

Face Detection in Images Using the Cuckoo Optimization Algorithm

Corresponding Author:

Reza Dehdashti

Department of Computer Engineering, Faculty of Engineering Boroujerd.Branch, Islamic
Azad University, Boroujerd, Iran

Article Received: 27-July-2025, Revised: 17-August-2025, Accepted: 07-September-2025, Online First: 08-September-2025

ABSTRACT:

In recent years, facial recognition has become a prominent area of interest in identity verification systems. Since facial recognition methods rely on computational algorithms designed to identify individuals based on a database of facial images, they have been widely studied. However, less attention has been given to detecting the number of faces within an image and accurately distinguishing facial regions from the background or other objects. Low accuracy and slow processing speed remain two major challenges in the hardware implementation of facial recognition systems. This study investigates face detection in images using the Cuckoo Optimization Algorithm, a metaheuristic approach. By combining image processing techniques with this algorithm, the research aims to determine the number of faces in images and to accurately isolate facial regions. The proposed method was implemented in MATLAB 2015b using the necessary custom commands. The dataset used in this study consisted of 20 images containing varying numbers of faces, collected by the researcher. The results indicate that the Cuckoo Optimization Algorithm can accurately detect the number of faces in an image. Moreover, the algorithm demonstrated high precision in separating facial regions from other areas. It was also observed that increasing the initial population size led to improved optimization performance.

Keywords: Face detection, Metaheuristic algorithm, Cuckoo Optimization.

INTRODUCTION:

The human face plays a fundamental role in personal identification and the expression of emotions. It can convey aspects of personality and emotional state. Humans possess a remarkable ability to recognize and distinguish faces, though this capability varies among individuals. The human capacity to perceive facial attributes such as expressions, age, and emotions—is particularly fascinating. Although numerous studies have explored the mechanisms through which the brain performs facial recognition, the exact process remains unclear. To replicate this human ability, researchers have conducted extensive studies in the fields of artificial intelligence and machine vision to enable computers to recognize faces in images (Young, 2014).

One of the key challenges in this domain is identifying all the faces present in an image. Using various mechanisms, it is possible to determine how many faces are in a photo and utilize this information for subsequent image-processing steps. In essence, accurately detecting faces in an image allows facial recognition algorithms to focus on relevant regions, thereby improving the identification of individuals. For example, airport security cameras often follow a two-step process: first, detecting all faces in a frame, and second, analyzing facial features to identify individuals. As Young (2010) notes, detecting the

number of faces in an image is important because it enables the extraction of the relevant areas for further processing in face recognition systems. More specifically, accurately selecting facial regions and clearly defining their edges and boundaries concerning other objects enhances the precision of facial recognition algorithms. Given the significance of detecting faces and distinguishing them from surrounding elements in an image, this task is a crucial step in identifying individuals. Therefore, developing precise algorithms to determine the number of faces in an image and accurately segment facial regions from the background is the focus of this research.

A major challenge in this process is achieving high accuracy in both face counting and facial region extraction. Various studies have aimed to accurately detect the number of faces in an image and to distinguish the boundaries of facial areas from surrounding objects. The task of face detection can be viewed as an optimization problem, where the goal is to identify the correct number of faces with minimal error and to accurately extract their regions relative to surrounding features (Kim, 2014). Optimization problems like this are common across many fields of industry and technology. In such cases, it is not enough to find just any solution; rather, the goal is to identify solutions that minimize implementation costs or maximize performance. Evolutionary algorithms, inspired by natural processes and the principles of

evolution and natural selection, aim to solve complex optimization problems efficiently and without prior assumptions. The Cuckoo Search Algorithm is one such evolutionary method, recognized for its high accuracy in solving optimization problems (Gandomi, 2013).

In a related study, Yildiz and colleagues (2007) employed various distance metrics to measure the similarity between an input image and stored images. They compared face recognition performance using edge distance, facial region distance, angular distance, and a combined metric of all four. The results showed that combining all four distance measures yielded the most accurate recognition outcomes.

Tseng (2011) proposed a method called Modular PCA, which demonstrates improved recognition accuracy compared to traditional PCA, particularly under varying lighting conditions and facial expressions. In this approach, facial images are divided into smaller segments, and PCA is applied independently to each segment. This segmentation reduces the impact of global variations (like lighting or expression changes) by preserving the local features of the face. Turkpeti (2014) introduced a hybrid facial recognition system that combines PCA with other algorithms. This system operates in two phases: training and classification. During the training phase, facial images are normalized, and features are extracted using PCA and ICA. These extracted features are then used to train a Back Propagation Neural Network (BPNN) to classify them into distinct groups. In the classification phase, new images are added to the training results, and a combination algorithm is applied to categorize them based on the trained classes.

Another study introduced a Multi-layer PCA method. In this approach, a linear neural network is used to compute the principal components, which significantly reduces computational requirements and simplifies hardware design. Additionally, due to the parallel processing capabilities of neural networks, computation speed is significantly improved. The term *multi-layer* here refers to a two-step recognition process: initially, a subset of faces with the highest similarity to the target face is selected. Then, the proposed recognition process is continued within this narrowed set, increasing overall accuracy (Altas, 2010).

Niu et al. (2013) proposed a novel face recognition method based on skin color. They used a set of training images to extract histograms and developed a fuzzy model based on the RGB channels' light intensity. This model was used to identify individuals by their skin tone, achieving 94% accuracy in correctly identifying the facial regions. Nashat et al. (2013) introduced a smartphone-based facial recognition system. Using video frames and a training dataset, they applied an artificial neural network to detect faces in real-time. Their method also utilized cloud-based distributed computing to recognize facial patterns. Tests in both bright and dark environments showed

promising results, with recognition accuracy exceeding 90% in some cases. Shin et al. (2017) proposed a face recognition technique based on Local Binary Patterns (LBP). Their method used neighborhood pixel information and multi-mapping strategies to extract binary representations of facial regions. The resulting binary patterns were then used to identify the facial image.

Zhou et al. (2015) introduced a genetic algorithm-based method for facial recognition. In their approach, essential image features were encoded into chromosomes, and a genetic algorithm consisting of crossover and mutation was applied to maximize a fitness function that measures the similarity between a face and the training data. Their results showed up to 98% accuracy on certain datasets. Emran (2013) emphasized the importance of hybrid systems in facial recognition, one of the most significant biometric technologies. His proposed system used a parallel hybrid architecture combining the strengths of PCA, LDA, and Gabor filters for feature extraction. A Feedforward Backpropagation Neural Network was used to integrate these features. The results demonstrated that this hybrid approach improved resistance to external factors such as lighting and facial expressions, leading to higher recognition accuracy. Finally, Pujol et al. (2017) discussed the foundations of fuzzy logic, contrasting it with classical binary logic. They highlighted that real-world phenomena often exist in a continuum rather than in binary true/false categories. By applying fuzzy logic along with fuzzification and defuzzification processes and identifying 21 key facial points, their system could accurately detect emotional states such as happiness, sadness, surprise, fear, and anger.

In the present study, PCA is used to reduce the dimensionality of the feature space, and the Cuckoo Search algorithm is employed during the testing phase to detect faces by identifying the optimal facial region. Since there is no pre-existing dataset available to implement the research objectives, a custom database was compiled. This dataset consists of 30 different images containing varying numbers of faces. To meet the research requirements, suitable images were selected based on specific criteria, and all input images were standardized in terms of size and resolution. Subsequently, using the Cuckoo Search algorithm implemented in MATLAB 2015, the number of faces in each image was detected. All implementation steps were carried out using MATLAB software.

Proposed Method Phases:

The proposed method begins with the collection of the required dataset. After that, by defining the necessary commands in MATLAB, the implementation of the research objectives is carried out.

Phase 1: Data Collection and Preprocessing:

Since there was no pre-existing dataset available to implement the research objectives, a custom database

was created for this study. This database consists of 30 different images, each containing a varying number of faces. After reviewing multiple images, only those that met the necessary conditions for implementing the research objectives were selected. The selected images were then standardized in terms of dimensions and file size. Once the dataset was compiled, the required hardware and software environment was set up, and the implementation began using MATLAB 2015b. In the first phase, the prepared images were loaded into MATLAB for further processing.

Phase 2: Implementation in MATLAB:

In the second phase, conducted within the MATLAB environment, the necessary scripts and functions were written and executed to achieve the research objectives. Specifically, a face detection object was defined using MATLAB's Computer Vision Toolbox via the command `vision.CascadeObjectDetector`. A loop was then implemented to read images 1 to 30 from the dataset.

Face detection systems are a type of biometric system and are among the most important technologies for identifying individuals. These systems are widely used in **access control** and, following the success of fingerprint recognition systems, have become one of the most significant biometric identification methods in recent years.

In this stage of the study, using detection and information extraction functions, the images are first evaluated for content, and irrelevant data—such as background, hair, neck, and shoulders—are removed to isolate only the facial region. The resulting image is then subjected to computational and operational processes to extract surface-level facial features and analyze the overall image data. Essentially, at this point, the output image, which will be used by the program for facial recognition, is prepared using PCA methods.

Since one of the main objectives of the present study is facial region detection, the rationale for using the PCA method can be explained as follows:

Given that dimensionality reduction techniques are generally divided into two categories:

The first method is based on feature extraction, which maps a high-dimensional space to a lower-dimensional one. In essence, by combining existing feature values, a smaller set of new features is generated, which still retains all or most of the information contained in the original features. Feature extraction methods are generally divided into two categories: linear and non-linear. Linear methods, which are simpler and easier to understand, aim to find a general flat subspace. In contrast, non-linear methods, which are more complex, seek to find a local flat subspace. The second method is based on feature selection, which attempts to reduce the dimensionality of data by selecting a subset of the original features. However, this method will not be discussed further in this text.

The problem of data dimensionality reduction can be mathematically stated as follows: Suppose we have a p -dimensional random variable $X = (x_1, x_2, x_3, x_4)^k$

We want to find a K -dimensional variable $S = (s_1, s_2, s_3, s_4)^T$ such that first, $P < K$, and second, S contains the information present in X according to a certain criterion. Linear methods attempt to obtain each of these K components as a linear combination of the original P components.

$$S = WX \rightarrow S, -$$

$$EW, X, (1, 2, \dots, K)$$

W_{kp} represents the weight matrix used for the linear mapping. It is important to note that PCA is considered the most effective method for linear dimensionality reduction. By discarding less significant coefficients derived from this transformation, PCA minimizes information loss compared to other techniques.

PCA, which primarily relies on the use of eigenfaces (which capture the structural information of facial components), was first proposed by Kirby and Sirovich in 1998. For PCA to work effectively, all input images must be of the same size and initially normalized based on key facial landmarks such as the line between the eyes and the mouth. PCA then reduces the dimensionality of the data by leveraging data compression principles and identifies the most influential dimensions that capture the facial pattern's structure. This dimensionality reduction eliminates redundant information and accurately extracts the core structural features of the face, known as eigenfaces, simplifying the complexity of the data.

Each facial image is represented as a weighted combination of these eigenfaces, stored as a one-dimensional vector. The process of facial recognition involves comparing the vector of the input image with those in the database by measuring the distances between their facial feature vectors. Images must be captured from a frontal view for PCA to perform reliably; otherwise, the recognition accuracy significantly decreases. The primary advantage of PCA is its ability to drastically reduce the amount of data required for identification while preserving essential facial information.

In PCA, new coordinate axes are defined for the data, and the data points are represented based on these new axes. The first axis is oriented in the direction where the variance of the data is maximized—that is, the direction with the greatest data spread. The second axis is orthogonal to the first and is chosen such that the variance of the data along it is also maximized. Similarly, each subsequent axis is orthogonal to all the previous axes and is placed in a direction where the data exhibits the highest remaining variance. Figure (2) illustrates this concept for two-dimensional data.

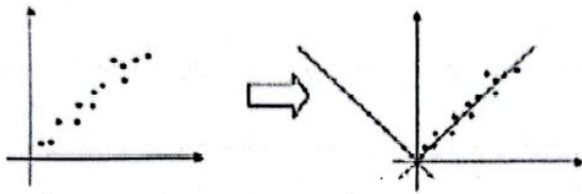


Figure 1: New axes based on eigenvectors aligned with the directions of highest data density (Nesing, 2010).

In general, a set of face images can be mapped onto a set of points in a high-dimensional space. Face images at a high level share many similarities and their data is not randomly distributed. Therefore, they can be represented as correlated vectors distributed within a smaller subspace.

In reality, a two-dimensional image is stored in an $N \times N$ matrix, which means that N^2 memory locations are needed to store and represent one image. Using PCA, this storage requirement is reduced to M , which corresponds to a one-dimensional space. The main idea behind PCA is to find the best vectors that represent the distribution of face images among the input faces. These vectors define a subspace of the images, known as the face image subspace. Each of these vectors corresponds to an $N \times N$ vector that is a linear combination of the original images. Since these vectors are the eigenvectors of the covariance matrix of the original face images and still resemble faces in their representation, they are known as “eigenfaces.” Overall, face recognition using PCA includes the following steps:

First, the initial image database is collected and created. Then, the process of calculating the number of faces present in the input images is performed using programming commands. By calling the cuckoo algorithm functions—which include defining the algorithm parameters in MATLAB and gathering the optimal function values defined in the proposed algorithm—and specifying the output display method (drawing green rectangles around the detected faces), the number of faces in each input image is identified.

Phase Three of the Proposed Method:

In this implementation phase, using the functions called in the previous phase, the face region is defined by identifying the Region of Interest (ROI). A loop is defined to separate the detected face region from other areas. Finally, output commands are used to display the results.

FINDINGS:

Parameters of the Proposed Algorithm:

In the implementation of the cuckoo algorithm within the proposed solution, three key parameters are particularly important: population size, habitat (the location of the cuckoos), and another influential factor. The population size parameter determines the initial

number of individuals for running the algorithm. In the proposed method, this size is set to 20 and is managed using the cuckoo algorithm.

This parameter represents the number of iterations of the algorithm. If the value is too small, the algorithm may fail to reach an accurate solution, while a very large value will result in excessive computation time. Therefore, choosing an appropriate value for this parameter is crucial for the efficiency of the algorithm’s execution. In the proposed solution, this parameter is set to 2000.

The suitability of a habitat is determined by evaluating the profit function (fp) at that habitat. The cuckoo evolutionary algorithm aims to maximize the profit function. To apply the cuckoo algorithm to minimization problems, it is sufficient to multiply the cost function by a negative sign. At the start of the optimization process, a habitat matrix of size $N_{pop} \times N$ is generated. Then, a random number of eggs is assigned to each habitat.

Each real cuckoo naturally lays its eggs within a specific range, known as the Egg Laying Radius (ELR). In an optimization problem, every variable has an upper bound (var_high) and a lower bound (var_low), which are used to define the ELR. The ELR is proportional to the total number of eggs, the current number of eggs laid by the cuckoo, as well as the upper and lower bounds of the problem variables.

Therefore, ELR is calculated by the following relationship:

$$ELR = \alpha \times \frac{\text{number of current cuckoo's eggs}}{\text{total number of eggs}} \times (Var_{hi} - Var_{low})$$

Upper Bound (Ub): The maximum allowable value for each parameter is 20

Lower Bound (Lb): The minimum allowable value for each parameter is 5

Number of Optimization Parameters (d): In this problem, the number of parameters to optimize is 3. These parameters are: face detection, face separation, and algorithm accuracy.

Simulation Environment

To evaluate the proposed solution, we used a dataset collected by the researcher consisting of 20 images. The implementation was performed in MATLAB 2016b. The results of the implementation are as follows: To detect the number of faces in an image, first, the target images from the dataset were loaded into MATLAB. After reading the images, face detection was performed using relevant commands, followed by invoking the cuckoo algorithm to address the research objectives.

Implementation Results

For the proposed algorithm to find the best solution in the shortest possible time, the defined parameters must be carefully tuned. Therefore, to find the optimal values for these parameters, the proposed algorithm was executed multiple times with different parameter values until the most suitable values were identified.

To obtain the best values for the cuckoo algorithm parameters, multiple runs were performed with various parameter settings on the dataset of 20 different images collected by the researcher.

First, a face detection object was defined in MATLAB to determine the number of faces present in the images. Then, images were read using the appropriate commands. Next, by invoking the cuckoo algorithm, optimal values for face count detection were obtained and the optimal function values were collected. Finally, the results were displayed by drawing rectangles around the detected faces.

The output results are shown in the following images (only six output images are displayed here) (Figures 2 and 3).

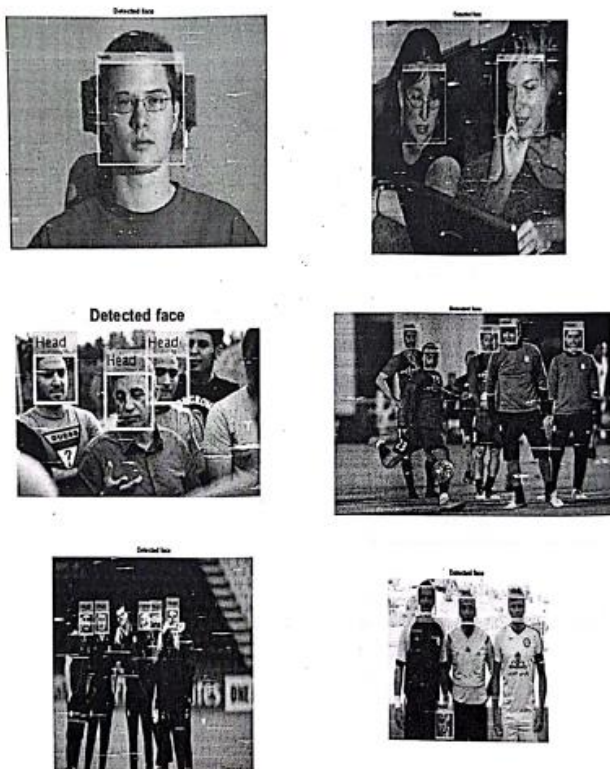


Figure 2: Implementation results of face count detection in images using the cuckoo algorithm.

To investigate the second research objective, which is to distinguish facial regions from other parts of the image, we used the detection object defined in the previous step. After loading the images with the relevant commands, we applied the Flower Pollination Algorithm and the Region of Interest (ROI) technique, which were employed in this study to separate the face from other areas of the image.

The detected face regions were distinguished from the rest of the image by pixelating the non-face areas. The output results are shown in the following images.

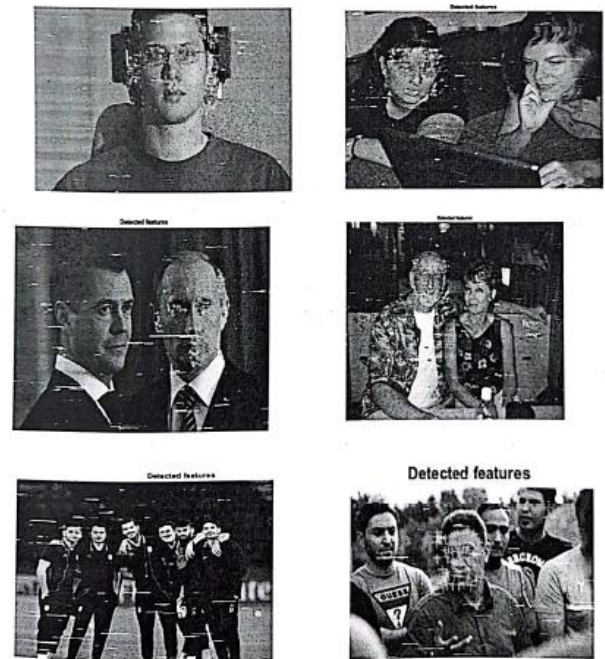


Figure 3: Results of face region detection from other parts of the image using the cuckoo algorithm.

Increasing the initial population size and the number of iterations in the cuckoo algorithm can improve the accuracy of the proposed method in both detecting the number of faces and identifying the regions where the faces are located.

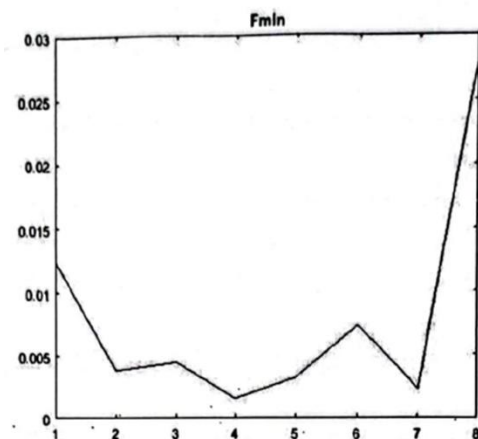


Figure 4: Evaluation of the Proposed Algorithm's Accuracy with Increasing Initial Population

As shown in Figure 4, the optimal objective function value increases as the number of images increases. The processing results in MATLAB indicate that the optimal function value for an image containing one face is 0.012. For image 3, which contains two faces, the optimal values of the proposed algorithm were 1.003, 1.004, and 0.001, respectively. For image 4, with five faces, the optimal values were 1.003 and 0.007, and for other images containing four and three faces, the values were 0.002 and 0.02, respectively. Consequently, it can be concluded that increasing the

initial population (number of images) results in a higher optimal function value.

Experimental Results and Comparisons:

The evaluation shows that the proposed algorithm achieves over 95% accuracy in both face region detection (separating faces from other image parts) and face count detection. This performance surpasses the method by Kashef et al. (2015), who introduced a novel approach for face recognition using skin color. Their method employed a training dataset and extracted histograms of each image. Based on the intensity frequencies of the three RGB channels, a fuzzy model was proposed to identify individuals by their skin color. Their results showed that in 94% of cases, the face region was correctly detected.

Additionally, the proposed algorithm outperforms the study by Emran et al. (2013), which presented a smartphone-based face detection system. Their method used video frames captured from the environment and a training dataset to train an artificial neural network to recognize faces. Their approach also involved cloud-based distributed systems for image pattern recognition. Their tests in various lighting conditions showed satisfactory results, with some experiments achieving accuracy above 90%.

Compared to Choudhary (2017), who investigated particle swarm optimization (PSO) and imperialist competitive algorithm (ICA) for feature selection in face detection, the proposed algorithm yielded better results. Choudhary used binary PSO to reduce non-optimal points and computational time, ultimately showing that the ICA selected suitable features for face detection. Their method achieved approximately 90% accuracy.

When compared to Mahmoud et al. (2014), who combined PCA and Gabor methods for face detection on the Yale and ORL datasets, reporting accuracies of 98.33% and 98.46% respectively, the proposed algorithm showed lower accuracy. Similarly, Zhou et al. (2015) used the PEA method and hierarchical functions on the Yale and other datasets, reporting 97.38% and 96.26% accuracy, which are higher than those obtained here.

Shin et al. (2016) proposed a novel face detection method using local binary patterns by analyzing neighborhood pixel information and multiple mappings of images. Their approach generated binary images based on pixel thresholding and matrix mapping, achieving approximately 92% accuracy, which is less than the proposed algorithm's performance.

Podul et al. (2017) developed a face detection method based on a genetic evolutionary algorithm, encoding key image features into chromosomes. The genetic algorithm applied mutation and crossover phases to maximize a fitness function measuring the match between the face and training data. Their results indicated an accuracy of around 98% in some datasets,

outperforming the flower pollination algorithm used in the current study.

Sushija (2016) introduced a hybrid parallel system combining the adaptive strengths of three well-known feature extraction algorithms—LDA, PCA, and Gabor—using a feedforward backpropagation neural network to fuse their outputs. Their system improved robustness against external factors such as lighting and pose variation, achieving an accuracy of 98%, which is higher than that of the current algorithm.

CONCLUSION:

Face recognition remains one of the most challenging problems in the field of image processing. Since the 1980s, it has been a widely researched and sometimes controversial topic, with various solutions proposed for different scientific challenges. Face recognition is a biometric technique primarily used to identify individuals based on their facial features. However, the process of face recognition as performed by the human brain is highly complex and difficult to replicate. This paper proposes a method combining singular value decomposition analysis with the cuckoo algorithm to detect the number of faces and to separate faces from other parts of an image. The results show that this algorithm reduces computation time and enhances processing speed. Moreover, increasing the initial population size leads to improved algorithm accuracy. The detection rate of the proposed algorithm was also observed to be superior.

The limitations of this study include a lack of available resources and academic articles on applying the cuckoo algorithm to image processing, the inability to implement the proposed algorithm in real-world software environments to further evaluate its accuracy, the complexity of face detection algorithms and methods, and the absence of standardized datasets for implementation.

In this research, a novel approach based on the cuckoo algorithm was introduced for identifying faces within images. The proposed method employs image processing techniques alongside the algorithm innovatively to minimize recognition errors and achieve more accurate pattern extraction. The outcomes of this study have potential applications in identity verification systems, photography, and videography.

The proposed solution utilizes the cuckoo algorithm, which is a metaheuristic optimization technique. Other metaheuristic algorithms, such as ant colony optimization or particle swarm optimization, could also be employed to enhance experimental results, and implementations can be explored with these alternatives. Additionally, combining metaheuristic algorithms for example, integrating genetic algorithms with ant colony optimization or other metaheuristics may further improve performance.

REFERENCES:

1. Chaudhry, S., & Chandra, R. (2017). Face detection and recognition in an unconstrained environment for mobile visual assistive system. *Applied Soft Computing*, 53, 168-180.
2. Gandomi, A. H., Yun, G. J., Yang, X. S., & Talatahari, S. (2013). Chaos-enhanced and Numerical Simulation, 18(2), 327-340. accelerated particle swarm optimization. *Communications in Nonlinear Science*
3. Imran, M., Hashim, R., & Khalid, N. E. A. (2013). An overview of particle swarm optimization variants. *Procedia Engineering*, 53, 491-496.
4. Imran, M., Hashim, R., & Khalid, N. E. A. (2013). An overview of particle swarm optimization variants. *Procedia Engineering*, 53, 491-496.
5. Kashef, S., & Nezamabadi-pour, H. (2015). An advanced ACO algorithm for feature subset selection. *Neurocomputing*, 147, 271-279.
6. Kim, J., Yu, S., & Lee, S. (2014). Random-Profiles-Based 3D Face Recognition System. *Sensors*, 14(4), 6279-6301.
7. Mahmud, M. E. Haque, S. T. Zuhori, and B. Pal, 2014. "Human face recognition using PCA based Genetic Algorithm," in *Electrical Engineering and Information & Communication Technology (ICEEICT)*, 2014 International Conference on, 2014
8. Neshat, M., Sepidnam, G., Sargolzaei, M., & Toosi, A. N. (2014). Artificial fish survey of the state-of-the-art, hybridization, combinatorial and indicative applications. *Artificial intelligence review*, 1-33.
9. Niu, B., Wang, H., Wang, J., & Tan, L. (2013). Multi-objective bacterial foraging optimization. *Neurocomputing*, 116, 336-345.
10. Pujol, F. A., Pujol, M., Jimeno-Morenilla, A., & Pujol, M. J. (2017). Face Detection Based on Skin Color Segmentation Using Fuzzy Entropy. *Entropy*, 19(1), 26.
11. Pujol, F. A., Pujol, M., Jimeno-Morenilla, A., & Pujol, M. J. (2017). Face Detection Based on Skin Color Segmentation Using Fuzzy Entropy. *Entropy*, 19(1), 26.
12. Shen, J., Zuo, X., Li, J., Yang, W., & Ling, H. (2017). A novel pixel neighborhood differential statistic feature for pedestrian and face detection. *Pattern Recognition*, 63, 127-138.
13. Sukhija, P., Behal, S., & Singh, P. (2016). Face Recognition System Using Genetic Algorithm. *Procedia Computer Science*, 85, 410-417.
14. Trupti M. Kodinariya: 2014. Hybrid Approach to Face Recognition System using Principle component and Independent component with score based fusion process. *CORR abs/1401.0395*
15. Tseng, P. (2010). Approximation accuracy, gradient methods, and error bound for structured Programming, 125(2), 263-295.
16. Yang, X. S. (2010). Firefly algorithm, Levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209-218). Springer London.
17. Yang, X. S., Karamanoglu, M., & He, X. (2014). Flower pollination algorithm: a novel approach for multiobjective optimization. *Engineering Optimization*, 46(9), 1222-1237.
18. Yildiz, A. R. (2013). A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing. *Applied Soft Computing*, 13(5), 2906-2912.
19. Zhou, Y., Luo, Q., Chen, H., He, A., & Wu, J. (2015). A discrete invasive weed optimization algorithm for solving traveling salesman problem. *Neurocomputing*, 151, 1227-1236.