Ingénierie des Systèmes d'Information
Vol. 31, No. 5, May, 2026, pp. 1725-1735
Journal homepage: http://iieta.org/journals/isi

Pareto-Guided Block-Level Feature Selection for Face Recognition Using Multi-Objective Particle Swarm Optimization

Mohamed Chatra[1,2], Nour Elhouda Chalabi[1,2], Abderrahim Zemmit[3], Abdelouahab Attia[4,5], Mohamed Benghanem[6*]

[1] Computer Science Department, University of M'Sila, M'Sila 28200, Algeria
[2] Laboratory of Informatics and Its Application of M'Sila, M'Sila 28000, Algeria
[3] Electrical Engineering Department, University of M'Sila, M'Sila 28000, Algeria
[4] Computer Science Department, Mohamed El Bachir El Ibrahimi, University of Bordj Bou Arreridj, Bordj Bou Arreridj 34000, Algeria
[5] Laboratory Materials and Electronic System of Bordj Bou Arreridj, Bordj Bou Arreridj 34000, Algeria
[6] Department of Physics, Faculty of Science, Islamic University of Madinah, Madinah 42351, Saudi Arabia

Corresponding Author Email: mbenghanem@iu.edu.sa

https://doi.org/10.18280/isi.310528

Received: 12 February 2026
Revised: 1 April 2026
Accepted: 15 May 2026
Available online: 31 May 2026

Keywords:
face recognition, optimization, Multi-Objective Particle Swarm Optimization, feature selection, Binarized Statistical Image Features

ABSTRACT

High-dimensional facial representations often contain redundant local information, increasing computational cost and weakening the generalization of face recognition systems. This study proposes a Pareto-guided block-level feature selection framework based on Multi-Objective Particle Swarm Optimization (MOPSO). Each face image is partitioned into non-overlapping local blocks, from which Binarized Statistical Image Features (BSIF) are extracted and further projected using Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Instead of selecting individual feature components, each particle represents a candidate subset of facial blocks. The optimization process simultaneously minimizes the Equal Error Rate (EER) and maximizes Rank-1 identification accuracy, producing a Pareto archive of alternative block combinations with different verification–identification trade-offs. A nearest-neighbor classifier is then used to evaluate the selected representations. Experiments were conducted on four benchmark datasets, namely AT&T, Georgia Tech (GT), AR, and Labeled Faces in the Wild (LFW). The proposed method achieved strong performance on controlled datasets, including 95.8 ± 0.8% Rank-1 accuracy with 1.98 ± 0.22% EER on AT&T, while using only a compact subset of BSIF block features. The results also show that increasing the number of selected blocks can improve performance on more challenging unconstrained data, although performance remains limited under severe pose, illumination, and image-quality variations. Overall, the findings indicate that Pareto-based block selection offers a compact and interpretable alternative to full-image feature representations, while preserving flexibility between verification accuracy and identification performance.

1. INTRODUCTION

The rapid growth of biometric technologies has enabled their widespread adoption in practical applications. Concurrently, the employed methodologies and the spectrum of biometric traits have undergone substantial evolution, propelled by ongoing improvements in acquisition devices. Consequently, contemporary biometric systems frequently produce high-dimensional feature representations, thereby augmenting the intricacy of data processing and management.

Face recognition, a well-established biometric method, has been widely used, along with iris and fingerprint recognition [1]. It has been extensively used in various areas, including surveillance, crime prevention, forensic analysis, access control, and social media [2]. Despite its growing use, face recognition systems still face several challenges, such as changes in head position, facial expressions, lighting, aging, and other external factors [3, 4].

A typical face recognition system follows a structured pipeline, where acquired images undergo a feature extraction stage to generate representative descriptors. This stage plays a central role in distinguishing between individuals by encoding discriminative facial characteristics.

Many feature descriptors have been introduced in the literature to improve data representation. However, these methods often create high-dimensional feature vectors with redundant or less useful parts. As a result, choosing the most relevant features is important for better system efficiency and recognition performance. Optimization techniques are commonly used to solve this problem because they are flexible and applicable across different fields.

Optimization techniques are generally divided into single-objective and multi-objective types. Some well-known multi-objective algorithms are NSGA-II [5-10], Multi-Objective

Particle Swarm Optimization (MOPSO) [11], MOEA/D [12], and the multi-objective Whale Optimization Algorithm [13].

In recent years, recent multi-objective optimization algorithms, such as the marine predator algorithm, which uses epsilon-dominance and a Pareto archive mechanism, have also been developed [14].

This study proposes a face recognition framework based on MOPSO for selecting discriminative facial blocks.

The objective is to enhance both identification accuracy and verification performance by identifying the most informative blocks of the face. The proposed approach also examines alternative configurations that may improve the selection process and overall system performance.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed methodology, including block partitioning, Binarized Statistical Image Features (BSIF)-based feature extraction, dimensionality reduction, and MOPSO-based block selection. Section 4 reports the experimental results on the AT&T (Formerly ORL), Georgia Tech (GT), AR, and Labeled Faces in the Wild (LFW) datasets. Section 5 provides a comparative analysis with existing methods. Section 6 concludes the paper and outlines future research directions.

The principal contributions of this study are as follows:

This study presents a block-based method to preserve local facial details while reducing the effects of irrelevant blocks.

A multi-objective framework using MOPSO selects discriminative block combinations by minimizing Equal Error Rate (EER) and maximizing Rank-1 accuracy.

The selection of a limited number of blocks, this method creates a compact set of features, reducing dimensionality while still maintaining accuracy.

The effectiveness of the method is demonstrated through extensive experiments on both controlled and unconstrained datasets.

Existing approaches primarily focus on feature-level selection and often neglect spatial structure within facial images. This work addresses this limitation by selecting discriminative facial blocks instead of individual features.

2. WORK IN THIS AREA

Face recognition systems have been extensively studied, with many approaches relying on optimization techniques for feature selection and classification. However, existing methods often fail to jointly address identification and verification performance within a unified optimization framework.

Early works have focused on integrating Particle Swarm Optimization (PSO) into feature selection processes. For instance, Khan et al. [15] employed PSO to select relevant sub-bands from wavelet-based representations, improving feature extraction efficiency. Tabassum et al. [16] integrated Discrete Wavelet Transform with Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Convolutional Neural Networks (CNN) to enhance recognition performance. Eleyan [17] investigated the use of PSO for selecting discriminative features in face recognition tasks. These approaches demonstrate the effectiveness of optimization in reducing feature redundancy.

Several hybrid methods have been introduced to combine optimization algorithms with classical and machine learning techniques. Gulshad and Marie-Sainte [18] proposed a projection pursuit method combined with Support Vector Machine (SVM) classifiers. Abdulameer et al. [19] developed an adaptive PSO variant to optimize SVM parameters. Annamalai [20] combined texture descriptors such as LTP and BRISK with Firefly Optimization for feature refinement, followed by classification using Deep Belief Networks. Revina and Emmanuel [21] introduced a system combining Whale Grasshopper Optimization with a Multi-Support Vector Neural Network, using SIFT and SLDP descriptors for feature extraction. Junior and Yen [22] proposed a PSO-based method to automatically design CNN architectures. Kalaiarasi and Rani [23] applied PSO to optimize CNN for face recognition. Appati et al. [24] applied PCA for dimensionality reduction followed by PSO-based feature optimization.

Other studies have explored more complex hybrid frameworks. Subramanian et al. [25] proposed a hybrid fuzzy-genetic PSO-based optimization approach, while Pati et al. [26] introduced a gradient-based PSO method for Block-ICA feature selection.

More recent approaches have incorporated deep learning models with optimization strategies. Soni et al. [27] introduced a hybrid framework combining Multi-Verse Optimization with deep neural networks for feature selection and classification.

Despite these advances, most existing methods focus on selecting individual features or optimizing model parameters, without explicitly considering the spatial structure of the face image. Only a limited number of studies, such as Chalabi et al. [1], have explored block-based feature selection strategies. However, these approaches are generally formulated as single-objective optimization problems and do not address the trade-off between verification and identification performance.

This limitation motivates the proposed approach, which introduces a block-based representation combined with MOPSO to jointly optimize multiple performance criteria and better capture discriminative facial blocks.

3. PROPOSED METHODOLOGY

The overall framework of the proposed system is illustrated in Figure 1 and consists of two main phases: a training phase and a testing phase.

During the training phase, each input face image is first partitioned into local blocks. Feature extraction is performed on each block using Binarized Statistical Image Features, followed by dimensionality reduction using PCA and LDA. The resulting feature representations are then provided to the MOPSO module, where different combinations of blocks are evaluated. Each candidate solution is assessed based on two criteria: EER and Rank-1 accuracy. The optimization process generates a set of non-dominated solutions. The final optimal block combination is selected and stored as the global best solution.

During the testing phase, the same preprocessing steps are applied to the input image. The feature vector is then constructed using the selected block combination obtained during training. Finally, classification is performed using a nearest neighbor classifier to determine the identity of the input face.

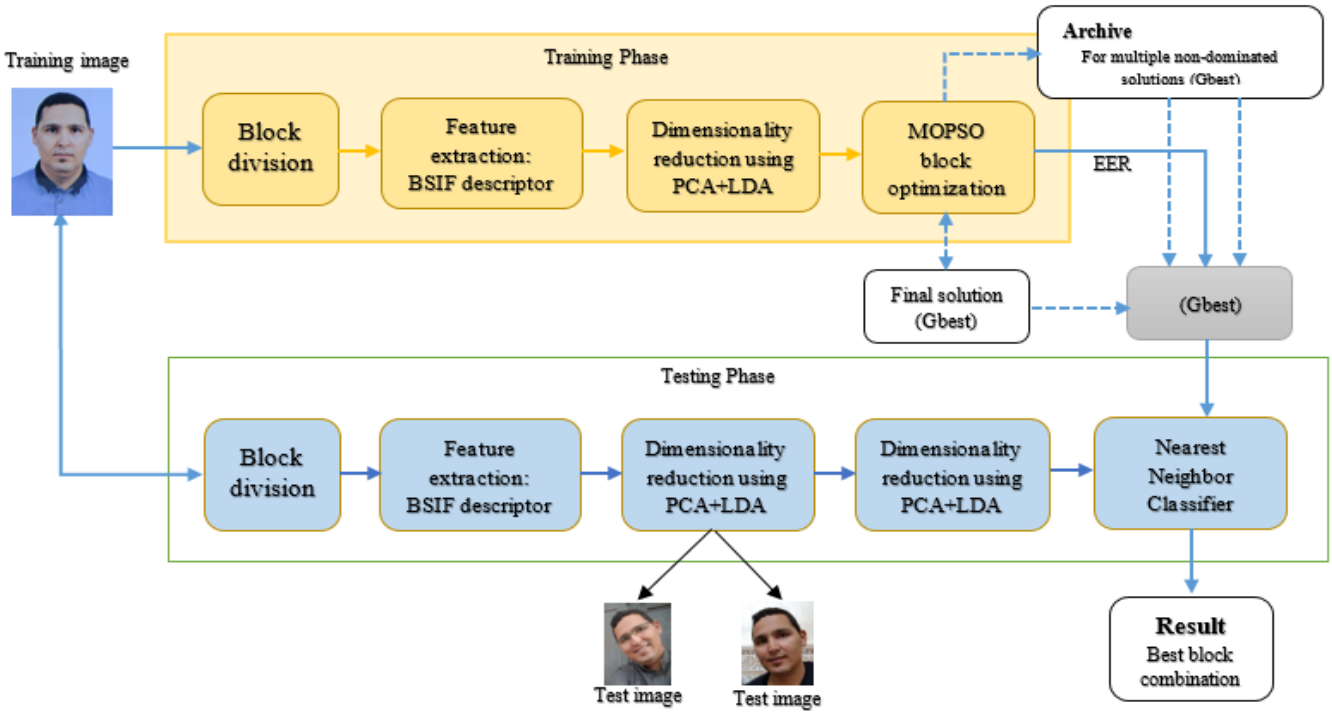

Figure 1. General diagram for the proposed method

3.1 Block division

As illustrated in Figure 2, a block-based partitioning strategy is applied as an initial preprocessing step. Each face image is divided into 16 non-overlapping blocks arranged in a regular 4 × 4 grid. This representation enables the preservation of local spatial information by isolating distinct facial blocks such as the eyes, nose, and mouth.

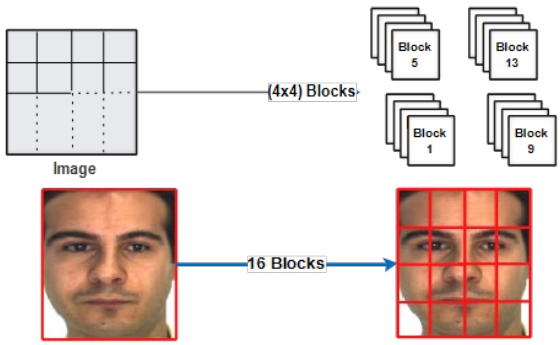

Figure 2. Block division illustration

The use of local blocks allows the system to capture variations that may affect specific blocks of the face, including illumination changes or facial expressions. Instead of relying on the entire image, this approach facilitates a more selective representation by focusing on informative blocks. It also reduces the influence of less informative areas.

The choice of 16 blocks provides a balance between spatial detail and computational complexity. A lower number of blocks may lead to loss of local information, whereas a higher number increases the dimensionality of the search space during optimization. Therefore, this configuration ensures an effective trade-off between representation capability and optimization efficiency.

In this work, block selection is not performed manually. Instead, subsets of blocks are automatically selected using an optimization process. This allows the model to identify the most relevant blocks for face recognition. The division scheme is illustrated in Figure 2.

The number of selected blocks d was empirically set to 4 after preliminary experiments on the AT&T dataset. Increasing d beyond 4 yielded marginal improvements in Rank-1 accuracy (less than 1%). However, it doubled the search space size, while d = 3 caused a drop of about 5%. Therefore, d = 4 provides a favorable trade-off between representational power and optimization complexity.

3.2 Feature extraction using Binarized Statistical Image Features

In this phase, feature extraction is performed independently on each image block using the BSIF descriptor [28], as illustrated in Figure 3. BSIF relies on a set of predefined linear filters to capture local texture information around each pixel.

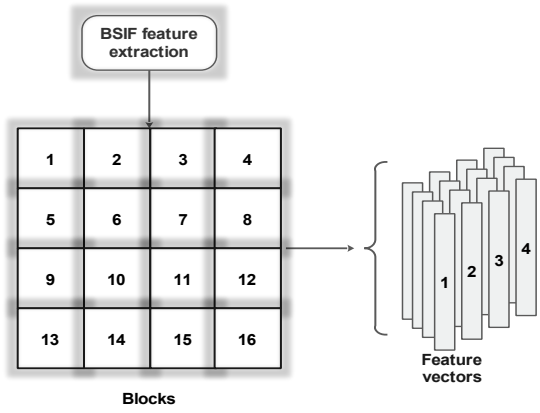

Figure 3. Feature extraction using Binarized Statistical Image Features (BSIF) descriptor

Given an image block X of size $m \times n$, each filter $\emptyset_i$ of size $k \times k$ is convolved with the image to produce a response

value. The filter response is computed as:

$$r_i = \sum_{m,n} \emptyset_i(m,n)X(m,n) \quad (1)$$

where, $\emptyset_i$ represents the i-th filter, and $i = 1, 2, \ldots, N$. With N denoting the number of statistically independent filters.

Each response rr_i is then binarized to generate a binary code according to the following rule:

$$b_i = \begin{cases} 1 & if\ r_i > 0 \\ 0 & otherwise \end{cases} \quad (2)$$

The binary outputs obtained from all filters are combined to form a binary string for each pixel. These binary values are then aggregated into a histogram, which represents the final feature vector of the block. It encodes local texture characteristics of the block.

The performance of the BSIF descriptor is mainly influenced by two parameters: the filter size k and the number of filters N. In this work, a fixed configuration is used to ensure consistency across all experiments.

3.3 Dimensionality reduction

After the feature extraction stage, the concatenation of block-level descriptors results in a high-dimensional feature vector. This increase in dimensionality may introduce redundancy and negatively affect classification performance. To address this issue, a dimensionality reduction step is applied. It uses a combination of PCA and LDA.

First, PCA is used to project the feature vectors into a lower-dimensional space by preserving the directions of maximum variance [29]. This step reduces noise and removes redundant information while retaining the global structure of the data.

Next, LDA is applied to enhance class separability by maximizing the between-class variance and minimizing the within-class variance [30]. This process leads to the computation of an optimal transformation matrix W, defined as:

$$W_{opt} = \arg\max_{W} \frac{|W^T S_B W|}{|W^T S_W W|} = [W_1 W_2 \cdots W_d] \quad (3)$$

where, S_B and S_W represent the between-class and within-class scatter matrices, respectively.

The optimal projection matrix W is obtained by solving the following generalized eigenvalue problem:

$$S_W^{-1} S_B W_j = W_j \lambda_j \quad (4)$$

where, λ_j are the eigenvalues and W_j are the corresponding eigenvectors. The final transformation matrix is constructed by selecting the leading eigenvectors associated with the largest eigenvalues.

By combining PCA + LDA, the resulting feature space becomes both compact and discriminative, which improves the performance of the subsequent classification stage. The PCA step retained 98% of the variance, reducing the feature dimension to approximately 150 components. Then LDA further reduced the dimension to C-1, where C is the number of classes (e.g., 39 for AT&T). PCA is first applied to avoid the small sample size problem before applying LDA.

Crucially, the PCA and LDA transformation matrices are computed only on the training set of each fold. The same transformations are then applied to the test set. No information from the test set is used during training.

3.4 Particle Swarm Optimization algorithm

SO is a population-based bio-inspired algorithm that simulates the collective foraging behavior of bird flocks. In PSO, each candidate solution is represented as a particle navigating the search space. The goal is to optimize an objective function [5, 31-38].

Each particle maintains two key pieces of information:

Personal best (Pbest): the best solution found by the particle itself during the search.

Global best (Gbest): the best solution discovered by the entire swarm.

During each iteration, particles update their positions and velocities based on both Pbest and Gbest, allowing the swarm to explore promising blocks of the search space efficiently.

In the context of the proposed system, each particle represents a candidate combination of blocks selected from the partitioned face image. The search space is therefore defined as all possible combinations of the 16 blocks. Each particle selects a fixed number of blocks (e.g., 4). The objective of PSO is to identify the subset of blocks that maximizes recognition accuracy while minimizing the EER.

3.5 Multi-Objective Particle Swarm Optimization for block selection

The MOPSO [11] is an extension of the standard PSO algorithm designed to handle multi-objective optimization problems. Similar to PSO, MOPSO updates the velocity and position of particles iteratively. However, unlike single-objective PSO, it generates a set of Pareto-optimal solutions rather than a single solution. These solutions are stored in an external archive, as illustrated in Figure 4.

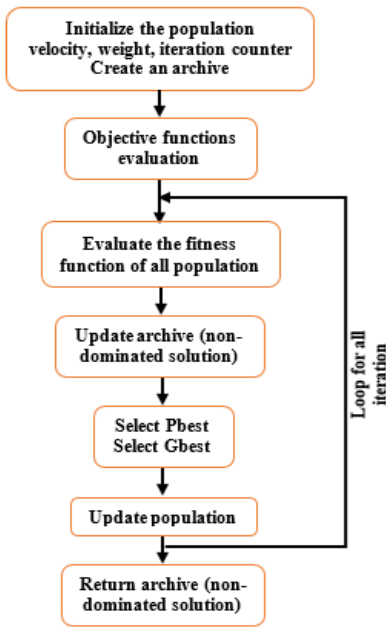

Figure 4. Multi-Objective Particle Swarm Optimization (MOPSO) flowchart

The MOPSO parameters were set as follows: population

size = 50, maximum iterations = 1000. The inertia weight w = 0.4, acceleration coefficients c1 = c2 = 2.0. The crowding distance was used to maintain archive diversity, and the mutation rate was set to 0.1.

To enhance exploration and avoid premature convergence, a polynomial mutation operator is applied to each particle's position with probability Pm = 0.1. For each selected particle, a randomly chosen block index is replaced by a new valid index sampled uniformly from the set of 16 blocks. This mutation is applied after the velocity update step.

The overall procedure of the proposed MOPSO-based block selection is summarized in Algorithm 1.

Algorithm 1. Multi-Objective Particle Swarm Optimization (MOPSO) for block selection

Input: Feature vectors
•Number of particles N
•Maximum iterations Tmax
•Number of selected blocks d
•Objective functions F1 (EER), F2 (Rank-1)
Output: Optimal subset of blocks
•Pareto archive A (non-dominated block ombinations)
1: Initialize particle positions Xi and velocities Vi randomly
2: Initialize personal best Pbesti = Xi
3: Initialize archive A with non-dominated solutions
4: t = 1
5: while t ≤ Tmax do
6: for each particle Xi do
7: Evaluate objectives F1(Xi), F2(Xi)
8: Update Pbesti based on dominance
9: end for
10: Update archive A using non-dominated sorting
11: Apply crowding distance to maintain diversity
12: for each particle Xi do
13: Select Gbest from archive A
14: Update velocity:
Vi = w·Vi + c1·r1·(Pbesti − Xi) + c2·r2·(Gbest − Xi)
15: Update position:
Xi = Xi + Vi
16: end for
17: t = t + 1
18: end while
19: return A

The Gbest is selected from the archive using a crowding-distance-based roulette wheel or random selection strategy to maintain diversity.

3.5.1 Dominance concept

A candidate solution s is said to dominate another solution s' ($denoted\ s \succ s'$) if:

$$\forall\, i \in \{1,2,\dots,M\}: f_i(s) \le f_i(s') \tag{5}$$

$$\exists\, j \in \{1,2,\dots,M\} : f_j(s) < f_j(s') \tag{6}$$

where, M is the number of objective functions.

The non-dominated set consists of solutions for which no other solution exhibits dominance within the population. These solutions form the Pareto front.

3.5.2 Crowding distance

To ensure diversity among solutions, a crowding distance mechanism is applied to the non-dominated set. The crowding distance measures the density of solutions surrounding a particular solution, allowing the algorithm to maintain a well-distributed Pareto front. Border solutions are assigned an infinite distance. While intermediate solutions are evaluated based on the mean distance to adjacent solutions along each objective dimension.

Particle initialization and representation

The MOPSO population is initialized using block indices corresponding to the image partition shown in Figure 2. Each particle represents a subset of 4 blocks (dimension d = 4). These blocks are concatenated to form a candidate solution: P = [b1, b2, b3, b4].

where b_i refers to the index of a block in the 16-block grid. Figure 5 illustrates the particle initialization procedure.

After initialization, each particle's position and velocity vectors are set. The algorithm parameters including acceleration coefficients $\boldsymbol{c_1}$ *and* $\boldsymbol{c_2}$, population size, and maximum iterations are initialized.

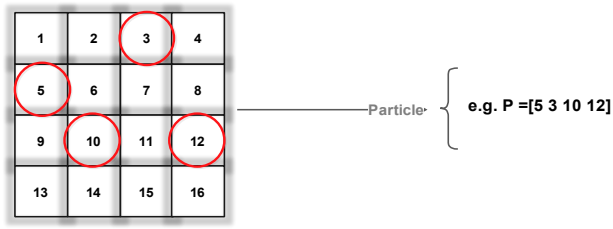

Figure 5. Particle initialization

Objective functions and fitness evaluation

Each particle is evaluated using two objective functions:

F1: Minimize the EER. EER is defined when the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR):

$$EER\ (error\ equal\ rate): FAR = \frac{FA}{TA}, FRR = \frac{FR}{TA} \tag{7}$$

where, FA and FR are the number of false acceptances and false rejections, respectively. TA is the total number of attempts.

F2: Maximize Rank-1 accuracy:

$$RANK - 1 = \frac{N_i}{N} \times 100\% \tag{8}$$

where, N_i is the number of correctly identified images, and N is the total number of test images.

MOPSO iterative procedure

The optimization process iterates as follows:

1. Evaluate each particle using F1 and F2.
2. Update the archive of non-dominated solutions (Pareto front) based on dominance and crowding distance.
3. Update each particle's *Pbest and Gbest*.
4. Update velocity and position of particles according to standard PSO equations.
5. Repeat steps 1–4 until the maximum number of iterations is reached.

At the end of the process, the archive contains multiple Pareto-optimal solutions (different block combinations). A final solution is selected from the Pareto set according to a trade-off between EER and Rank-1 accuracy. This forms the optimized block selection used for classification.

Illustrative Example of MOPSO Block Selection

To clarify the optimization process, a simple example of particle evaluation is provided. Assume that a particle

represents the following block combination: P = [5,3,10,12]

This means that only blocks 5, 3, 10, and 12 are selected from the 16-block partition of the face image. The corresponding BSIF features are extracted from these blocks. They are concatenated and then reduced using PCA and LDA.

The resulting feature vector is evaluated using the nearest neighbor classifier. Suppose that this particle produces the following results: Rank-1 accuracy = 93.5%The resulting feature vector is evaluated using the nearest neighbor classifier. Suppose that this particle produces the following results: Rank-1 accuracy = 93.5%.

EER = 1.08%

Now consider another particle: P′ = [1,4,5,13]

which yields: Rank-1 accuracy = 96%

EER = 1.97%

In this case, neither solution dominates the other. The first particle provides a lower EER, while the second achieves higher accuracy. Therefore, both solutions are considered non-dominated and are stored in the Pareto archive.

During the optimization process, multiple such particles are evaluated, and the archive is updated accordingly. The crowding distance mechanism ensures that solutions are well distributed across the trade-off space between accuracy and EER.

At the end of the iterations, a final solution is selected from the Pareto set. For example:

If the application requires higher security → solution with lower EER is selected

If the application requires better identification → solution with higher Rank-1 is selected.

This example illustrates how MOPSO enables the selection of discriminative block combinations while balancing competing objectives.

4. RESULTS AND DISCUSSION

All experiments were repeated for 30 independent runs with different random seeds. Reported results were the mean ± standard deviation (std) of the best solution found in each run. The selection was based on Pareto dominance and crowding distance.

This section presents the experimental evaluation of the proposed method on four benchmark datasets: AT&T [31], GT [32], AR [33], and LFW [34]. The objective is to assess the effectiveness of the proposed block selection strategy in both identification and verification tasks.

All experiments were conducted using the same parameter configuration to ensure consistency. The population size was set to 50 particles, and the maximum number of iterations was fixed at 1000. For feature extraction, a BSIF configuration was used with a filter size of 17 × 17 and a filter length of 12 bits.

The performance of the system was evaluated using two metrics: Rank-1 accuracy and EER, which correspond to the two objectives of the optimization process.

4.1 Experiment results using AT&T database

The AT&T dataset consists of 400 grayscale images from 40 subjects, with 10 images per subject. The images include variations in facial expression, illumination, and the presence or absence of glasses. Each image has a resolution of 92 × 112 pixels. Figure 6 illustrates sample facial images from the AT&T dataset.

In this experiment, a standard protocol was adopted in which 5 images per subject were used for training and the remaining 5 for testing.

The obtained results are summarized in Table 1 and illustrated in Figure 7. The MOPSO algorithm generated a set of non-dominated solutions corresponding to different block combinations.

Figure 6. Sample images of AT&T database

Figure 7 shows the convergence of the hypervolume indicator (mean ± std over 30 runs) for the AT&T dataset. The shaded area represents one standard deviation.

From Table 1, the solution [9,1,12,6] achieved the lowest EER of 1.012%, with a Rank-1 accuracy of 91.5%.

On the other hand, the highest accuracy of 96% was obtained using the block combination [13,4,5,1], with a corresponding EER of 1.974%.

This behavior reflects the trade-off between the two objectives. An increase in accuracy was associated with a slight increase in EER, confirming the effectiveness of the multi-objective optimization strategy.

An additional observation is the frequent selection of certain blocks, particularly blocks 1, 5, 12, and 13. These blocks appear multiple times across different solutions. This indicates that these blocks contain more discriminative facial information and play a key role in the recognition process.

Overall, the results demonstrate that selecting a small subset of blocks is sufficient to achieve high recognition performance, while reducing the dimensionality of the feature space.

Table 1. AT&T database results

Particle	Rank-1 %	EER %
[9 1 12 6]	91,5	1,012
[1 12 13 8]	93	1,051
[10 1 5 8]	93,5	1,083
[13 1 12 5]	95	1,435
[13 4 5 1]	96	1,974

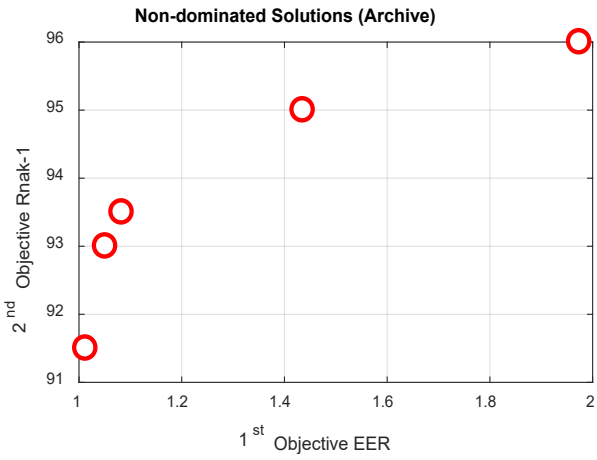

Figure 7. Convergence for AT & T database

4.2 Experiment results using Georgia Tech database

The GT dataset consists of 750 images from 50 subjects, with 15 images per subject. Representative samples from the dataset are shown in Figure 8. The images were acquired under unconstrained conditions, including variations in illumination, facial expression, pose, and scale. The original resolution was 640 × 480 pixels, and a cropped version of size 150 × 200 pixels was used in this study.

Figure 8. Sample images of Georgia Tech (GT) database

The obtained results are presented in Table 2 and Figure 9. In contrast to the AT&T dataset, the performance on GT is significantly lower, with the best Rank-1 accuracy reaching only 13.5% and the lowest EER equal to 42.51%.

This performance degradation can be attributed to several factors. First, the GT dataset exhibits higher intra-class variability due to changes in pose, scale, and lighting conditions. These variations reduce the consistency of local texture patterns extracted using BSIF, especially when applied to small blocks.

Table 2. Georgia Tech (GT) database results

Particle	Rank-1	EER
[13 12 5 15]	9	42.51
[16 9 1 13]	10	43.09
[16 9 15 5]	11	43.44
[16 1 6 7]	12	44.48
[9 1 12 16]	12.5	44.58
[10 12 1 16]	13	46.47
[6 5 4 1]	13.5	47

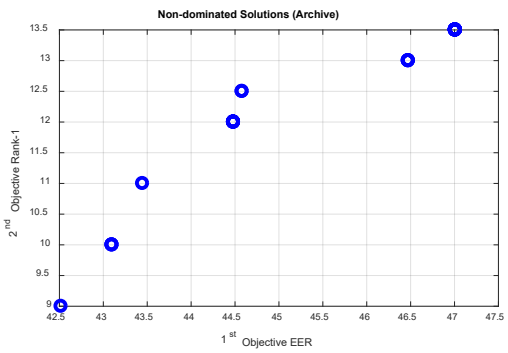

Figure 9. Convergence for Georgia Tech (GT) database

Second, the use of a fixed number of blocks (d = 4) limits the amount of discriminative information captured from each face. While this configuration is effective for controlled datasets such as AT&T, it becomes insufficient for more complex datasets where facial variations are more pronounced.

Third, the nearest neighbor classifier relies directly on feature similarity and does not incorporate a learning mechanism. This limits its ability to handle high variability between samples, further affecting identification performance in unconstrained scenarios.

Despite the low accuracy, an important observation emerges regarding the diversity of selected blocks. Most blocks are selected at least once across the Pareto solutions, indicating that no single region consistently dominates the recognition process. This contrasts with the AT&T results, where specific blocks were repeatedly selected.

This behavior suggests that, in challenging conditions, discriminative information is distributed across the entire face rather than concentrated in specific blocks. As a result, selecting a small subset of blocks may not be sufficient to capture the variability of the data

4.3 Experiment results using AR database

The AR dataset contains 4,000 color images of 126 subjects, captured under controlled conditions with variations in facial expression, illumination, and occlusions such as sunglasses and scarves. The images were acquired in two sessions separated by a time interval of approximately two weeks. Representative examples illustrating these variations are shown in Figure 10.

In this experiment, 13 images per subject were used for training and 13 for testing. The obtained results are presented in Table 3 and Figure 11.

Figure 10. Sample images of AR database

Table 3. AR database results

Particle	Rank-1%	EER%
[1 3 5 2]	100	0
[1 3 8 9]	100	0
[9 5 6 1]	100	0
[5 8 10 1]	100	0
[13 6 1 4]	100	0
[15 9 4 1]	100	0

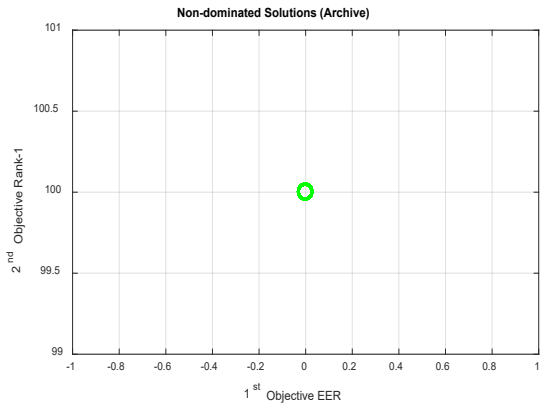

Figure 11. Convergence for AR database

All evaluated block combinations achieved a Rank-1 accuracy of 100% and an EER of 0%. While these results indicate perfect recognition performance, they should be interpreted with caution.

This outcome can be explained by the relatively controlled acquisition conditions of the AR dataset. Despite the presence of occlusions, the images maintain a consistent frontal pose and limited variation in scale and background. These characteristics reduce intra-class variability and facilitate the extraction of stable local features using BSIF.

Another contributing factor is the block-based representation. The repeated presence of block 1 across all optimal solutions suggests that certain facial blocks consistently contain strong discriminative information. In this dataset, these blocks remain stable across samples, which simplifies the recognition task.

However, such performance may not generalize to more challenging datasets. The results observed on the GT and LFW datasets indicate that the proposed approach is sensitive to variations in pose, illumination, and image quality. Therefore, the AR dataset should be considered a controlled scenario rather than a representative real-world condition.

The perfect scores on AR (100% Rank-1, 0% EER) should be interpreted with caution. They were achieved under controlled conditions (frontal pose, stable illumination) and after ensuring no data leakage. On more challenging datasets, performance degrades as shown in Sections 4.2 and 4.4.

4.4 Experiment results using Labeled Faces in the Wild database

For the LFW dataset, a subset of 158 subjects with at least 10 images each was selected. For each subject, 5 images were randomly chosen for training and the remaining 5 for testing. This protocol is similar to the one used for AT&T and AR to maintain consistency. No subject appears in both training and testing sets.

The LFW dataset contains more than 13,000 images of 5,749 individuals. These images were collected from the web under unconstrained conditions. The images exhibit significant variations in pose, illumination, background, and image quality, as shown in Figure 12.

In this experiment, a subset of 158 subjects was selected, with 10 images per subject. The images were preprocessed by cropping the facial block. The obtained results are reported in Table 4 and illustrated in Figure 13.

Figure 12. Sample images from the Labeled Faces in the Wild (LFW) database

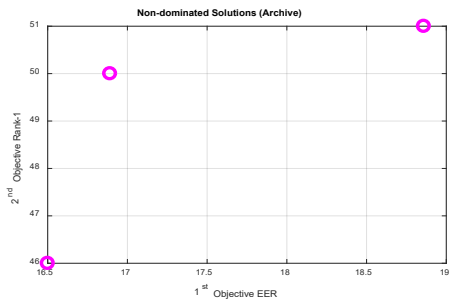

Figure 13. Convergence for Labeled Faces in the Wild (LFW) database

The best achieved Rank-1 accuracy was 51%, while the lowest EER reached 16.50%. These results were lower compared to those obtained on the AT&T and AR datasets.

This performance can be explained by the high variability present in the dataset. Unlike controlled datasets, LFW includes large variations in pose, occlusion, lighting conditions, and image resolution. These factors affect the stability of local texture descriptors such as BSIF, especially when applied to small image blocks.

Another limitation arises from the use of a reduced number of selected blocks. While selecting four blocks reduces dimensionality, it also limits the amount of discriminative information captured from the face. In unconstrained scenarios, relevant information is often distributed across multiple blocks, making it difficult to represent a subject using a small subset of blocks.

In addition, the nearest neighbor classifier does not model complex decision boundaries, which reduces its effectiveness in handling large intra-class variations.

Despite these limitations, the results remain consistent with those obtained on the GT dataset, confirming that the proposed method behaves similarly under unconstrained conditions. This consistency indicates that the method is stable, even when the recognition task becomes more challenging.

Table 4. Comparative study of the proposed method with the existing approaches

Reference	Method	Verification EER %	Identification Rank-1 %
This paper	PSO based block feature selection	1.083	93.50
This paper (with 8 blocks and 10 iteration)	PSO based block feature selection	0.500	97.00
BSIF (on the whole image)	Feature extraction BSIF	5.000	40.83
[35]	BPSO based feature selection (187 feature)	----------	98.14
[36]	DCT+PSO feature selection	----------	94.70

Note: BSIF: Binarized Statistical Image Features; PSO: Particle Swarm Optimization.

Table 5. Labeled Faces in the Wild (LFW) database results

Particle	Rank-1	EER
[10 6 2 12]	50	16,89
[7 6 10 5]	51	18,85
[9 12 14 6]	46	16,50

The overall trade-off between verification and identification performance across all datasets is illustrated in Figure 14. The Pareto front highlights the relationship between EER and Rank-1 accuracy obtained using the proposed MOPSO-based block selection.

The Pareto front shows that solutions with lower EER tend to correspond to slightly reduced Rank-1 accuracy. Conversely, higher accuracy is associated with a moderate increase in EER. This behavior confirms the effectiveness of the multi-objective optimization framework.

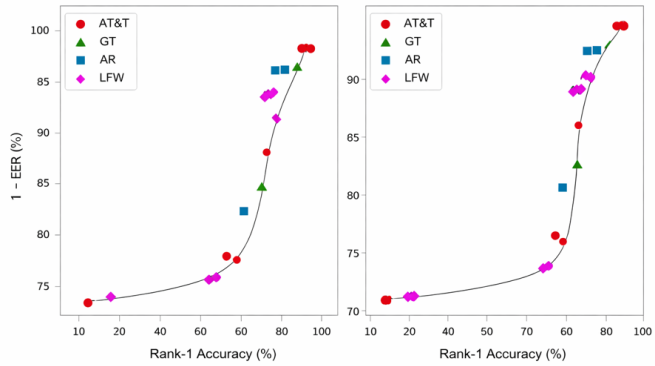

Figure 14. Pareto front illustrating the trade-off between Equal Error Rate (EER) and Rank-1 accuracy achieved by the proposed method on benchmark datasets

In addition, the distribution of solutions differs across datasets. Controlled datasets such as AT&T and AR exhibit more compact and stable Pareto fronts, whereas unconstrained datasets such as GT and LFW show more dispersed solutions, which reflects higher variability in facial conditions.

The poor performance on LFW (max 51% Rank-1) suggests that the BSIF descriptor with a 17 × 17 filter is not robust to severe pose and illumination variations. Possible remedies include: (i) using a learnable descriptor (e.g., from a CNN), (ii) increasing the number of selected blocks d as shown in Section 4.6, and (iii) replacing the nearest neighbor classifier with a more powerful model such as SVM or a shallow neural network.

4.5 Stability analysis

To evaluate the stability of the proposed method, multiple independent runs of the MOPSO algorithm were performed. The results showed consistent convergence behavior across runs, indicating that the optimization process is stable and not significantly affected by random initialization.

4.6 Trade-off analysis

The obtained Pareto solutions illustrate a clear trade-off between EER and Rank-1 accuracy. Solutions with lower EER tend to produce slightly lower accuracy, while higher accuracy is associated with a moderate increase in EER. This behavior confirms the necessity of using a multi-objective optimization framework.

4.7 Complexity discussion

The computational complexity of the proposed method depends on three main components: feature extraction, dimensionality reduction, and the optimization process. While MOPSO introduces an iterative search procedure, the reduction in feature dimensionality compensates for this cost during classification.

4.8 Generalization discussion

The experimental results on GT and LFW datasets highlight the limitations of the proposed method under unconstrained conditions. This behavior suggests that local texture descriptors combined with a limited number of blocks may not fully capture large variations in pose and illumination.

4.9 Computational complexity analysis

The computational cost of the proposed method has three components: (i) BSIF feature extraction per block: $O(16 \times H \times W \times k^2)$, (ii) PCA+LDA: $O(\min(N^2, D^2))$, (iii) MOPSO: $O(T_{max} \times N_{particles} \times (C + M))$. Here, C is the cost of kNN classification. The total offline training time for AT&T was approximately 25 minutes on a standard CPU (Intel i7, 16GB RAM).

However, the testing phase only requires feature extraction on the selected 4 blocks and a single kNN query. This approximately took 0.3 seconds per image. This makes the method suitable for real-time applications after training.

5. COMPARATIVE STUDY

To further evaluate the performance of the proposed approach, a comparison with existing methods from the literature is presented in Table 5. The comparative evaluation is extended to include both classical optimization-based approaches and recent learning-based methods in order to provide a more comprehensive assessment. As shown in Table 6, the proposed method achieves competitive performance while relying on a reduced number of selected features. Unlike traditional approaches that operate on full feature representations, the proposed method focuses on selecting a subset of informative facial blocks.

Table 6. Quantitative evaluation of the proposed method versus existing approaches

Method	Dataset	Rank-1 (%)	EER (%)	Features
Proposed (MOPSO, d = 4)	AT&T	95.8 ± 0.8	1.98 ± 0.22	4 × BSIF
Proposed (MOPSO, d = 8)	LFW	63.7 ± 2.0	14.2 ± 1.5	8 × BSIF
Random Block Selection	AT&T	72.3 ± 3.1	8.5 ± 1.2	same
BSIF (full image)	AT&T	40.8	5.0	full
[35]	AT&T	98.1	-	187
[17]	AT&T	94.7	-	-

Note: BSIF: Binarized Statistical Image Features; MOPSO: Multi-Objective Particle Swarm Optimization: EER: Equal Error Rate.

This results in a compact representation while maintaining recognition performance. Compared to BSIF applied on the entire image, a significant improvement is observed in both Rank-1 accuracy and EER. This confirms the effectiveness of block-wise selection. When compared to optimization-based methods such as BPSO and DCT + PSO, the proposed approach achieves comparable or better performance while using fewer features. This indicates that selecting discriminative blocks is more effective than increasing feature dimensionality. In addition, an extended configuration using a larger number of blocks shows further improvement. This suggests that the trade-off between dimensionality and performance can be controlled through the number of selected blocks [35, 36].

The comparison also highlights that the proposed method achieves this performance using a significantly reduced number of features. This directly impacts computational efficiency. This aspect is particularly important for real-time or resource-constrained applications, where both accuracy and efficiency must be considered.

6. CONCLUSION

In this paper, a block-based feature selection framework using MOPSO has been presented for face recognition.

The proposed framework is based on block-level representation combined with multi-objective optimization for discriminative feature selection.

The proposed method simultaneously optimizes two conflicting objectives, namely the EER and the Rank-1 accuracy, leading to a set of Pareto-optimal solutions. This allows flexible selection of block combinations depending on the application requirements.

The experimental evaluation on AT&T, GT, AR, and LFW datasets shows that the method achieves strong performance on controlled datasets, while maintaining consistent behavior under more challenging conditions. The results also demonstrate that a reduced number of selected blocks can effectively represent facial information, which reduces feature dimensionality without significant loss in performance.

However, the experiments on GT and LFW highlight certain limitations. The use of a fixed number of blocks and a simple nearest neighbor classifier restricts the ability of the system to handle large variations in pose, illumination, and image quality.

The limitations of this study are: (1) the fixed number of selected blocks d = 4 is insufficient for unconstrained datasets; (2) the nearest neighbor classifier is sensitive to noise; (3) the BSIF descriptor is handcrafted and may not generalize.

Therefore, future work will: (i) adapt d dynamically using a variable-length particle representation; (ii) replace kNN with a lightweight neural network (e.g., a 2-layer MLP) trained on the selected blocks; (iii) integrate deep features (e.g., from a pretrained MobileNet) as an alternative to BSIF.

Future work will focus on improving the robustness of the approach by adapting the number of selected blocks dynamically and integrating more advanced classification models. In addition, combining local block selection with deep feature representations may further enhance performance in unconstrained environments.

REFERENCES

[1] Chalabi, N.E., Attia, A., Bouziane, A., Akhtar, Z. (2021). Particle swarm optimization based block feature selection in face recognition system. Multimedia Tools and Applications, 80: 33257-33273. https://doi.org/10.1007/s11042-021-11367-0

[2] Amirgaliyev, B., Mussabek, M., Rakhimzhanova, T., Zhumadillayeva, A. (2025). A review of machine learning and deep learning methods for person detection, tracking and identification, and face recognition with applications. Sensors, 25(5): 1410. https://doi.org/10.3390/s25051410

[3] Gulshad, S., Aldahlawi, A. (2025). DArFace: Deformation aware robustness for low quality face recognition. In 2025 IEEE International Joint Conference on Biometrics (IJCB), Osaka, Japan, pp. 1-11. https://doi.org/10.1109/IJCB65343.2025.11411390

[4] Ali, W., Tian, W., Din, S.U., Iradukunda, D., Khan, A.A. (2021). Classical and modern face recognition approaches: A complete review. Multimedia Tools and Applications, 80: 4825-4880. https://doi.org/10.1007/s11042-020-09850-1

[5] Xiaodong, Y., Zefan, C. (2017). Particle swarm optimization and cuckoo search paralleled algorithm. In 2017 International Conference on Computer Communication and Informatics, Chengdu, China, pp. 2236-2240. https://doi.org/10.1109/CompComm.2017.8322933

[6] Goldberg, D.E., Holland, J.H. (1988). Genetic algorithms and machine learning. Machine Learning, 3: 95-99. https://doi.org/10.1023/A:1022602019183

[7] Dorigo, M., Birattari, M. (2011). Ant colony optimization. In Encyclopedia of Machine Learning, Boston, USA, pp. 36-39. https://doi.org/10.1007/978-0-387-30164-8_22

[8] Kennedy, J. (2011). Particle Swarm Optimization. In Encyclopedia of Machine Learning, pp. 760-766. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_630

[9] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2): 182-197. https://doi.org/10.1109/4235.996017

[10] Deb, K., Jain, H. (2012). Handling many-objective problems using an improved NSGA-II procedure. In 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, pp. 1-8. https://doi.org/10.1109/CEC.2012.6256519

[11] Reyes-Sierra, M., Coello, C.A.C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International Journal of Computational Intelligence Research, 2(3): 287-308.

[12] Zhang, Q., Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 11(6): 712-731. https://doi.org/10.1109/TEVC.2007.892759

[13] Kumawat, I.R., Nanda, S.J., Maddila, R.K. (2017). Multi-objective whale optimization. In TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, Malaysia, pp. 2747-2752. https://doi.org/10.1109/TENCON.2017.8228329

[14] Chalabi, N.E., Attia, A., Bouziane, A., Hassaballah, M. (2023). An improved marine predator algorithm based on epsilon dominance and Pareto archive for multi-objective optimization. Engineering Applications of Artificial Intelligence, 119: 105718. https://doi.org/10.1016/j.engappai.2022.105718

[15] Khan, S.A., Ishtiaq, M., Nazir, M., Shaheen, M. (2018). Face recognition under varying expressions and illumination using particle swarm optimization. Journal of Computational Science, 28: 94-100. https://doi.org/10.1016/j.jocs.2018.08.005

[16] Tabassum, F., Islam, M.I., Khan, R.T., Amin, M.R. (2022). Human face recognition with combination of DWT and machine learning. Journal of King Saud University - Computer and Information Sciences, 34(3): 546-556. https://doi.org/10.1016/j.jksuci.2020.02.002

[17] Eleyan, A. (2019). Particle swarm optimization based feature selection for face recognition. In 2019 Seventh International Conference on Digital Information Processing and Communications (ICDIPC), Trabzon, Turkey, pp. 1-4. https://doi.org/10.1109/ICDIPC.2019.8723831

[18] Gulshad, S., Marie-Sainte, S.L. (2020). Face identification based bio-inspired algorithms. The

International Arab Journal of Information Technology, 17(1). https://doi.org/10.34028/iajit/17/1/14

[19] Abdulameer, M.H., Mohammed, D., Mohammed, S.A., Al-Azawi, M., Al-Mayali, Y.M.H., Alameri, I.A. (2018). Face recognition technique based on adaptive-opposition particle swarm optimization (AOPSO) and support vector machine (SVM). ARPN Journal of Engineering and Applied Sciences, 13(6): 2259-2266.

[20] Annamalai, P. (2020). Automatic face recognition using enhanced firefly optimization algorithm and deep belief network. International Journal of Intelligent Engineering and Systems, 13(5): 19-28. https://doi.org/10.22266/ijies2020.1031.03

[21] Revina, I.M., Emmanuel, W.R.S. (2019). Face expression recognition with the optimization based multi-SVNN classifier and the modified LDP features. Journal of Visual Communication and Image Representation, 62: 43-55. https://doi.org/10.1016/j.jvcir.2019.04.013

[22] Junior, F.E.F., Yen, G.G. (2019). Particle swarm optimization of deep neural networks architectures for image classification. Swarm and Evolutionary Computation, 49: 62-74. https://doi.org/10.1016/j.swevo.2019.05.010

[23] Kalaiarasi, P., Rani, P.E. (2021). Optimizing convolutional neural network using particle swarm optimization for face recognition. Turkish Journal of Computer and Mathematics Education, 12(11): 3672-3679.

[24] Appati, K., Abu, H., Owusu, E., Darkwah, K. (2021). Analysis and implementation of optimization techniques for facial recognition. Applied Computational Intelligence and Soft Computing, 2021: 6672578. https://doi.org/10.1155/2021/6672578

[25] Subramanian, R.R., Mohan, H., Jenny, A.M., Sreshta, D., Prasanna, M.L., Mohan, P. (2021). PSO based fuzzy-genetic optimization technique for face recognition. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, pp. 374-379. https://doi.org/10.1109/Confluence51648.2021.9377028

[26] Pati, R., Pujari, A.K., Gahan, P. (2021). Face recognition using particle swarm optimization based block ICA. Multimedia Tools and Applications, 80: 35685-35695. https://doi.org/10.1007/s11042-021-10792-5

[27] Soni, N., Sharma, E.K., Kapoor, A. (2021). Hybrid meta-heuristic algorithm based deep neural network for face recognition. Journal of Computational Science, 51: 101352. https://doi.org/10.1016/j.jocs.2021.101352

[28] Kannala, J., Rahtu, E. (2012). BSIF: Binarized statistical image features. In 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, pp. 1363-1366.

[29] Turk, M., Pentland, A. (1991). Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1): 71-86. https://doi.org/10.1162/jocn.1991.3.1.71

[30] Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7): 711-720. https://doi.org/10.1109/34.598228

[31] Samaria, F.S., Harter, A.C. (1994). Parameterisation of a stochastic model for human face identification. In Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, Sarasota, USA, pp. 138-142. https://doi.org/10.1109/ACV.1994.341300

[32] Nefian, A.V., Hayes, M.H. (2000). Maximum likelihood training of the embedded HMM for face detection and recognition. In Proceedings 2000 International Conference on Image Processing, Vancouver, Canada, pp. 33-36. https://doi.org/10.1109/ICIP.2000.900885

[33] Martinez, A., Benavente, R. (2016). The AR Face Database: CVC Technical Report, 24. 1998. http://www.cat.uab.es/Public/Publications/1998/MaB1998/CVCReport24.

[34] Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E. (2008). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition.

[35] Ajit Krisshna, N.L., Deepak, V.K., Manikantan, K., Ramachandran, S. (2014). Face recognition using transform domain feature extraction and PSO-based feature selection. Applied Soft Computing, 22: 141-161. https://doi.org/10.1016/j.asoc.2014.05.007

[36] Ramadan, R., Abdel-kader, R. (2025). Face recognition using particle swarm optimization based selected features. International Journal of Signal Processing Image Processing and Pattern Recognition, 2(2): 51-65.

[37] Chatra, M., Bourahla, M. (2024). Agent-based simulation of crowd evacuation through complex spaces. Ingénierie des Systèmes d'Information, 29(1): 83. https://doi.org/10.18280/isi.290110

[38] Chatra, M., Attia, A., Akhrouf, S., Maaref, Z. (2025). Particle swarm optimized ALMMo* for interpretable and accurate diabetic retinopathy detection. Journal of Medical Systems, 49(3): 112-127. https://doi.org/10.5996/jzund.2025.v56i9.2335