

Research Article

Analysis and Implementation of Optimization Techniques for Facial Recognition

Justice Kwame Appati ¹, Huzaifa Abu ¹, Ebenezer Owusu ¹ and Kwaku Darkwah²

¹Department of Computer Science, University of Ghana, Accra, Ghana

²Department of Mathematics, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

Correspondence should be addressed to Ebenezer Owusu; ebeowusu@ug.edu.gh

Received 1 November 2020; Revised 25 February 2021; Accepted 5 March 2021; Published 12 March 2021

Academic Editor: Cheng-Jian Lin

Copyright © 2021 Justice Kwame Appati et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Amidst the wide spectrum of recognition methods proposed, there is still the challenge of these algorithms not yielding optimal accuracy against illumination, pose, and facial expression. In recent years, considerable attention has been on the use of swarm intelligence methods to help resolve some of these persistent issues. In this study, the principal component analysis (PCA) method with the inherent property of dimensionality reduction was adopted for feature selection. The resultant features were optimized using the particle swarm optimization (PSO) algorithm. For the purpose of performance comparison, the resultant features were also optimized with the genetic algorithm (GA) and the artificial bee colony (ABC). The optimized features were used for the recognition using Euclidean distance (EUD), K-nearest neighbor (KNN), and the support vector machine (SVM) as classifiers. Experimental results of these hybrid models on the ORL dataset reveal an accuracy of 99.25% for PSO and KNN, followed by ABC with 93.72% and GA with 87.50%. On the central, an experimentation of the PSO, GA, and ABC on the YaleB dataset results in 100% accuracy demonstrating their efficiencies over the state-of-the-art methods.

1. Introduction

Automated biometric recognition is fast gaining recognition as the most trusted security systems in the 21st century. This is perhaps attributed to the recent significant advances in parallel processing techniques and also the search for most reliable security systems due to the sharp increases in crimes worldwide. The earliest biometric features that were automated for recognition include fingerprints where the unique ridge skin patterns were utilized. Others include the retina, iris, palm, skin, and nose tip. Fingerprints, retina, and iris recognition systems are known to yield very accurate results [1], but hardened criminals, being sensitively aware of the security implications, mostly avoid presenting their biometric features to be captured into databases. Thus, automated face recognition systems are now the obvious choice [2] because people cannot hide their facial images from installed CCTV cameras all the time. This makes the technology the least intrusive and a hotbed research area as

researchers continue to propose newer algorithms that outperform existing ones.

Since automated face recognition study is new as compared to fingerprint and others already stated, the problems associated with it are still eminent. For example, Zhang, Luo, Loy, and Tang [3] perceived the problem of facial landmark detection, which is among the central focus of the system development. Most of the face detection algorithms are slow and produce poor recognition accuracies (Owusu, Zhan, and Mao, 2014). Other unraveled challenges in the face recognition research have to do with occlusion, pose variation, illumination normalization, age, and gender [4]. In unconstrained environments, there is a significant decrease in recognition accuracy, thus making it difficult to accurately identify faces. Therefore, there is a need to have techniques that improve face recognition in these environments. Tu, Li, and Zhao [5] attempted to solve the problem of illumination and pose by using DL-Net and N-Net methods. However, this method could not adequately

account for large-scale normalized albedo images and face recognition in the wild. Another challenge in the face recognition research has to do with testing for the efficacy of the experimental results. There are no standard datasets that is generally recognized by the research community to be used for testing. The use of specific datasets depends on the individual researcher's choice. Most of the datasets are premeditated and therefore do not represent a real-world scenario. In terms of ethnicity too, there is a challenge. Currently, there is no dataset that is well-balanced for race, gender, and age.

The problem of nonuniform illumination also arises when the lighting conditions vary at different angles. Thus, the proportion of light reflected by the face is different. This phenomenon can lead to the misidentification of an individual [6]. Similarly, a random gyration due to individual movement can also lead to misclassifications in 4D recognitions. An input image and interperson image could appear dissimilar due to the rotation of the image [7]. The main purpose of this study is to explore the popular techniques and bring forth an approach that leverage on computational cost. Moreover, this method will take into account illumination, pose, and the facial expression. The proposed approach enhances the outcomes of the principal component analysis (PCA) technique using the optimization techniques approach. Additionally, the improvement in accuracy in this research transform to a general improvement in the security and integrity of biometric locks.

In this study, we explored the question of which is the finest or suitable optimization algorithm to use to maximize recognition. It also responds to which classifier suits the recommended approach and again which method utilizes less computational resource and time. The proposed method for this research requires the preprocessing of image; then, features are subsequently examined and extracted using PCA. This will be followed by the augmentation of the said features using PSO, ABC, and GA with classification culminating the entire process.

2. Related Works

Face recognition is mainly performed in four phases, *visa-a-viz.*, feature extraction, face detection, face synthesis, and recognition [8]. Chihaoui et al. [9] stated that face recognition techniques are mainly in three categories. The first is the use of procedures that require the usage the whole face as input. The second approach is considering only some features or regions of the face, and the final method is the simultaneous usage of global and local facial traits. Furthermore, numerous datasets are geared towards the solution of specific face recognition problems, and these datasets are taken under laboratory conditions. However, there are some datasets that attempt to solve multiple problems and are taken under real-world conditions [9]. Fazilov, Mirzaev, and Mirzaeva [10] examined an algorithm to enhance the classification of objects in higher dimensions. The proposed algorithm formed a subset of correlated images, and then, a feature representation was elected to build elementary transformation models in the representative features' subspace. The algorithm pursues the augmentation of the accuracy of

recognition, learning time, and finally, object recognition time. The solution of the problem of low face recognition accuracy due to large samples and limited availability of training samples was solved by He, Wu, Sun, and Tan [11] when they proposed cross-modality images of heterogeneous face recognition (HRF). The study proposed the Wasserstein CNN framework that utilizes one network to project near infra-red and visual images to a Euclidean space. The proposed method is a modality invariant deep feature learning architecture for NIR-VIS HFR. The Wasserstein space that separates the NIR and VIS distribution is subsequently computed, and then, the correlation is levied on the connected layers to mitigate overfitting on small NIR datasets.

Similarly, Rahimzadeh, Arashloo, and Kittler [12] solved the optimization problem of MAP inference using the Markov random field (MRF) model by utilizing the processing power of the GPU's. The multiresolution analysis technique, incremental subgradient approach, and efficient message passing approach were used to obtain the maximum efficiency gain. Efficiency was enhanced by using the multiresolutional daisy features to attain invariance against occlusion and illumination. The proposed approach reduced the computational cost by 200% when compared to baseline methods. Likewise, Chan et al. [13] attempted the problem of training and adapting deep learning networks to different data and tasks. Chan et al. offered a method of passing images into a cascaded principal component analysis (PCA) filter for training PCANet. PCANet is subsequently used for feature extraction using the MultiPIE, extended YaleB, AR, FERET, and LWF databases. Moreover, PCANet is also a reference for reviewing advanced deep learning architectures containing a large number of image classifications. Also, Deng, Hu, Wu, and Guo [14] put forward the creation of a face image to mitigate varying illumination and pose, respectively, using only one frontal face image to develop an extended generic elastic model (GEM) and a multidepth model. Pose-aware metric learning (PAML) was learned by means of linear regression to synthesize each pose in their corresponding metric space, and it yielded an accuracy of 100%. Chen et al. [15] on the other hand proposed a residual-based deep face reconstruction neural network for the extraction of features from varying poses and illumination. This method changes illumination and pose images to frontal face images with an average lighting condition. By comparing the proposed triplet loss and the Euclidean loss, the experimentation proved better for the performance of the latter over the former. However, only one database was used for this study, and there were no results to compare the proposed method with.

Tu, Li, and Zhao [5] also solved the problem of illumination, pose, and expression by using a DL-Net and normalization network (N-Net). The DL-Net purges the illumination and then rebuilds the input image to an albedo image. The N-Net normalizes the albedo image and extracts features by supervised learning. The MultiPIE database establishes efficiency of the proposed method in augmenting face recognition accuracy under illumination, expression, and varying poses. The study concludes by stating that the extracted features can improve conventional feature

extraction methods. Zhang et al. [16] also proposed an emotion recognition model with better accuracy than the SOTA model. They extracted the facial expressions of seven different emotions. The extracted image is filtered through a combination of the Shannon entropy and multiscale feature extraction, and the result is classified using a fuzzy support vector machine (SVM). The study used the stratified cross-validation as the validation metric, and thus, an overall accuracy of 96.77% accuracy was achieved. Ghazi and Ekenel [17] improved the accuracy under occlusion, variations in illumination, and misalignment of facial features by using two deep CNN models, VGG-Face and Lightened pretrained on large datasets. These datasets were then used to extract facial features. They also used 5 databases to attempt a solution to the problem. The AR face dataset was used as the analytical tool for the effects of facial obstruction, CMU PIE, and the Extended Yale dataset B to analyze the variation in illumination. The color FERET database was used for impact analysis on view invariance, and last, the FRGC dataset is for evaluation of multiview catalogues. The authors then used the Facial Bounding Box Extension to scan the entire head and extract deep features, thus improving the results. They compared their results between the Facial Bounding Box Extension to other methods, and there was a significant improvement in results [18]. However, Zhang et al. optimized face landmark detection by taking advantage of supplementary data from the attributes of the features. The study proposed feature extraction using four convolutional layers. Each one of these layers produces several feature maps that are activated using rectified linear units. The layers are then coupled using max-pooling to produce a shared vector. The Multi-Attribute Facial Landmark (MAFL), AFLW, and Caltech Occluded Faces in the Wild (COFW) are subjects to mean error and failure rate validation. The study concluded that the auxiliary task is more efficient by learning the dynamic task coefficient, and this, in turn, makes the proposed method more robust to occluded faces and significant view invariance [19].

This approach encouraged Ding and Tao [20] to propound a homographic pose normalization approach which handles the loss of semantic correspondence, occlusion, and nonlinear facial texture wrapping in PIFR. The proposed method first projects a lattice of three-dimensional facial landmarks into a two-dimensional face for feature extraction. Second, an optimal warp is appraised using a homographic corrective texture deformation due to pose variation. This is performed around each landmark on the local patch. The restored occluded features are used for face recognition using established face descriptors [20]. However, Sharma and Patterh [21] proposed a technique, whereby the face is identified by the Viola–Jones algorithm. Then, the eyes, nose, and mouth are discovered by means of the proposed hybrid PCA. The features are subsequently mined using LBP for every part found. PCA is then applied to each feature extracted for recognition. The ORL face dataset was used with the recognition rate as the recognition metric. The study concluded that there is a higher recognition rate for the proposed hybrid PCA approach for varying facial expressions and pose when pitted with SOTA,

PCA + wavelet, CA, 2DPCA + DWT, and local binary pattern algorithms. They claimed that this approach can be extended to illumination, age, or partial occlusion problems. Interestingly, Duong, Luu, Quach, and Bui [22] presented an approach to deep appearance models (DAM) that accurately capture shape and texture variation under large variations using the deep Boltzmann machine (DBM). DAM replaced the active appearance model (AAM). This method begins by employing the use of DBM to ascertain the landmark distribution points on the face data, and then, the facial data are vectorized as a texture model. The two layers (shape and texture) are then interpreted by constructing and using a high-level layer. The LFW, Helen, and FG-NET databases were used for the experimentation. The RMSE values of the proposed method to the controlled method (bicubic and AAM) showed a significant improvement in the recognition rate [22].

Duan and Tan [23] also proposed a method of the low complexity method of learning pose-invariant features without the need for prior pose information. The proposed approach removes the pose from a face image and, by so doing, extracts local features. Self-similarity features are first generated from a face image when the distance that separates the features of different nonoverlapping blocks is evaluated. Then, the linear transformation is subtracted from the local features, and the transformation matrix is acquired by reducing the distance between pose variant features. This matrix is created while discriminative information across persons is retained. Nevertheless, Singh, Zaveri, and Raghuwanshi [24] have proposed a rough membership classifier (RMF) for the classification of pose images. Feature extraction was performed using log-Gabor, and SVD's are used for the reduction of redundant features. KNN classifier is finally applied on the reduced Gabor features. ORL, Georgian Face database, CMU PIE, Head Pose Image databases were used with similar performance metrics to Duan and Tan [23]. The study concluded that the proposed method is best suited for mug shots in law enforcement. Moreover, it improves the recognition of face images with occlusion, and the method is augmented using modeling techniques to gain improved results. However, the use of three methods for testing reduces the optimality of the proposed methods for substantial datasets with varying images. Nevertheless, Zhao, Li, and Liu [25] have proposed a MSA + PCA for pose-invariant FR. First, features are extracted using the affine-invariant multiscale autoconvolution (MSA) transformation. Furthermore, the decorrelation of these traits and the reduction of the MSA proportions are performed using principal component analysis. Finally, the principal components with the highest eigenvalues are classified using KNN. The experimentation points out how computationally expensive the proposed method is during the MSA feature extraction phase.

Abdalhamid and Jeberson [26] presented an abled pose-invariant FR system via artificial bee colony optimized K-nearest neighbor classifier (ABC-KNN). The method used video as input for conversion into frames. During the preprocessing of the converted images, the adaptive Lee filter (ALF) was applied for image enhancement by removing

noise. The Viola–Jones (VJ) algorithm is then used for face segmentation from the right eyes, nose, and mouth. Complete-LBP (CLBP), center symmetric local binary pattern (CS-LBP) features, Gabor features (GF), and patterns of gradient orientation magnitudes (POEM) descriptors are used for when quirks are extracted from the segmented image. ABC-KNN is applied as classification for the image. Recognition accuracy was the performance evaluation metric. Consequently, F. Zhang, Yu, Mao, Gou, and Zhan [27] propounded an approach for the PIFER framework based on feature learning using deep learning. The PCA-Net used frontal images that were not labeled during the learning process of the features. The latter are consequently used by CNN for feature mapping across the space separating the nonfrontal and frontal faces. The novel description generated by the maps is then used to describe nonfrontal faces to achieve a standard characteristic to describe arbitrary faces. The multiview robust features are then trained using a single classifier for varying poses. BU-3DFE Static FEW was used during the experimentation stage and recognition as a performance evaluation metric. After this technique has been contrasted with other techniques and frameworks, the proposed process seems to outperform SOTA techniques. Additionally, this method can be used to pose robust feature extraction when trained instead of training the model for different pose variations.

Finally, Sang, Li, and Zhao's [28] method for PIFR fuses texture and depth into a framework using joint Bayesian classifiers. The output is then identified using a similarity estimator between the input and the face database. However, there is a high computational cost for recognition of face images in large face databases. Furthermore, experimentation was extensive for various poses, and multiple methods were not compared to the current method.

3. Research Methodology

The research design for this study includes image pre-processing, feature extraction with PCA, the optimization of these features using PSO, ABC, and GA, and finally the classification of objects using KNN, SVM, and EUD. The datasets for the study are YaleB and AT&T popularly known as ORL. These datasets were selected with the justification that they have well-defined challenges necessary for validating the facial recognition algorithm. Subsequent sections explain in detail the major parts of the study design.

3.1. Feature Extraction. This component of the design acquires relevant biometric descriptors from a given image. In the process, high volume of data is obtained making it necessary to select only high contributing descriptors. Several techniques exist for this task; however, PCA is adopted for this study due to its popularity and efficiency in this domain [29].

3.1.1. Principal Component Analysis. The primary goal of principal component analysis for facial recognition is the transformation of higher dimensional data into a lower

feature subspace known as the eigenface. This eigenspace represents the locus of the covariance matrix of the feature landmarks. Despite its usefulness, they are computationally expensive given a higher dimensional data. This necessitates the adoption of an alternate algorithm with similar properties and structures [30] as PCA but relatively inexpensive known as singular value decomposition (SVD). Taking a matrix X with dimension $n \times m$, a PCA can be defined as the Eigen decomposition of the covariance matrix $X^T X$. This yields an eigenvalue λ with its corresponding eigenvectors W . These eigenvectors are used as the transformation operator on X to obtain a new matrix T with the same dimension as X as shown in

$$T = XW . \quad (1)$$

Equation (1) is with the assumption that all components (i.e., columns) in W are principal. However, in practice, some of these components are expected to be redundant; hence, W is ordered by λ . With the ordered W , truncations can be performed using the first r components for analysis. By implication, we have W_r being an m by r matrix giving us the new transformed matrix T_r shown in

$$T_r = XW_r . \quad (2)$$

As stated earlier, operations of PCA are expensive, and SVD with properties mathematically identical to PCA is preferred for implementation. Equation (3) shows the SVD of X .

$$X = \mu \Sigma V^* , \quad (3)$$

where μ is the left singular vector, V^* is the conjugate transpose of the right singular vector, and Σ contains the singular values on its diagonals. Computing the eigenvalue decomposition for $X^T X$ with equation (3) to obtain $\mu \sigma \mu^T$, it becomes obvious that W is identical to V , while the ordered singular values ($\sigma_1 \sigma_2 \sigma_3 \dots$) are proportional to λ . Again, with the property that μ and V are unitary matrices, we have

$$\mu^* \mu = I , \quad (4)$$

$$V^* V = I , \quad (5)$$

where I is the identity matrix. From equations (1) and (3) and noting that W is identical to V , we have

$$T = XV = \mu \Sigma V^* V = \mu \Sigma I , \quad (6)$$

$$T = \mu \Sigma . \quad (7)$$

These equations further justify why SVD is computationally inexpensive compared to PCA which computes the covariance $X^T X$. Taking the principal components of equation (7), we have

$$T_r = \mu_r \Sigma_r . \quad (8)$$

Finally, since the requirement is W and not the Eigen decomposition of $X^T X$, SVD can be used to efficiently compute W .

3.2. Feature Optimization. The section of the study describes the swarm intelligence algorithms used for the feature optimization. Among these methods are artificial bee colony, genetic algorithm, and particle swarm optimization.

3.2.1. Artificial Bee Colony. The artificial bee colony (ABC) is one of the swarm-based algorithms designed with the foraging actions of the honeybees. The four components of the behavioral model of ABC are mainly the food source, scouting bees, onlooker bees, and employed bees. The food source denotes a possible solution to the clustering problem as the scout bee carries out a global search. This search is performed stochastically, while the onlooker and employed bee search for adjacent solutions. The employed bees subsequently evaluate the precision of the solution from the previously stored solutions in memory. This information is successively passed on to onlooker bees in the dance area. This ensures that the best food source is chosen, and the stagnated food sources within an already set cycle are abandoned and replaced with new sources. This process is repeated until there is a convergence to obtain the optimal solution. Mathematically, we have the following steps.

Step one: randomly initialize solutions x_i for $i = \{1, 2, \dots, FS\}$, where i represents each food source, and FS represents the total food source. Furthermore, initialize onlookers and employ bees using a random function generator in

$$x_{ij} = x_j \text{rand}(0, 1)(x_{\max j} - x_j), \quad (9)$$

where $x_{ij} = [x_{i1}, x_{i2}, \dots, x_{iD}]$ is a vector of length D with $x_{\max j}$ and x_j denoting the maximum and minimum values of the j^{th} dimension.

Step two: iteratively new solutions are found by each employed bee using

$$v_{ij} = x_j \varphi_{ij}(x_{ij} - x_j), \quad (10)$$

where $v_{ij} = [v_{i1}, v_{i2}, \dots, v_{iD}]$ signifies the new solutions within the local range of $x_{ij} = [x_{i1}, x_{i2}, \dots, x_{iD}]$ and $\varphi_{ij} \in (-1, 1)$. The sum of the Euclidean distance between the sample points and their cluster midpoints is known to be inversely proportional to the fitness value of all candidate sources. In the selection of the sources, a greedy algorithm is employed by comparing the fitness values of old and new positions.

Step three: probability p_i of the solution x_i is computed using

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{fs} \text{fit}_n}, \quad (11)$$

where fit_i is the fitness value of x_i . Onlooker bees use this probability to select new x_i values by searching for the local optimums while following step two to calculate the fitness value.

Step four: if onlooker and employed bees are unable to identify new and better candidate solution through the local search after some predefined iterations, the solution x_i is discarded and substituted with scout bees' new solution. These scout bees then use random global selection to search for new solutions.

Step five: step two to four is repeated until the defined stopping criterion is met returning the optimal output

3.2.2. Genetic Algorithm. Genetic algorithm (GA) on the other hand is based on genetics and the theory of natural selection. It is a stochastic algorithm which finds the best solution by effectively finding the global optimum in a larger space. A nonnegative fitness value is obtained using the fitness function. This value is used to summarize how close the optimal solution is to the global best (Mahmud, Haque, Zuhori, and Pal, 2014). A GA begins by generating random numbers (called chromosomes) with population size n . Each chromosome has its fitness value computed, and the stopping criterion is checked. The GA operators such as selection, crossover, and mutation to drive the chromosomes toward convergence are explained further.

Selection. This operator creates offspring from an existing population by using a process comparable to natural selection in biological lifeforms. Selection once more accentuates on the better performance of individuals in the population. This helps with the expectancy of their offspring having the likelihood of carrying on the genetic information to a successive generation. Consequently, the convergence is impacted greatly by the magnitude of the selection process. Hence, the selection criteria should prevent premature convergence by maintaining population diversity and balance with the crossover and mutation operations.

Crossover. The crossover operator mixes information between two parents in a manner matching sexual reproduction. The objective of the crossover procedure is to give "birth" to an improved offspring. This is achieved by exploring different portions of the search space.

Mutation. Mutation procedure changes the values of the randomly selected bit within each string, thereby preventing the GA from being stuck at the local minimum through the scattering of genetic data, hence maintaining the variation in the population. This process is repeated until the optimal solution is achieved or the predetermined number of generations elapses.

3.2.3. Particle Swarm Optimization. Particle swarm optimization (PSO) is also an optimization algorithm influenced by biology. It was derived by observing the collective behavior and swarming of a flock of birds and fish schools [30]. The algorithm comprises of solutions known as population, with each having a series of parameters which represent a coordinate in a space with multiple dimensions. Furthermore, a collection of these particles becomes a population

with the particles probing the search space to find the optimal solution. Each particle tracks its former optimal solution in memory and then labels these solutions as the personal best and global best. The locus of the i^{th} particle is then defined in the D-multidimensional space as

$$x_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}], \quad (12)$$

and the population of the swarm as

$$X = [x_1, x_2, x_3, \dots, x_N]. \quad (13)$$

The particles then iteratively update their respective positions in the parameter space when searching for the optimal solution using

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (14)$$

where v_i is the velocity components of the i^{th} particle along the D-dimensions with t and $t+1$, indicating a dual consecutive run of the process. Velocity of the i^{th} particle is defined in equation (15) with three terms: the first is inertia which prevents the particles from drastically changing direction, the second term describes the ability of particles returning to the previously known best position, and the last term describes the particles moving (swarm) closer to the best position:

$$v_i(t+1) = v_i(t) + c_1(p_i - x_i(t))R_1 + c_2(g - x_i(t))R_2, \quad (15)$$

where p_i is the personal best of the particle, g is the global best, and c_1 and c_2 , in the range of $0 \leq c_1, c_2 \leq 4$, are the cognitive and social coefficients respectively. Finally, R_1 and R_2 are the two diagonal matrices randomly generated from a uniform distribution in $[0,1]$. This ensures that the social and cognitive components have a random effect on the velocity update in equation (15). Since the particles are derived from the convergence of the personal and global best solutions, the stochastic weight of the two accelerating terms and the trajectories are semirandom. This requires that equations (14) and (15) are iterated until a stopping criterion is met. Algorithmically, we have the following pseudocode.

3.3. PSO Algorithm.

(1) N particle initialization

- (a) Initialize the position $x_i(0) \forall i \in 1: N$
- (b) Initialize the particles best position to its position $P_i(0) = x_i(0)$
- (c) Calculate the fitness of each particle, and if $f(x_j(0)) \geq f(x_i(0)) \forall i \neq j$, initialize the global best as $g = x_j(0)$

(2) Repeat until condition is met

- (a) Update the particle velocity in accordance with equation (15)

$$v_i(t+1) = v_i(t) + c_1(p_i - x_i(t))R_1 + c_2(g - x_i(t))R_2. \quad (16)$$

- (b) Update the particle position using equation (14)

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (17)$$

- (c) Evaluate the fitness of the particle $f(x_i(t+1)) \geq f(p_i)$
- (d) If $f(x_i(t+1)) \geq f(p_i)$, update personal best: $p_i = (x_i(t+1))$
- (e) If $f(x_i(t+1)) \geq f(g)$, update global best: $g = (x_i(t+1))$.
- (3) Assign the best solution to g at the end of the iterative process.

3.4. Classification. After the optimization of the extracted feature vectors, classification models are built to address the face recognition challenges. There are myriads of predefined models for this task given the feature set. Among these are SVM, KNN, K-means, Euclidean distance, VGGNet, and CNN. Other pretrained face classifiers such as the VGG-Face also exists which estimate the similarity between the face image of a subject and relevant features selected from the face images in the database. In this study, the Euclidean distance (EUD), K-nearest Neighbor (KNN), and the support vector machine (SVM) were used.

4. Implementation of Methods

4.1. Implementation Pipeline. The implementation pipeline for this study is shown in Figure 1. From the figure, every image undergoes a series of preprocessing and subsequent feature selection and finally features optimization. These optimized features are trained for feature matching.

4.2. Environmental Setup. The face recognition system implemented in this study was developed, trained, and tested using Matlab R2018b on an HP desktop processor Intel® Core™ i7-770T CPU @ 2.90 GHz, Linux Ubuntu 20.04 LTS operating system.

4.3. Image Preprocessing. The first step taken in image analysis is the preprocessing of the image for undesirable noise. These components are detrimental to the examination of the image and thus are removed via preprocessing. All images with dimensions more than 96-by-84 pixels are downsampled. This is followed by the conversion of all colored images to grayscale. The outputs of the images are separated into training and test sets. Eighty percent of the images are considered as training sets with 20 percent as the test set. This preprocessing is implemented so that the complexity will be reduced and the computational time improved.

4.4. Feature Extraction. This section further illuminates on the feature extraction approach used in this study. Among the objectives of this study is the implementation of an offline facial recognition system with an improved and robust feature extraction method using optimization

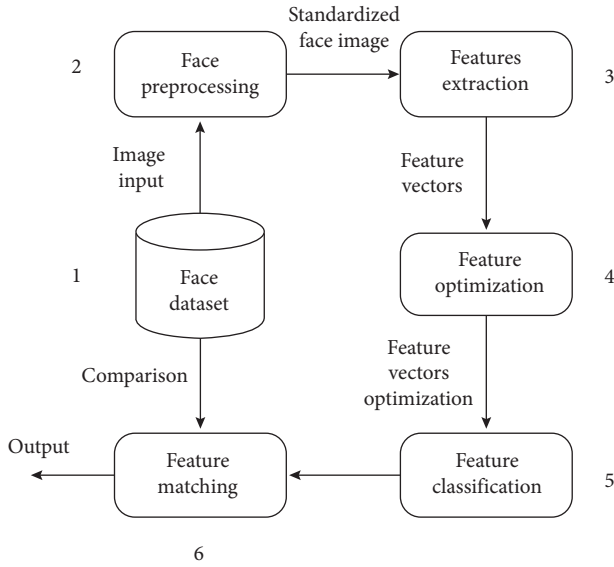


FIGURE 1: Implementation pipeline.

techniques. This method will be tested using the AT&T and YaleB face datasets as they contain faces with varying illumination, different poses, occluded faces, dissimilar expressions, or a combination of them. The mean of the features is computed, and the feature of the first principal component of each image is selected. The mean face for AT&T and YaleB datasets is shown in Figures 2 and 3, respectively.

4.5. Dimensionality Reduction and Feature Selection. Given the computed mean face of the training data, the binary singleton expansion function is applied as an element-wise operator. The resultant image is decomposed with the single value decomposition function to reduce the coefficient used to characterize the image. The cumulative sum of the square of the diagonal matrix is computed to produce the principal component with the first k eigenvalue of the component selected. The eigenvectors are then normalized into eigenfaces. The sample output of this process on the AT&T and YaleB datasets is shown in Figures 4 and 5, respectively.

Once more, the binary singleton expansion function is used to transform the test data by using the mean face. These transformed train and test data are then optimized for better classification results.

5. Results and Discussion

This section describes in detail the results of the experiment and the analysis of the results. Moreover, comparisons between other optimization methods using the same database and three different classifiers will be discussed.

5.1. Numerical Results. Generally, in recording the performance of a facial recognition model, statistical metrics such as accuracy, recall, precision, F-measure, and among others

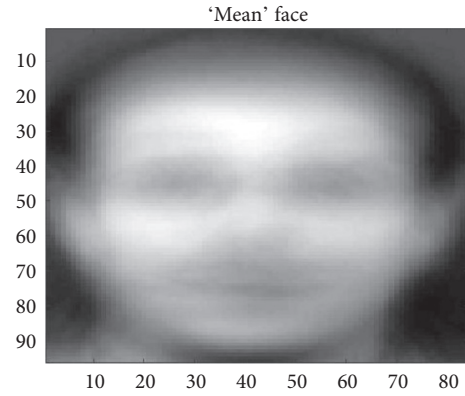


FIGURE 2: Mean face for the AT&T dataset.

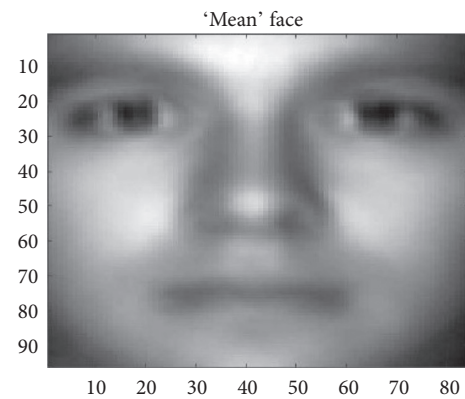


FIGURE 3: Mean face the YaleB dataset.

are used. For an efficient evaluation and a valid comparison with the existing study, the accuracy metric is selected. The recognition accuracy is computed for all the classification methods as applied on different datasets with varying optimization methods. Tables 1–7 show the average, maximum, and minimum recognition accuracies for the datasets with different classification methods. This experiment was conducted with a thousand five hundred (1500) iterations with/without considering the optimization of the extracted features.

5.2. Discussion. From the result shown in Section 4.1, it is observed that the model's performance on the AT&T dataset is fairly low in general. This could be attributed to the occlusion, varying pose, and expression exhibited in the face images making it naturally difficult to model. On the contrary, the model's performance was relatively good as it contains only images with varying illumination. From Table 1, it can be seen that the accuracy is highest for KNN and SVM at 100% each for the YaleB dataset. Nevertheless, optimizing the features with GA saw a significant decrease of the KNN classifier to 70% with a 9.8% reduction using the Euclidean distance method as shown in Table 3. There is no loss as shown in Tables 2 and 4 for KNN and SVM when ABC and PSO optimization is performed. However, 4% and 5% reduction for PSO and ABC, respectively, was noted

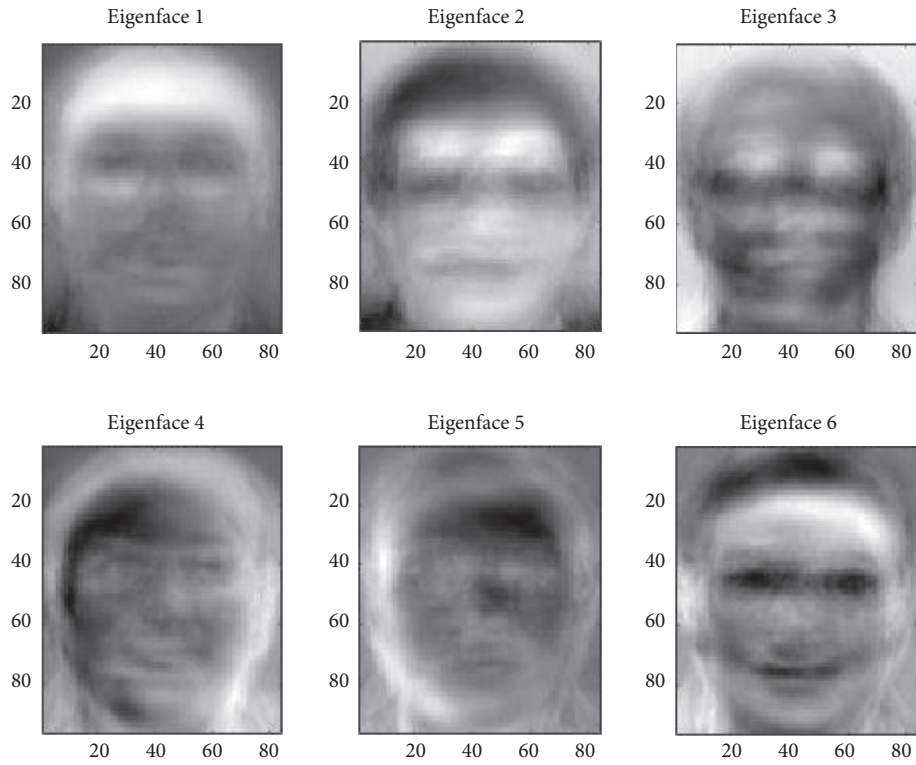


FIGURE 4: First six eigenfaces of AT&T.

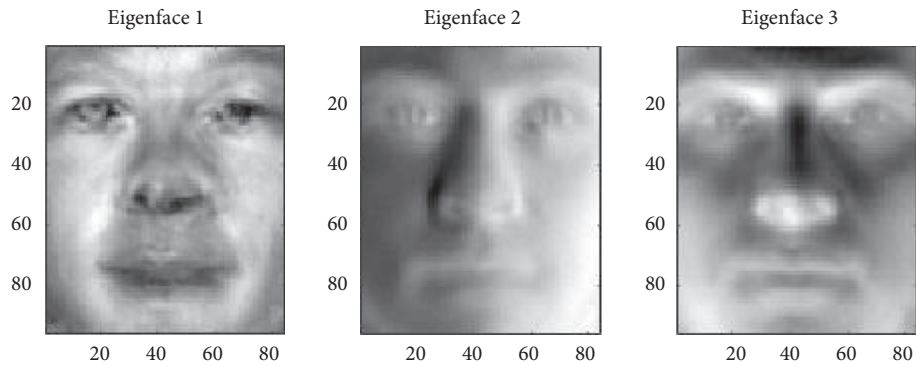


FIGURE 5: First three eigenfaces of YaleB.

TABLE 1: Recognition accuracy without the use of the optimization algorithm.

Default recognition accuracies			
EUD	KNN	SVM	Dataset
29.94	77.84	82.05	AT&T
82.63	100	100	YaleB

TABLE 2: Recognition accuracies for the YaleB database with the PSO algorithm.

YaleB-particle swarm optimization (PSO)			
EUD	KNN	SVM	
87.92	100	91.72	Average
100	100	100	Maximum
10.53	100	0	Minimum

TABLE 3: Recognition accuracies for the YaleB database with the GA algorithm.

YaleB–genetic algorithm (GA)			
EUD	KNN	SVM	
70.04	70.04	99.15	Average
100	100	100	Maximum
10.53	10.523	91.23	Minimum

TABLE 4: Recognition accuracies for the YaleB database with the ABC algorithm.

YaleB–artificial bee colony (ABC)			
EUD	KNN	SVM	
74.71	100	99.60	Average
100	100	100	Maximum
17.54	100	95.61	Minimum

TABLE 5: Recognition accuracies for the AT&T database with the PSO algorithm.

AT&T–particle swarm optimization (PSO)			
EUD	KNN	SVM	
28.46	80.46	59.85	Average
36.25	99.25	78.75	Maximum
18.75	67.5	5	Minimum

TABLE 6: Recognition accuracies for the AT&T database with the GA algorithm.

AT&T–genetic algorithm (GA)			
EUD	KNN	SVM	
28.78	47.05	54.86	Average
40	87.5	86.25	Maximum
17.5	7.5	15	Minimum

TABLE 7: Recognition accuracies for the AT&T database with the PSO algorithm.

AT&T–artificial bee colony (ABC)			
EUD	KNN	SVM	
28.50	66.78	35.55	Average
43.75	93.75	72.5	Maximum
17.5	8.75	8.75	Minimum

when the EUD classifier was used. Consequently, there is a large difference in recognition accuracy using the AT&T database. Without the use of the optimization method, the AT&T database's recognition plummeted to 29.94, 77.84, and 82.05 for EUD, KNN, and SVM, respectively, as shown in Table 1. However, using the PSO optimization technique saw an improvement in the average recognition accuracy to 80.46% for KNN. There is a significant degradation when using the SVM classifier with an average recognition accuracy of 59.85%. Again, EUD saw 28.46% average recognition accuracy for PSO as shown in Table 5. Conversely, the average recognition accuracy for GA and ABC reduced to 47.05 and 66.78, respectively, when using KNN and 54.86% and 35.55% when using SVM as can be separately observed

in Tables 6 and 7. The order of experimentation is given as follows.

PCA + EUD
 PCA + KNN
 PCA + SVM
 PCA + PSO + ED
 PCA + PSO + KNN
 PCA + PSO + SVM
 PCA + GA + ED
 PCA + GA + KNN
 PCA + GA + SVM

PCA + ABC + ED
 PCA + ABC + KNN
 PCA + ABC + SVM

By observation, the PCA + PSO + EUD has 15.19% of the recognition accuracy below 79.83%, which is the recognition accuracy of PCA + EUD for the YaleB dataset. This indicates that PSO optimizes the features well with 84.81% of the experiments producing better results. In addition, there is no change in the recognition accuracy of KNN. This points with PSO not reducing the results achieved when the optimization technique is performed. Conversely, SVM saw less than 1% of the recognition accuracies below 90%. This demonstrates that over 99% of the results for the PCA + PSO + SVM have recognition accuracy above 90% with 95% of the recognition accuracy at 100%. Therefore, the 5% reduction in PCA + SVM can be considered insignificant. As a final point, PSO optimizes well for EUD and SVM. Similarly, PCA + GA + EUD has over 71% of the results above that of PCA + EUD. KNN and SVM, however, have 60% and 100% of the results greater than that of PCA + KNN and PCA + SVM, respectively. Yet still, PCA + GA + KNN shows significant decay of results from its default 100%. SVM, on the other hand, displays a negligible reduction in average recognition accuracy. With this, SVM seems to produce better results than both KNN and EUD with respect to the use of the GA optimization algorithm.

In like manner, the YaleB dataset results for PCA + ABC + EUD give rise to 28% of the data above the default 79.83% of PCA + EUD. However, the ABC optimized recognition for KNN and SVM revealed no significant loss of results with an average recognition accuracy of 100% and 99.6% for KNN and SVM, respectively. It can be established that the result optimized by ABC and classified using KNN are appropriate for the YaleB dataset, and ABC optimizes well for KNN and SVM on the said dataset. Table 8 shows the first 20 results of the total experiments for PSO, ABC, and GA optimization algorithms implemented on the YaleB dataset using EUD, KNN, and SVM classifiers. Nevertheless, the substantial reduction of the results observed when the AT&T dataset is used stems from the increase in parameters for recognition. The AT&T dataset contains images that are occluded, and it also has varying poses and expressions. PCA + PSO + EUD for the AT&T face dataset produced results that are on average below PCA + EUD for the database. 51% of the 1500 results obtained were lower than the default 29.94% for PCA + EUD. The overall average recognition of PCA + PSO + EUD for the AT&T database, however, was 28.46% as shown in Table 5. It is perceived that the deterioration of average recognition is offset by the larger values of the other recognitions. 48% of the result above the default 29.94% is not insignificant, yet it is a small percentage for consideration. KNN on the other hand has 31% of the results above the 77.84% default recognition. Still, the average recognition accuracy achieved was 3% higher than the default. Thus, 80.46% average recognition accuracy for KNN with PSO-optimized features (PCA + PSO + KNN) is the best combination for the AT&T database since none of the results for SVM was above its 82.05% baseline recognition.

Again, PCA + GA + EUD indicates 28.78% average recognition accuracy. This is similar to the average results got by all 3 optimization methods using EUD as the classifier. However, GA and ABC achieved 47.05% and 66.78% average recognition accuracy for the KNN classifier, respectively. This illustrates an atrophy of the result from 77.84% to 47.05% for GA and 66.78% for ABC. GA suffers 30% degradation, while ABC saw an 11% reduction in average recognition accuracy. Moreover, the average recognition accuracy for both GA and ABC for the SVM classifier plummeted further than that of KNN. A 27% reduction in average recognition accuracy using the SVM classifier for GA supersedes that of ABC, which has 46.5% reduction. Thus, it concludes that GA and ABC using SVM as the classifier is not suitable for this approach. The first 20 results are shown in Table 9.

Again, the linear kernel was used for the SVM classifier when the experiment was performed. This kernel has the propensity of improving computational time compared to other SVM kernels, and it is suitable for high dimensional data [31]. However, the linear kernel in this experiment appears to have sacrificed the accuracy for computational time. Thus, the kernel chosen does not produce good results. Other kernels such as the polynomial, Gaussian, radial basis function (RBF), or ANOVA could be used for SVM in future research, and the result is compared to the proposed method. Similarly, Table 1 indicates that SVM is a better classifier when the linear kernel is used and when no optimization algorithms are utilized. Thus, both AT&T and YaleB datasets produce the best results for SVM. Now, comparing Tables 2–4, it is perceived that a perfect recognition accuracy of 100% for the maximum of all meta-heuristic algorithms and classifiers is achieved. This indicated that all optimization methods can be used for the YaleB database regardless of the classifier. Conversely, the maximum recognition accuracy for the algorithms used for augmentation gave the impression that the KNN classifier was better. This means that PSO + KNN, ABC + KNN, and GA + KNN have better recognition accuracy than their SVM counterparts. This shows that the optimization algorithms have degraded the results produced by the SVM classifier. Nonetheless, GA's maximum recognition was better than that of the default SVM (PCA + SVM). Therefore, GA should be preferred when an SVM classifier with a linear kernel is chosen. Furthermore, the algorithms improved the highest recognition accuracy achieved by the EUD classifier only. With this, PSO is selected as the ideal optimization algorithm for the YaleB and AT&T datasets. Juxtaposing the proposed method to other approaches, it is shown in Table 10 that the offered approach is effective than other SOTA methods. The culmination of this research presented the proposed optimization method and classifier, given their respective datasets in Table 11.

Finally, Table 12 shows the time taken for each experiment carried out. It is seen that PSO has the lowest average time for the experiment with 1.594s, 1.592s, and 55.46s for EUD, KNN, and SVM, respectively. PSO + SVM saw the highest computational cost with 55.46s for all experimentation. However, it required less than 2 seconds for

TABLE 8: YaleB experiment results for PSO, GA, and ABC using EUD, KNN, and SVM classifiers.

YaleB-PSO			YaleB-GA			YaleB-ABC		
EUD	KNN	SVM	EUD	KNN	SVM	EUD	KNN	SVM
100	100	100	60.526	60.526	99.123	36.842	100	100
100	100	100	99.123	99.123	100	35.088	100	100
71.93	100	100	51.754	51.754	100	28.07	100	100
100	100	100	100	100	98.246	71.93	100	99.123
95.614	100	100	41.228	41.228	98.246	86.842	100	100
99.123	100	100	40.351	40.351	100	100	100	100
90.351	100	100	94.737	94.737	98.246	71.053	100	99.123
99.123	100	100	99.123	99.123	99.123	47.368	100	100
100	100	100	35.088	35.088	99.123	53.509	100	100
100	100	100	99.123	99.123	100	54.386	100	100
85.965	100	100	100	100	100	100	100	100
67.544	100	98.246	100	100	100	57.895	100	100
82.456	100	100	46.491	46.491	100	91.228	100	100
87.719	100	100	97.368	97.368	99.123	30.702	100	98.246
88.596	100	100	41.228	41.228	100	92.982	100	98.246
83.333	100	98.246	28.07	28.07	96.491	100	100	100
93.86	100	100	100	100	98.246	50	100	100
100	100	100	50	50	97.368	72.807	100	100
70.175	100	99.123	100	100	100	27.193	100	100

TABLE 9: AT&T experiment results for PSO, GA, and ABC using EUD, KNN, and SVM classifiers.

AT&T-PSO			AT&T-GA			AT&T-ABC		
EUD	KNN	SVM	EUD	KNN	SVM	EUD	KNN	SVM
30	82.5	70	28.75	53.75	51.25	27.5	71.25	53.75
26.25	80	61.25	37.5	31.25	37.5	31.25	55	33.75
28.75	76.25	40	32.5	78.75	76.25	28.75	38.75	20
32.5	76.25	75	36.25	20	30	23.75	76.25	41.25
27.5	77.5	62.5	23.75	51.25	70	31.25	10	8.75
27.5	80	75	27.5	31.25	35	26.25	71.25	27.5
30	86.25	55	28.75	35	38.75	31.25	55	27.5
33.75	87.5	73.75	27.5	63.75	62.5	30	42.5	23.75
23.75	71.25	66.25	33.75	35	40	27.5	77.5	47.5
23.75	76.25	68.75	30	56.25	71.25	28.75	42.5	16.25
31.25	72.5	62.5	36.25	28.75	35	26.25	50	22.5
26.25	73.75	68.75	27.5	46.25	56.25	23.75	76.25	37.5
32.5	86.25	7.5	27.5	46.25	60	30	77.5	52.5
30	83.75	63.75	28.75	11.25	16.25	25	45	31.25
30	85	65	27.5	53.75	50	27.5	67.5	37.5
22.5	83.75	50	30	28.75	33.75	30	45	40
31.25	78.75	63.75	26.25	67.5	65	26.25	70	33.75
28.75	68.75	56.25	35	58.75	62.5	28.75	86.25	56.25
31.25	87.5	70	30	57.5	60	23.75	45	10

TABLE 10: Recognition accuracy of other methods on the YaleB and AT&T datasets.

Author	Method	Recognition accuracy (%)	Database
[32]	Generalized low-rank approximation of matrices (GLRAM)	82.18	YaleB
[33]	FDDL	96.2	YaleB
[34]	Local nonlinear multilayer contrast patterns (LNL MCP)	97.50	YaleB
[35]	Discriminative sparse representation via l_2 regularization	82.61	YaleB
[32]	GLRAM	97.25	AT&T
[33]	Fisher discriminative dictionary learning (FDDL)	96.7	AT&T
[31]	PSO-KNN	98.75	AT&T
[31]	PCA-LDA fusion algorithm	98.00	AT&T
[35]	Discriminative sparse representation via l_2 regularization	95.00	AT&T

TABLE 11: Proposed classification and optimization techniques for both datasets.

Proposed selection	AT&T	YaleB
Optimization technique	Particle swarm optimization (PSO)	Particle swarm optimization (PSO)
Classification method	K-nearest neighbor	K-nearest neighbor

TABLE 12: Average time taken for experiments.

Time in seconds for experimentation	Particle swarm optimization	Artificial bee colony	Genetic algorithm
Euclidean distance	1.594	2.104	1.648
K-nearest neighbor	1.592	2.115	1.646
Support vector machine	55.46	4.871	4.445

PSO + EDU and PSO + KNN trials. Subsequently, the ABC and GA meta-heuristic algorithms produced a similar result to PSO, but PSO is computationally less expensive than both.

6. Conclusion

This study looks at how to augment PCA feature with the selected optimization method to improve the accuracy of face recognition models. The proposed implementation shows that the choice of PSO as an optimization method works well in an unconstrained environment of the real world, since pose, occlusion, and expression are among the dominate face recognition problems found in the unconstrained environments. The default recognition accuracy of the YaleB showed 100% accuracy for both SVM and KNN classifiers. However, the ORL database did not attain perfect recognition due to the inherent nature of the dataset. Nonetheless, the use of optimization algorithms on the selected features saw an increase in recognition accuracy from 82.63% to a maximum of 100% for EUD. This indicates that all three evolutionary algorithms can be used to improve the accuracy of results. However, due to the ORL database catering for 3 parameters, the maximum recognition did not reach 100% but 99.25% which is promising using the PSO algorithm and KNN classifier. Last, the PCA + PSO + KNN approach is chosen for this study due to its ability to handle the increase in parameters, and it also outperforms other SOTA algorithms. These parametric increases move the recognition closer to real-world human face recognition. Moving forward, this study can be extended by looking at other recent swarm intelligent optimization models used in other fields with the property of it being less expensive. Other private datasets with more stricter challenges could be used to further validate this model. This remains a limitation to this study.

Data Availability

The secondary data source used to support the findings of this study are available from the AT&T database (<https://www.kaggle.com/kasikrit/att-database-of-faces>) and YaleB database (<https://github.com/Suchetaaa/CS663-Assignments/tree/0426d951d0212ed3dd831377a0df11551670ab87/Assignment-4/1/CroppedYale>).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Li and W. Deng, "Deep facial expression recognition: a survey," 2018, <http://arxiv.org/abs/1804.08348>.
- [2] R. D. Labati, A. Genovese, E. Muñoz, V. Piuri, F. Scotti, and G. Sforza, "Biometric recognition in automated border control," *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–39, 2016.
- [3] F. Zhang, Y. Yu, Q. Mao, J. Gou, and Y. Zhan, "Pose-robust feature learning for facial expression recognition," *Frontiers of Computer Science*, vol. 10, no. 5, pp. 832–844, 2016.
- [4] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: past, present and future," *Computer Vision and Image Understanding*, vol. 138, pp. 1–24, 2015.
- [5] H. Tu, K. Li, and Q. Zhao, "Robust Face Recognition with Assistance of Pose and expression Normalized albedo images," *ACM International Conference Proceeding Series*, vol. 93, 2019.
- [6] X. Chen, X. Lan, G. Liang, J. Liu, and N. Zheng, "Pose-and-illumination-invariant face representation via a triplet-loss trained deep reconstruction model," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22043–22058, 2017.
- [7] C. Ding, C. Xu, and D. Tao, "Multi-task pose-invariant face recognition," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 24, no. 3, pp. 980–993, 2015.
- [8] C. Ding and D. Tao, "A comprehensive survey on pose-invariant face recognition," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 3, 2016.
- [9] M. Chihaoui, A. Elkefi, W. Bellil, and C. B. Amar, "A survey of 2D face recognition techniques," *Computers*, vol. 5, no. 4, pp. 41–68, 2016.
- [10] S. K. Fazilov, N. M. Mirzaev, and G. R. Mirzaeva, "Modified recognition algorithms based on the construction of models of elementary transformations," *Procedia Computer Science*, vol. 150, pp. 671–678, 2019.
- [11] R. He, X. Wu, Z. Sun, and T. Tan, "Wasserstein CNN: learning invariant features for NIR-VIS face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1761–1773, 2019.
- [12] S. Rahimzadeh Arashloo and J. Kittler, "Fast pose invariant face recognition using super coupled multiresolution Markov Random Fields on a GPU," *Pattern Recognition Letters*, vol. 48, pp. 49–59, 2014.
- [13] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: a simple deep learning baseline for image classification?" *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.
- [14] W. Deng, J. Hu, Z. Wu, and J. Guo, "From one to many: pose-Aware Metric Learning for single-sample face recognition," *Pattern Recognition*, vol. 77, pp. 426–437, 2018.

- [15] Z. Chen, W. Shen, and Y. Zeng, "Sparse representation for pose invariant face recognition," *International Journal for Engineering Modelling*, vol. 30, no. 1–4, pp. 37–47, 2017.
- [16] T. Zhang, W. Zheng, Z. Cui, Y. Zong, J. Yan, and K. Yan, "A deep neural network-driven feature learning method for multi-view facial expression recognition," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2528–2536, 2016.
- [17] M. M. Ghazi and H. K. Ekenel, "A comprehensive analysis of deep learning based representation for face recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 102–109, Las Vegas, Nevada, USA, July 2016.
- [18] M. M. Ghazi and H. K. Ekenel, "Automatic emotion recognition in the wild using an ensemble of static and dynamic representations," in *Proceedings of the ICMI 2016 - Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pp. 514–521, Tokyo, Japan, November 2016.
- [19] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 918–930, 2016d.
- [20] C. Ding and D. Tao, "Pose-invariant face recognition with homography-based normalization," *Pattern Recognition*, vol. 66, pp. 144–152, 2017.
- [21] R. Sharma and M. S. Patterh, "A new hybrid approach using PCA for pose invariant face recognition," *Wireless Personal Communications*, vol. 85, no. 3, pp. 1561–1571, 2015.
- [22] C. N. Duong, K. Luu, K. G. Quach, and T. D. Bui, "Deep appearance models: a deep Boltzmann machine approach for face modeling," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 437–455, 2019.
- [23] X. Duan and Z.-H. Tan, "A spatial self-similarity based feature learning method for face recognition under varying poses," *Pattern Recognition Letters*, vol. 111, pp. 109–116, 2018.
- [24] K. Singh, M. Zaveri, and M. Raghuwanshi, "Rough set based pose invariant face recognition with mug shot images," *Journal of Intelligent & Fuzzy Systems*, vol. 26, no. 2, pp. 523–539, 2014.
- [25] Y. Zhao, L. Li, and Z. Liu, "A novel algorithm using affine-invariant features for pose-variant face recognition," *Computers & Electrical Engineering*, vol. 46, pp. 217–230, 2015.
- [26] K. H. Abdalhamid and W. Jeberson, "Pose-invariant face recognition by means of artificial bee colony optimized knn classifier," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 8, pp. 525–539, 2019.
- [27] Y.-D. Zhang, Z.-J. Yang, H.-M. Lu et al., "Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation," *IEEE Access*, vol. 4, pp. 8375–8385, 2016.
- [28] G. Sang, J. Li, and Q. Zhao, "Pose-Invariant Face Recognition via RGB-D Images," 2016.
- [29] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "CloudID: trustworthy cloud-based and cross-enterprise biometric identification," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7905–7916, 2015.
- [30] M. Tiwari and A. K. Shukla, "An Implementation of FACE recognition system (FARS) using PCA and PSO based techniques," *State of the Art in Face Recognition*, vol. 211007, no. 6, pp. 225–229, 2016.
- [31] K. Sasirekha and K. Thangavel, "Optimization of K-nearest neighbor using particle swarm optimization for face recognition," *Neural Computing and Applications*, vol. 31, no. 11, pp. 7935–7944, 2019.
- [32] S. Ahmadi and M. Rezghi, "Generalized low-rank approximation of matrices based on multiple transformation pairs," *Pattern Recognition*, vol. 108, 2020.
- [33] B. B. Benuwa, B. Ghansah, and E. K. Ansah, "Kernel based locality – sensitive discriminative sparse representation for face recognition," *Scientific African*, vol. 7, Article ID e00249, 2020.
- [34] L. Zhou, W. Li, Y. Du, B. Lei, and S. Liang, "Adaptive illumination-invariant face recognition via local nonlinear multi-layer contrast feature," *Journal of Visual Communication and Image Representation*, vol. 64, Article ID 102641, 2019.
- [35] Y. Xu, Z. Zhong, J. Yang, J. You, and D. Zhang, "A new discriminative sparse representation method for robust face recognition via l_{21} regularization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2233–2242, 2017.