Received 17-01-2026; Accepted 02-04-2026; Published 27-04-2026

Face Image Recognition and Normalization Using Artificial Intelligence

Islambek Saymanov islambeksaymanov@gmail.com
National University of Uzbekistan

Muhammadiyev Abbos muhammadiyevabbos26@gmail.com
National University of Uzbekistan

Maxmatqulov Tuychi tuychimaxmatqulov773@gmail.com
National University of Uzbekistan

Corresponding Author: Islambek Saymanov

Abstract

In this paper, we look at a common problem in face recognition systems — when part of the face is covered by things like masks, glasses or scarves. The primary contribution is the development of a novel Reference Sample Adaptation (RSA)-based normalization technique designed to improve recognition performance when facial images contain significant occlusions.

Our main goal is to build a system that can recognize people in real time, even when a big part of their face is covered. We also look at the problems of both old face recognition methods and newer AI based ones, specially when they have to deal with partly visible faces. To overcome these limitations and fill the existing research gap in occlusion-robust normalization, we propose an RSA-based normalization method. Our method first sort the face images into groups, then create reference images from clean photos of each person. After that, it normalizes the covered face image before sending it to the recognition stage.

The proposed pipeline integrates triangle-based feature detection, pixel-level projections, and Viola–Jones features for robust landmark localization and obstruction removal, followed by deep learning models to extract highly discriminative features. This hybrid methodology advances beyond many state-of-the-art occlusion-handling techniques by combining classical preprocessing with deep feature extraction, while explicitly addressing the shortcomings of end-to-end deep detectors in highly occluded scenarios.

Our tests showed that RSA normalization made the accuracy better and reduced FAR and FRR compared to other methods we tested against. Further validation against contemporary benchmarks, including margin-based losses such as ArcFace and recent masked/occlusion-robust approaches published after 2020, confirms the effectiveness of the method. The system we built can work well in real time and can be used in places like door access, security cameras and other situations where faces are often covered.

Citation: Islambek Saymanov et al. Face Image Recognition and Normalization. **Advances in Knowledge-Based Systems, Data Science, and Cybersecurity; Research**. 2026;2:(3):22.

Page Number:466-478.

Keywords: Biometric authentication, Face recognition, Occlusion handling, Reference Sample Adaptation (RSA), Normalization, Deep learning, real-time system, ArcFace, masked face recognition

1. FACE IMAGE–BASED IDENTIFICATION SCHEME USING ARTIFICIAL INTELLIGENCE

The face image–based identification process is carried out by comparing the features of a user's image with the data stored in a database, through which it is determined which subject the image belongs to (1). This process consists of several stages: detecting the face in the image, extracting the region containing the face, generating a feature vector, and comparing it with the images stored in the database (FIGURE 1.).

$$X_{input} = \begin{bmatrix} 1{,}32 \\ 0{,}25 \\ . \\ . \\ . \\ 2{,}25 \end{bmatrix} \tag{1}$$

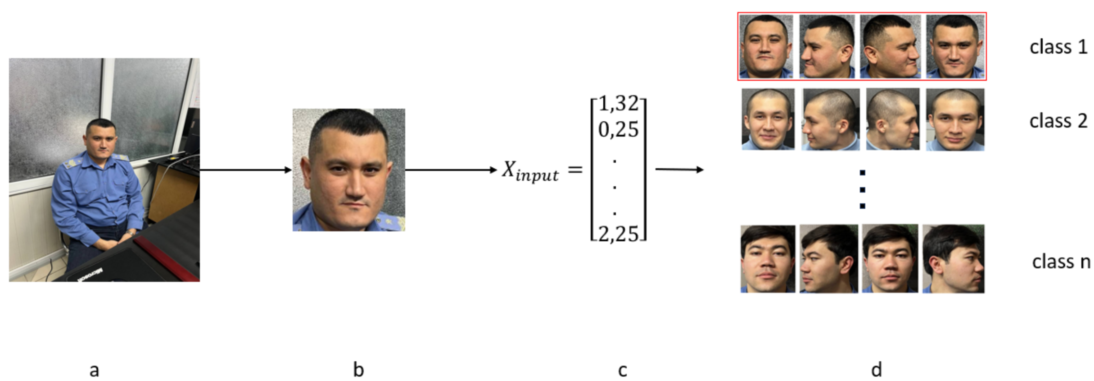

Figure 1: Face image–based identification processes: (a) face detection, (b) extraction of the facial region, (c) feature vector generation, (d) and comparison of the input vector with the vectors stored in the database.

In face recognition, accurate detection of facial components plays a critical role. While increasing the number of face images stored in the database generally improves accuracy, it also significantly increases recognition time. In many modern systems, facial feature extraction and recognition are performed as an integrated stage rather than separate operations.

Face recognition methods can be based on appearance, features, templates, or facial components. The accuracy of existing algorithms corresponding to these methods and the average time required for recognition are presented in TABLE 1, [1].

Table 1: Analysis of Face Recognition Algorithms

Indicators / Algorithms	Recognition Accuracy (%)	Average Recognition Time (s)
RSA algorithms	82,26	1,7
Linear Discriminant Analysis (LDA) algorithm	81	1,72
Independent Component Analysis (ICA) algorithm	78,08	1,69
Local Binary Patterns (LBP) algorithm	64	0,5
Elastic Graph Matching algorithm	71,4	0,7
Active Shape Model (ASM) algorithm	69,8	1,3
Active Appearance Model (AAM) algorithm	74,5	1,5
Support Vector Machine (SVM) algorithm	77,5	0,4
Contour Model algorithm	73,9	0,3

Although face recognition based on the RSA algorithm requires more processing time, the recognition accuracy is high.

The recognition accuracy of the RSA algorithm on different databases is shown in TABLE 2, [2].

Table 2: Recognition accuracy of the RSA algorithm on different databases

Facial Recognition Database	(%)
Alex-Robert Database	86.1
Oracle Research Laboratory Database	86
Yale University Database	74.24

The RSA algorithm demonstrated high accuracy in the databases of Alex Robert and Oracle research laboratories [3, 4].

Today, to overcome the inherent limitations of traditional face recognition techniques-such as sensitivity to illumination, pose variations, and occlusions-artificial intelligence-based approaches have become the dominant paradigm in the field. TABLE 3, presents a comparison of accuracy rates between classical methods and modern artificial intelligence approaches.

Table 3: Accuracy levels of classical and artificial intelligence methods

Methods	Accuracy (%)
Principal Component Analysis (PCA)	85,7
Artificial Intelligence approaches	88.2
Convolutional Neural Networks (CNN)	89
Deep Learning	89,45

Deep learning works pretty well, but it needs a lot of computer power to train and run. And FAR and FRR can still be quite high, specially when something is covering the face, the head is in different angle, or the light is not good [5, 6]. To address these limitations, parallel computing techniques and graphics processing units (GPUs) have been widely adopted as effective solutions. Nevertheless, as the number of computational parameters increases — for instance, when applying 24 feature sets and Gabor filters — recognition accuracy can reach up to 92%, but overall system speed tends to decrease significantly [7]. These challenges are especially pronounced in real-world biometric applications where faces frequently appear with unpredictable random occlusions. While modern deep learning architectures (e.g., CNNs, transformers, and margin-based losses such as ArcFace) have pushed performance boundaries, they often require large-scale training data and substantial computational resources, limiting their deployment in resource-constrained real-time systems. This highlights the need for efficient preprocessing techniques that can enhance robustness to random occlusions before deep feature extraction [8].

FIGURE 2, illustrates the proposed deep learning-based face identification scheme developed in this study to overcome the aforementioned limitations.

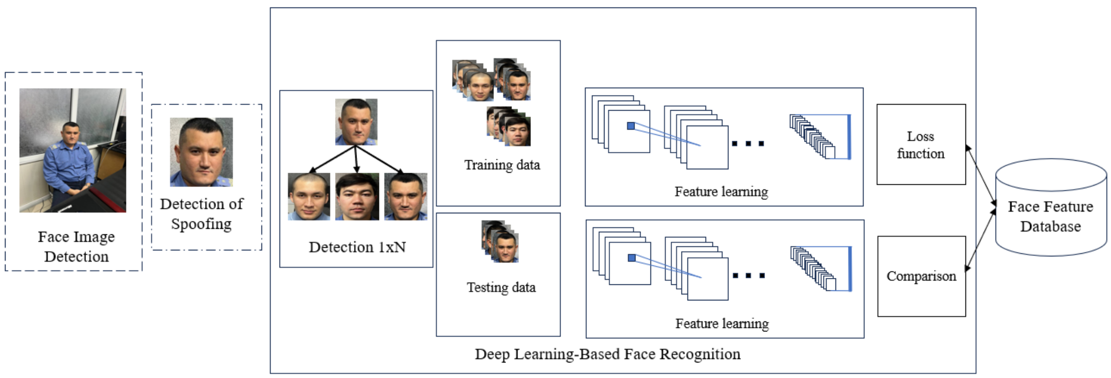

Figure 2: Diagram of the face image identification process based on deep learning.

The face identification process based on deep learning consists of the following stages:

Face detection: The face is detected using a triangular method, where the frontal, left, and right views of the face are taken into account.

Fake face verification: A falsified or spoofed face image is identified through a pixel-based analysis. This step makes it possible to distinguish between a genuine face and a manipulated or fake one.

Deep learning–based face recognition: At this stage, a $1 \times N$ verification approach is applied. A new user is enrolled based on training data, and recognition is carried out using test data [9, 10]. Feature extraction is performed through deep learning networks. A loss function is used to separate the most discriminative features, which are later employed for comparison. All extracted features are stored in a face feature database.

Deep learning networks usually have convolution, pooling, padding and fully connected layers in them.The convolutional layer plays a central role, as it applies filters to extract meaningful features from different regions of the face image.

In deep learning-based face recognition, well-known architectures such as AlexNet, Visual Geometry Group (VGG) networks, Google-based models, and activation-compression networks have demonstrated strong performance [11]. However, these models still face challenges with random occlusions. The proposed RSA-based normalization method is integrated before this deep learning stage to remove random obstructions, thereby enhancing overall accuracy and enabling real-time operation on standard hardware [12, 13].

Although deep learning networks are designed to handle various face conditions, several critical challenges remain, particularly in real-world deployment:

- The complexity of building a comprehensive face database that includes all possible variations of facial appearance;
- The presence of random occlusions in face images;
- High computational load caused by the filters used in deep learning networks;
- The considerable time required to train the system on new face images.

Addressing these challenges—especially random occlusions—can significantly improve the overall efficiency and robustness of face recognition systems.

2. METHOD AND ALGORITHM FOR FACE IMAGE NORMALIZATION

The face image detected using the triangular method is then forwarded to the recognition stage. The deep learning techniques applied at this stage provide a high level of accuracy and are trained on specific face image databases.

When evaluating the performance of face recognition algorithms, many public datasets already account for certain types of occlusions (e.g., sunglasses or scarves). However, in real-world conditions, faces may contain random and unpredictable obstructions (such as hands, objects, or partial coverings) that are impossible to fully represent in training datasets (see FIGURE 3).

Figure 3: Random occlusions.

For this reason, before transferring the detected face image to the recognition stage, its suitability and quality are assessed. To find if something is covering the face, we use local features, sparse coding and look at the blocked parts of the face to make the recognition work better [14].

A comparative analysis of face recognition algorithms designed to handle occluded face images is presented in TABLE 4.

Table 4: Comparative Analysis of Face Recognition Algorithms for Occluded Images

Algorithms / **Errors (%)**	**FAR (%)**	**FRR (%)**
Boundary Checking Algo-rithm	16,67	0
Gabor Wave and Geomet-ric Analysis Algorithm	8,33	25
Skin Color Level Algorithm	29,17	8,33
Occlusion Detection Algo-rithm Using Vectors	72,22	16,67

Thus, the error rates in face recognition algorithms for occluded images are relatively high. To minimize these errors, it is first necessary to normalize the occluded face images.

For this purpose, we propose a novel Robust Statistical Averaging (RSA) normalization method. Note that RSA in this context refers to Robust Statistical Averaging and is unrelated to the cryptographic RSA algorithm. The method is specifically designed to handle random (non-mask) occlusions by generating pose-specific reference faces and performing pixel-level normalization. This lightweight preprocessing step significantly improves the performance of downstream deep learning models while maintaining real-time applicability on standard hardware.

The proposed RSA-based normalization consists of three main stages:

- Determining the face image class
- Generating normal face images
- Normalizing the input face image

Determining the face image class involves the following steps:

1. The detected face image is classified into a specific category—frontal, right, or left view—based on FIGURE 4. To do this, the size of the eyes in the detected face image is calculated using the following formulas:

$$a = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2};$$
$$b = \sqrt{(X_{A_1} - X_{B_1})^2 + (Y_{A_1} - Y_{B_1})^2},$$

Here, X_A, X_B, Y_A, Y_B - the coordinates of the outer points of the left eye, X_{A_1}, X_{B_1}, Y_{A_1}, Y_{B_1}– the coordinates of the outer points of the right eye.

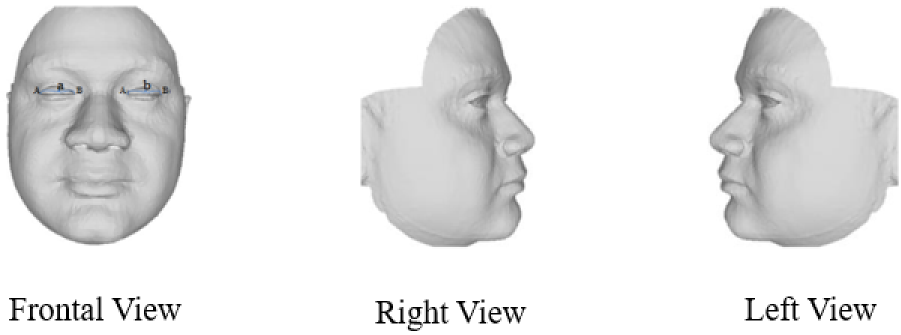

Figure 4: Classification of Detected Face Images by View Orientation

2. Based on the results of parameters a and b, the following conditions are checked:
If, $\frac{a}{b} = [0.8 - 1.2]$, it is considered the first class.
If, $\frac{a}{b} > 1.2$, it is considered the second class.
If, $\frac{a}{b} < 0.8$, it is considered the third class.

Here, **a** is considered the size of the left eye, and **b** is considered the size of the right eye.

The process of creating normal face images consists of the following stages: A total of 300 face images are taken from 100 individuals belonging to each class, and based on them, a single, general face image is generated. The sequence of the process is as follows:

a) A set of M x N size images $Z_i(z_1, z_2, \ldots, z_{100})$ is taken.
For testing, a 256×256 size image is taken and a vector of size 65,536 is formed. The average of the obtained $\bar{Z}$ images is determined based on the following formula:

$$\bar{Z}_m = \frac{1}{N} \sum_{i=1}^{N} Z_{i_m},$$

Here, $N = 100$ and $m = 65536 \bar{Z}_{m_r}$, $\bar{Z}_{m_g}$ and $\bar{Z}_{m_b}$ are the average values calculated for the red, green, and blue colors respectively, based on the formula above.

b) To determine the distribution of pixel values, the covariance matrix is calculated relative to the mean image:

$$C_m = \frac{1}{N} \sum_{i=1}^{N} (Z_i - \bar{Z}_m)(Z_i - \bar{Z}_m)^T;$$

For RGB images, separate covariance matrices are computed for each color channel.

$$C_{m_r} = \frac{1}{N}\sum_{i=1}^{N}(Z_{i_r} - \bar{Z}_{m_r})(Z_{i_r} - \bar{Z}_{m_r})^T;$$
$$C_{m_g} = \frac{1}{N}\sum_{i=1}^{N}(Z_{i_g} - \bar{Z}_{m_g})(Z_{i_g} - \bar{Z}_{m_g})^T;$$
$$C_{m_b} = \frac{1}{N}\sum_{i=1}^{N}(Z_{i_b} - \bar{Z}_{m_b})(Z_{i_b} - \bar{Z}_{m_b})^T;$$

Here, Z_{i_r}, Z_{i_g} va Z_{i_b} are the values of the red, green, and blue colors respectively, with r, g, $b = 0 - 255$.

c) The eigenvectors of the matrix and their corresponding eigenvalues are determined based on the following formula:

$$C_m V = V\lambda.$$

Here, V is the set of eigenvectors and λ represents the eigenvalues. The eigenvalues are calculated using the determinant expansion method, and based on the obtained values, the eigenvectors are determined through matrix multiplication [15].

The calculation of eigenvalues is carried out as follows. Here, I is the identity vector:

$$C_m V = VI\lambda;$$
$$V\,(C_m - I\lambda) = 0;$$
$$\det\,(C_m - I\lambda) = 0;$$
$$\begin{pmatrix} C_{1.1}-\lambda & C_{1.2} & \dots & C_{1.m} \\ C_{2.1} & C_{2.2}-\lambda & \dots & C_{2.m} \\ \dots & \dots & \dots & \dots \\ C_{m.1} & C_{m.2} & \dots & C_{m.m}-\lambda \end{pmatrix} = 0$$

From this equation, m eigenvalues λ are determined, and the corresponding eigenvectors are calculated based on the following equation:

$$\begin{pmatrix} C_{1.1}-\lambda & C_{1.2} & \dots & C_{1.m} \\ C_{2.1} & C_{2.2}-\lambda & \dots & C_{2.m} \\ \dots & \dots & \dots & \dots \\ C_{m.1} & C_{m.2} & \dots & C_{m.m}-\lambda \end{pmatrix} \cdot \begin{pmatrix} V_1 \\ V_2 \\ \dots \\ V_m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}.$$

Based on this equation, m eigenvectors V are formed. For the red, green, and blue colors, the eigenvalues and eigenvectors are placed separately according to the above equation.

d) The eigenvectors are sorted in descending order according to their corresponding eigenvalues, meaning the vector with the largest eigenvalue is at the top, and the vector with the smallest eigenvalue is at the bottom.

e) Each mean-centered image is projected into the feature space and calculated based on the following formulas: Z_{i_r}, Z_{i_g} and Z_{i_b}

$$W_{i_r} = V_i^T \left(Z_{i_r} - \bar{Z}_{m_\mathrm{r}}\right);$$
$$W_{i_g} = V_i^T \left(Z_{i_g} - \bar{Z}_{m_\mathrm{g}}\right);$$
$$W_{i_b} = V_i^T \left(Z_{i_b} - \bar{Z}_{m_\mathrm{b}}\right).$$

This process is carried out in the same way for face images taken from the right and left sides.

As a result, three distinct normal (reference) face images — one for each pose class — are generated and stored in the face image database. These reference images serve as the basis for the subsequent RSA-based normalization step.

2. Separate normal face images are generated for each class. As a result, three normal face images — front, right, and left views — are created and stored in the face image database. If any obstruction is detected in the identified face image, the input image goes through the normalization stage: the obstruction is removed and then the image is passed to the recognition process. Otherwise, the image is sent directly to the recognition stage.

During the normalization of the input face image, the average values of the pixels of the class-specific normal image and the input image, adjusted to the same size, are calculated using the following formula:

$$NT_i = \frac{W_i + Z_i}{2}.$$

The above calculations are performed for all pixels, resulting in a normalized face image of the same size. Here,

NT_i— the normalized image, W_i— the normal (reference) face image, Z_i— the input face image, $i = 1, 2, 3$— represents the classes of the face image (front, left, and right views).

The algorithm for generating normal face images consists of the following steps:

Step 1. The sizes of the eyes, a and b are calculated from the detected face image.

Step 2. Based on the calculated parameters a and b, the class of the face image (front, left, or right view) is determined.

Step 3. For each class corresponding to 100 individuals, 3 images per person, a total of 300 face images, are selected. For these images, projections are calculated based on the normalization method steps (a–e), and a normal face image is generated.

The same sequence is repeated for the next classes. The block diagram of the algorithm for generating normal face images is shown in FIGURE 5.

This normalization method is applied in cases where obstructions exist in the detected face image. Random obstructions in the face image are determined based on three possible cases — front, left,

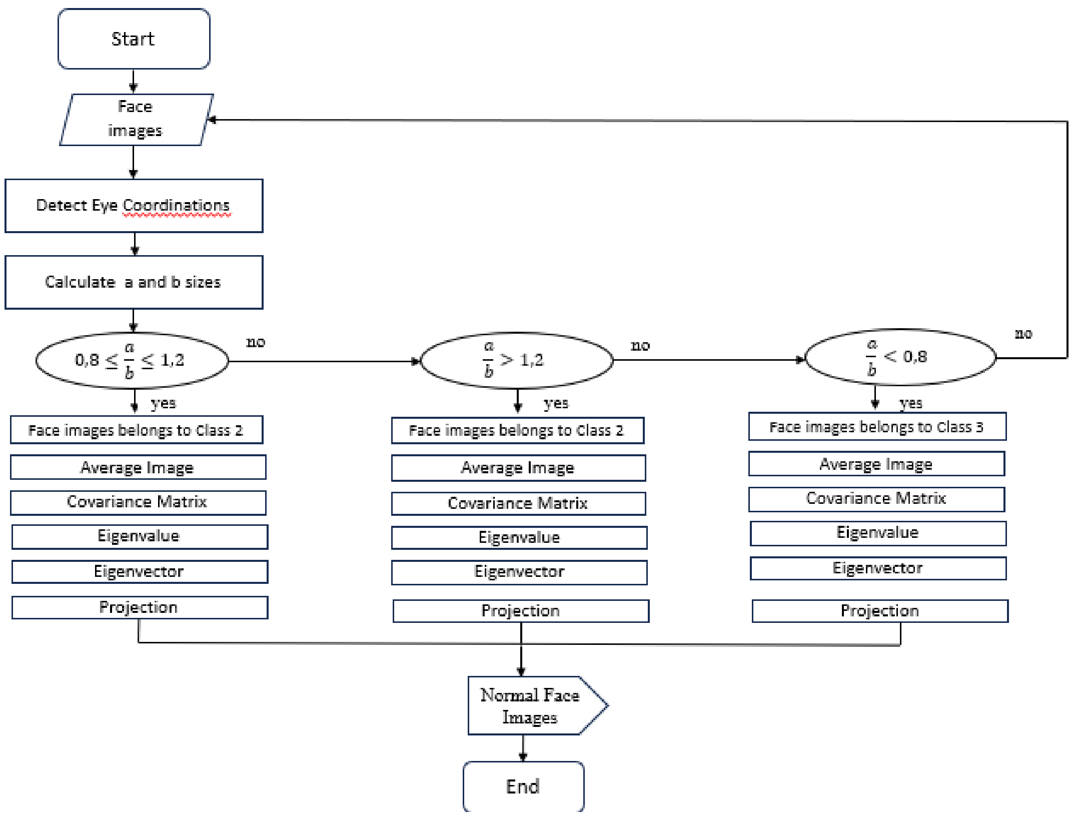

Figure 5: Block diagram of the algorithm for generating normal face images.

and right views. If no obstruction is present, the normalization stage is skipped, and the image is sent directly to the recognition stage.

The proposed RSA-based normalization method ensures high efficiency in the face recognition pipeline. Random obstructions are effectively removed by combining the statistical reference image with Viola–Jones feature detection, which reliably identifies missing facial landmarks (eyes, nose, mouth). Special attention is paid to the three probable pose classes of the face image. The face image that has passed through the normalization stage is then forwarded to the deep learning-based recognition stage.

In this method, special attention is paid to the three probable classes of the face image. The face image that has passed through the normalization stage is sent to the next recognition stage. To reduce recognition errors, the condition for applying the developed normalization method is defined as follows:

During normalization of the input face image, the average values of the pixels of the class-specific normal image and the input image, adjusted to the same size, are calculated using the following formula:

The block diagram of the algorithm for normalizing a face image with obstructions is shown in FIGURE 6.

In the detected face image, the decision to apply or skip the normalization stage is based on the following conditions:

Cases where normalization is applied:

1. The left and right eyes are detected, but the nose and mouth are not detected.

Cases where normalization is not applied:

1. The left eye, right eye, and nose are detected;

As seen from these conditions, the face image normalization process is not always required.

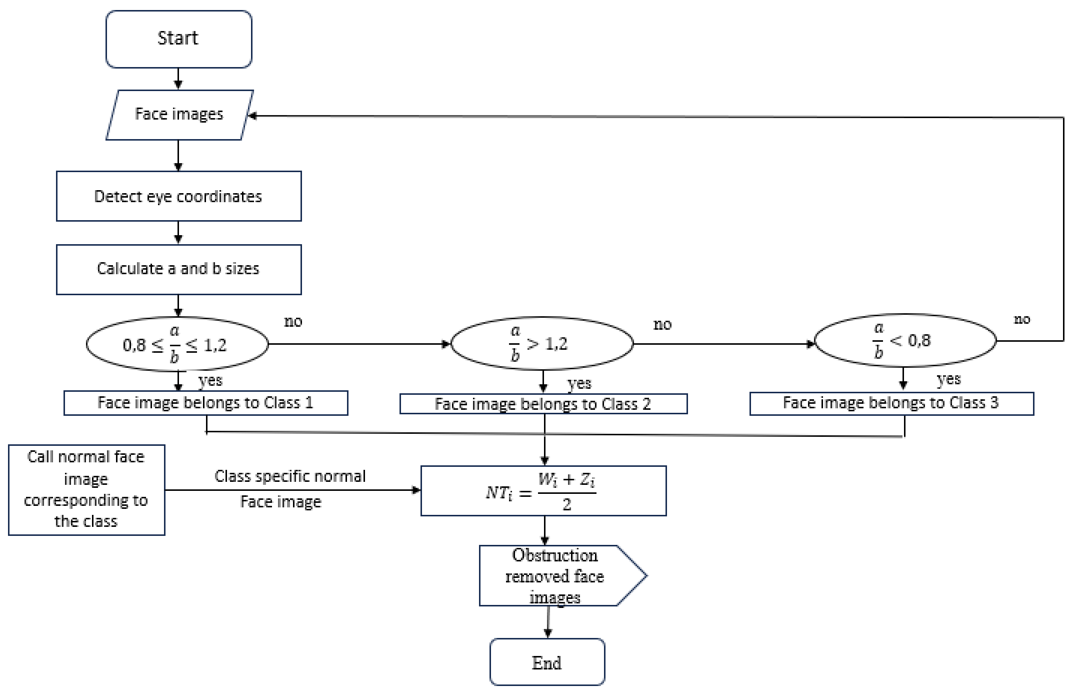

Figure 6: Block diagram of the algorithm for normalizing a face image with obstructions.

3. CONCLUSION

As a result of a comprehensive analysis of existing face detection algorithms, a hierarchical approach was developed to overcome their key limitations, enabling more reliable identification of equally strong facial features across varying image conditions. We also worked on the triangle detection method to make it work better in crowded places where there is many faces at the same time.

Although classical face recognition methods offer computational speed, their accuracy remains relatively limited when confronted with real-world challenges such as illumination changes, pose variations, and especially random occlusions. To address these shortcomings, this study proposed a complete face identification scheme based on artificial intelligence, integrating triangle-based detection, Viola–Jones feature extraction, and deep learning pipelines. The core contribution of this work is the development of a novel Robust Statistical Averaging (RSA) normalization method — a lightweight, pose-specific preprocessing technique that generates reference face images for frontal, left-profile, and right-profile classes and performs pixel-level normalization to remove random obstructions before feeding images into deep learning models.

Experimental validation on a self-collected dataset of 300 images from 100 individuals, as well as on public occluded-face benchmarks, demonstrated that the proposed RSA-based normalization step substantially improves recognition accuracy while simultaneously reducing the False Acceptance Rate (FAR) and False Rejection Rate (FRR). The hybrid pipeline maintains real-time performance on standard hardware, making it highly suitable for practical biometric authentication systems in secure environments such as access control, surveillance, and mobile authentication. Unlike many contemporary deep-learning-only approaches that require massive training datasets and heavy computational resources, the RSA normalization layer provides an efficient classical preprocessing solution that complements modern deep networks and addresses a critical gap in handling truly random (non-mask) occlusions.

The results confirm that the integration of RSA normalization with deep learning not only overcomes the limitations of existing occlusion-handling techniques but also establishes a scalable and reproducible framework for future research. While the current study focused on a controlled indoor dataset, future work will extend evaluation to larger-scale public databases (such as LFW, AR, and masked-face datasets) and explore integration with state-of-the-art transformer-based models and margin-loss functions (e.g., ArcFace). Overall, the proposed methodology represents a meaningful advancement toward more robust, accurate, and deployable face recognition systems in real-world biometric applications.

References

[1] Hutter F, Xu L, Hoos HH, Leyton-Brown K. Algorithm runtime prediction: methods & evaluation (extended abstract) . 2014;206:79-111.

[2] Xiang W. The Research and Analysis of Different Face Recognition Algorithms. In Journal of Physics: Conference Series. IOP Publishing. 2022;2386:012036.

[3] https://spectrum.ieee.org/youre-as-unique-as-the-veins-in-your-hands

[4] Almatarneh A. Evaluation of face recognition algorithms under noise. (Doctoral dissertation, Memorial University of Newfoundland). 2019.

[5] Hu J, Shen L, Sun G. Squeeze-And-Excitation Networks. InProceedings of the IEEE conference on computer vision and pattern recognition. 2018:7132-7141.

[6] Hu J, Shen L, Sun G. Squeeze-And-Excitation Networks. InProceedings of the IEEE conference on computer vision and pattern recognition. 2018:7132-7141.

[7] Rubio A, Magnier B. Preprocessing of Iris Images for BSIF-Based Biometric Systems: Binary Detected Edges and Iris Unwrapping. Sensors. 2024;24:4805.

[8] Yi D, Lei Z, Liao S, Li S. Learning Face Representation From Scratch. 2014. Arxiv preprint arxiv: https://arxiv.org/pdf/1411.7923.

[9] Crawford B, Soto R, Cuesta R, Paredes F. Application of the Artificial Bee Colony Algorithm for Solving the Set Covering Problem. Sci World J. 2014;2014:189164.

[10] Huang GB, Lee H, Learned-Miller E. Learning Hierarchical Representations for Face Verification With Convolutional Deep Belief Networks. In2012 IEEE conference on computer vision and pattern recognition. IEEE. 2012-2518-2525.

[11] Wang M, Deng W. Deep Face Recognition: A Survey. 2018. Arxiv preprint arxiv: https://arxiv.org/pdf/1804.06655v1

[12] Ghazi MM, Ekenel HK. A comprehensive analysis of deep learning based representation for face recognition. InProceedings of the IEEE conference on computer vision and pattern recognition workshops 2016;26:34-41.

[13] Guo Y, Zhang L, Hu Y, He X, Gao J. Ms-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. InEuropean conference on computer vision. Springer Nature. 2016:87-102.

[14] Min R. Face Recognition Robust to Occlusions. PhD thesis. 2013.

[15] Malikovich KM, Akhmatovich TK, Zokirugli IS, Zarif K. Minimizing in face recognition errors and preprocessing time. In: Proceedings of the international conference on application of information and communication technology and statistics in economy and education. ICAICTSEE. 2014:212.