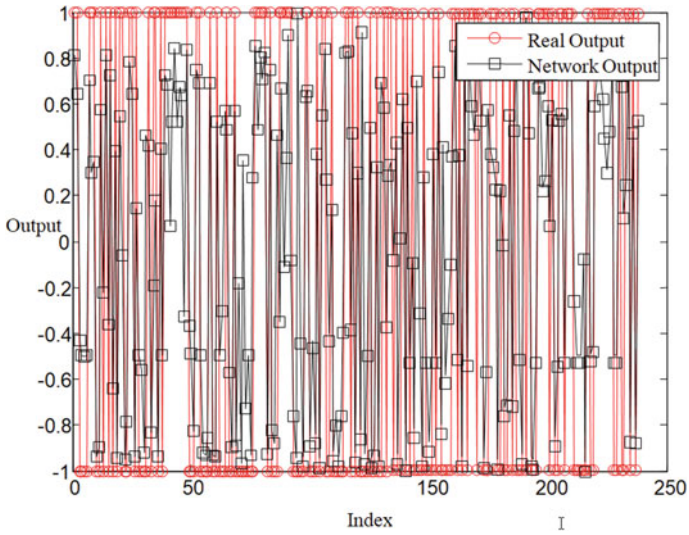


Fig. 7 Real output and CGOA-ANN network output for the training data

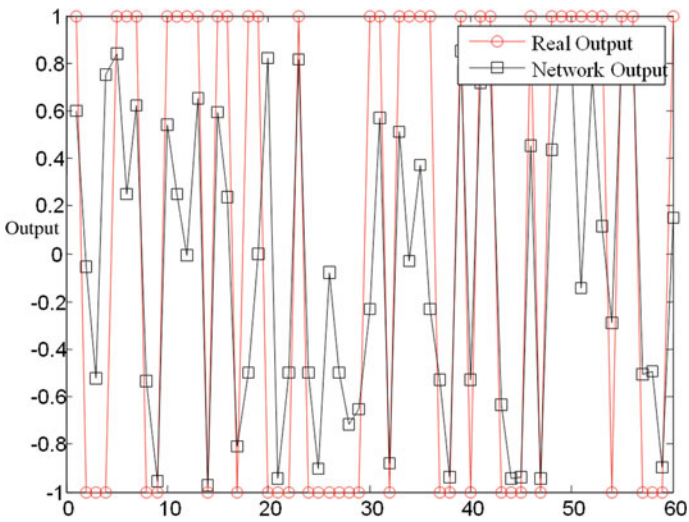


Fig. 8 Real output and CGOA-ANN network output for the test data

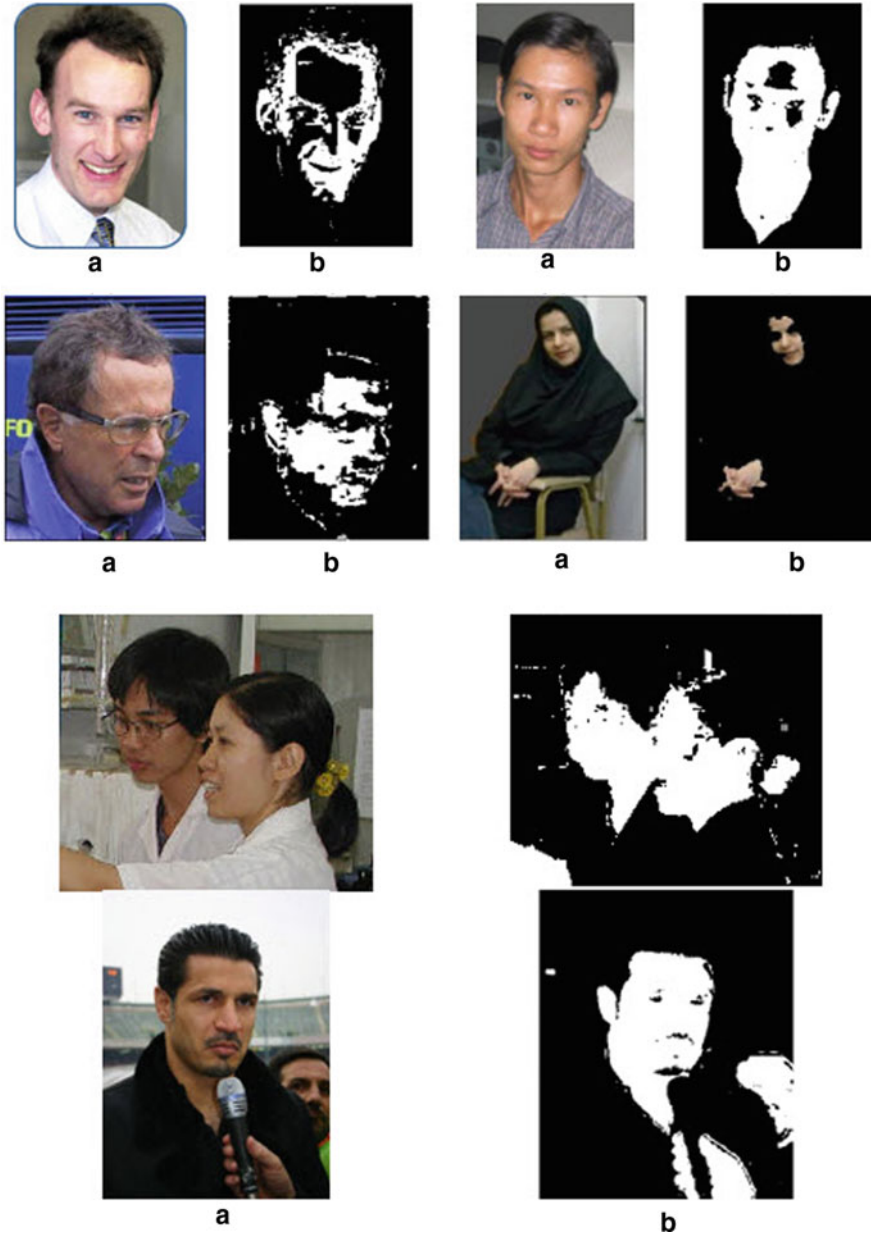


Fig. 9 Experimental results for skin region detection: **a** original image, **b** skin segmented result

References

1. Rashid Sheykhahmad F, Razmjoo N, Ramezani M (2015) A novel method for skin lesion segmentation. *Int J Inf Secur Syst Manag* 4:458–466
2. Sun X, Wu P, Hoi SC (2018) Face detection using deep learning: an improved faster RCNN approach. *Neurocomputing* 299:42–50
3. Jalal A, Kamal S, Azurdia-Meza CA (2019) Depth maps-based human segmentation and action recognition using full-body plus body color cues via recognizer engine. *J Electr Eng Technol* 14:455–461
4. Tian C, Zhang X, Wei W, Gao X (2018) Color pornographic image detection based on color-saliency preserved mixture deformable part model. *Multimed Tools Appl* 77:6629–6645
5. Aygun R, Benesova W (2018) Multimedia retrieval that works. In: 2018 IEEE conference on multimedia information processing and retrieval (MIPR), pp 63–68
6. Razmjoo N, Mousavi BS, Soleymani F (2013) A hybrid neural network Imperialist Competitive Algorithm for skin color segmentation. *Math Comput Model* 57:848–856
7. Razmjoo N, Ramezani M (2016) Training wavelet neural networks using hybrid particle swarm optimization and gravitational search algorithm for system identification. *Int J Mechatron Electr Comput Technol* 6:2987–2997
8. Razmjoo N, Sheykhahmad FR, Ghadimi N (2018) A hybrid neural network–world cup optimization algorithm for melanoma detection. *Open Med* 13:9–16
9. Zhang Z (2018) Artificial neural network. In: *Multivariate time series analysis in climate and environmental research*. Springer, pp 1–35
10. Holland JH (1992) Genetic algorithms. *Sci Am* 267:66–73
11. Razmjoo N, Khalilpour M, Ramezani M (2016) A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: theory and its application in PID designing for AVR system. *J Control Autom Electr Syst* 27:419–440
12. Namadchian A, Ramezani M, Razmjoo N (2016) A new meta-heuristic algorithm for optimization based on variance reduction of Gaussian distribution. *Majlesi J Electr Eng* 10:49
13. Clapham PJ (2000) The humpback whale. In: *Cetacean societies, field studies of dolphins and whales*. The University of Chicago, Chicago, pp 173–196
14. Dhiman G, Kumar V (2018) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. *Knowl-Based Syst* 159:20–50
15. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
16. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems. *Appl Intell* 48:805–820
17. Yang D, Li G, Cheng G (2007) On the efficiency of chaos optimization algorithms for global optimization. *Chaos Solitons Fract* 34:1366–1375
18. Rim C, Piao S, Li G, Pak U (2018) A niching chaos optimization algorithm for multimodal optimization. *Soft Comput* 22:621–633
19. Abedinia O, Amjadi N, Ghadimi N (2018) Solar energy forecasting based on hybrid neural network and improved metaheuristic algorithm. *Comput Intell* 34:241–260
20. Bansal JC (2019) Particle swarm optimization. In: *Evolutionary and swarm intelligence algorithms*. Springer, pp 11–23
21. Frischholz R (2012) Bao face database at the face detection homepage
22. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9:62–66
23. Razmjoo N, Mousavi BS, Soleymani F (2012) A real-time mathematical computer method for potato inspection using machine vision. *Comput Math Appl* 63:268–279