Journal of Advanced Research Design 137, Issue 1 (2026) 243-266

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Integrating facenet and extreme learning machines for enhanced face identification: addressing real-world challenges with advanced image processing techniques

Areej A. Abed[1,*], Abdul Monem S. Rahma[2], Omar A. Dawood[3]

[1] Computer Science Department College of Computer Science and Information Technology, University of Anbar, AL-Anbar, Iraq
[2] Computer Science Department, College of Science, University of Al-Maarif, Baghdad, Iraq
[3] Computer Science Department, College of Computer Science and Information Technology, University of Anbar, AL-Anbar, Iraq

ARTICLE INFO

Article history:
Received 7 January 2025
Received in revised form 23 April 2025
Accepted 10 July 2025
Available online 20 July 2025

Keywords:
Identity Verification; Face identification; FaceNet; Extreme Learning Machines

ABSTRACT

Face identification forms an important area of computer vision due to its application in many real-time applications related to the security and surveillance of law enforcement agencies. Deep learning algorithms have revolutionized building face identification systems with high accuracy and speed. There is a dire need for systems that achieve more reliability and can handle a diverse range of scenarios. This work uses a junction of FaceNet, a convolutional neural network (CNN)-based model for feature extraction, and extreme learning machines (ELM) to form a face identification system. The projected performance of the proposed system is expected to outperform that of the existing systems in terms of accuracy and resilience. The improved performance of the proposed systems is rooted in FaceNet and ELM, which can capture intricate facial features and patterns accurately. The ELM runs very fast with a single hidden layer feed forward neural network. Extensive experiments were conducted on the Youtube-faces dataset, and images were captured in real time. The proposed method had a recognition accuracy of 99.1 percent, a precision of 98.5 percent, a recall of 97.8 percent, and an F1-score of 98.1 percent. Further, we have also applied pruning and quantization to compress the FaceNet+ELM model for its efficient performance on low computational power devices. Pruning reduces redundant weights and neurons, while quantization converts parameters from 32-bit to 8-bit, greatly reducing the model size and increasing the inference speed.

1. Introduction

The human brain is exceptionally good at recognizing objects in real time because it performs extremely complex operations to identify patterns [1]. Computer vision is a sub-discipline of computer science that seeks to replicate this functionality to help create systems that perform mandates such as face identification [2]. These systems have become integral to various applications, including security surveillance, employee attendance tracking, and criminal identification [3]. Deep learning, more so through CNNs, has revolutionized tasks such as image classification, object

* *Corresponding author.*
E-mail address: are21c1004@uoanbar.edu.iq

https://doi.org/10.37934/ard.137.1.243266

detection, and image segmentation [4]. CNNs are very good at face recognition because of their layered architecture, which learns to identify complex visual features from data [5]. CNNs, in this respect, do not require manual feature extraction, making the process faster and more accurate, unlike traditional methods such as Fisher Faces [6], Local Binary Pattern Histograms [7], and Eigen Faces [8].

Modern smartphones apply CNNs to the user's facial recognition. These models are trained to facilitate reliable unlocking mechanisms by training on images taken under different conditions. Most public places apply security systems that use CNNs to identify suspects involving facial image matching with databases containing images of wanted people. High-security places, such as airports or banks, also apply CNN-based systems that involve cross-referencing one's identity through facial images and official documents [9]. These methods are normally incorporated with Support Vector Machines to enhance face recognition accuracy through classification. CNNs work excellently in feature extraction, while the last classification layer uses the SVM method. The drawbacks of the SVM method include high computational resources and time complexity with large datasets. Training an SVM on huge data sets can be slow and memory-consuming; advanced optimization techniques are required to optimize efficiency [10].

FaceNet is a gigantic leap in the field of face recognition. It projects facial images as vectors in a space characteristic of precise discrimination. Trained with the triplet loss algorithm, FaceNet minimizes the distance for similar vectors and maximizes it for different vectors [11]. These vectors are used after feature extraction as an input to many other models in classification, like Extreme Learning Machines. ELM resolves the limitations that exist with SVMs. Characterized by rapid training and high efficiency, ELMs use random hidden layers that do not need training, reducing the complexity of training to a large extent. The suitability of ELMs for large datasets, free from memory and time complexities as in the case of SVM, shall provide integration with features extracted from face nets and ELM to enable fast and accurate face classification [12].

Preliminary steps can be applied before feeding images to FaceNet to deal with the most frequent problems related to light, noise, and distortion. For example, histogram equalization [13] will standardize the lighting distribution, while Gaussian filters [14] are used to remove noise. Geometric distortion correction techniques ensure that the projected face will be consistent. These preprocessing steps will enhance the quality of features extracted by FaceNet and ensure more accurate and reliable classification. Pruning and quantization techniques [15], which include model compression, were applied to let our model of FaceNet+ELM run on devices with restricted computational power. Consequently, pruning reduces redundant weights and neurons, while quantization reduces the parameters from 32 bits to 8 bits; it significantly reduces the model size and improves speed in inference.

Although face recognition has benefited from tremendous technological advancements, there is still a gap in achieving extremely accurate and efficient systems under diverse conditions encountered in the real world. This paper discusses a face identification system marrying the feature extraction capabilities of FaceNet with an ELM for classification. The core research question to be answered is: How can we build a face identification system that is much more accurate and resilient than current approaches yet efficient enough to run on devices with limited computational power? The main contributions of the work are presented here:

1. **Integration of FaceNet and ELM**: Combining FaceNet for robust feature extraction with Extreme Learning Machines (ELM) for efficient and accurate face classification.

2. **Enhanced Image Processing**: Utilizing advanced techniques like histogram equalization, Gaussian filtering, and geometric distortion correction to improve feature extraction quality and recognition accuracy.
3. **Real-Time and Large Dataset Efficiency**: Demonstrating the system's capability for real-time applications and efficient handling of large datasets, addressing traditional classifiers' limitations.
4. **Model Compression**: Applying pruning and quantization to the FaceNet+ELM model for reduced size and enhanced inference speed on low computational power devices.

This study presents a robust and efficient face identification system capable of addressing real-world challenges, thereby advancing the field of computer vision and its applications.

1.1 Related Work

Facial identification is, therefore, one of the oldest challenges in the field of computer vision, and much effort has been channeled by researchers to devise ways of increasing the accuracy and efficiency of the procedure through different approaches. The feature extraction procedure is vital to any face identification system and comprises the selection and subsequent extraction of salient facial features. In most cases, manual feature extraction methods have been implemented effectively in certain scenarios through Principal Component Analysis and the Viola-Jones algorithm [16]. However, most of these methods have one or more of the following limitations: they require domain knowledge from an expert, vulnerability to changes in lighting and pose, and complex nonlinear relationships in the data cannot be captured. For this purpose, researchers have also explored machine learning algorithms such as Support Vector Machines for improved facial identification [17]. Methods based on SVMs are also very promising but still rely on manually engineered features and can be brittle to complex, nonlinear patterns in data. Deep learning techniques, especially Convolutional Neural Networks, have changed the fate of facial identification overnight. Not only do CNNs remove the need for feature engineering by automatically learning hierarchical features directly from raw images, but due to their ability to model these complex nonlinear relationships in the data and being very robust to lighting variations, pose variations, and many more, making CNNs very efficient for facial identification [18].

While manual feature extraction methods and approaches based on SVM continue to find applications in this area, the better performance and flexibility of CNNs make them the overwhelming choice for modern face identification systems [19]. In recent research, it has been underlined that one should take care not only of the accuracies of these systems but also of intersectional biases within them. Given that AI attendance monitoring systems are increasingly being rolled out, it becomes very important to design and roll out such systems as ethically and inclusively as possible by accounting for a diversity of facial characteristics and demographic groups. FaceNet has been one of the leading deep models in state-of-the-art performance concerning face identification. This FaceNet learns to extract robust and discriminative features invariant to lighting, pose, and other factors from raw image data. Recent research shows that FaceNet significantly performs with high accuracy values across different race and gender groups, outperforming traditional methods that have been proven biased in most cases [20].

Moreover, FaceNet helped to build individualized face recognition systems that have become personally adapted. The outcome is more inclusive and fair. Even though CNN and FaceNet introduced a new level of performance in terms of image identification, there is still a wide scope for improvement, mainly in the areas of bias and discrimination, which leads to the development of another promising approach: the hybrid Convolutional Neural Network and Extreme Learning

Machine model. In this case, the hybrid model will integrate the feature learning ability of CNN with the fast classification capability of ELM. As such, it can provide quick and accurate classification while maintaining robustness due to variations in light, pose, and other factors. Moreover, research has been focused on incorporating facial attributes, including gender, race, and age, during face identification. This extra information can be used with features to increase performance in general prediction and better handle problems w.r.t bias and discrimination [21].

The research into face identification has improved considerably from the early traditional manual extraction of features to state-of-the-art current machine learning algorithms, especially CNNs. Although manual feature extraction can find applications and approaches based on SVM, which are still finding them, the excellence in performance and flexibility has kept CNN at the top of all modern face identification systems. Another point must be how to develop unbiased, fair, and inclusive solutions during the development of facial recognition systems. Deep learning can, however, along with facial attributes and, keeping in mind possible bias in such systems, lead one through to fair and accurate systems that can serve diverse communities in this field [22]. Table 1 details the recent studies on face identification, including their contributions, limitations, achieved results, and numerical performance data.

Table 1
Summary of Recent Studies on Face Identification.

Study	Contributions	Limitations	Achieved Results	Numerical Results
EFaR 2023: Efficient Face Recognition Competition [23]	Developed a ranking system (Borda count) to evaluate face recognition solutions based on verification accuracy and deployability.	Performance depends on specific datasets; compactness and efficiency vary across solutions.	High accuracy on datasets like CPLFW, CFP-FP, CALFW, AgeDB30, LFW, and TAR of IJB-C.	LFW: 99.85%, CPLFW: 92.45%, CFP-FP: 96.20%, AgeDB30: 98.35%, TAR@FAR=0.01% on IJB-C: 93.40%
Near-Infrared and Visible Light Face Recognition [24]	Proposed adversarial cross-spectral face completion for better NIR-VIS recognition.	Sensitive to cross-spectral variances and lighting conditions.	Improved recognition rates in challenging lighting and cross-spectral scenarios.	NIR-VIS: 85.50%
Past, Present, and Future of Face Recognition [25]	Reviewed 2D and 3D methods, highlighting the impact of deep learning and future research directions.	Performance degrades under uncontrolled conditions such as varied lighting and facial expressions.	Identified significant advancements and the need for robust models in unconstrained environments.	N/A
Face Image Quality Enhancement Study [26]	Evaluated 21 photometric normalization methods; proposed Weber-face for illumination-	Effectiveness varies with different image qualities; frontalization and deblurring methods can fail	Enhanced recognition rates by improving image quality through	Middle-quality: 87.30%, Low-quality: 75.60%

	insensitive representation.	in extreme cases.	normalization and pose correction techniques.	
A Review on Face Recognition Systems [27]	Summarized recent approaches, including deep learning, for improving pose-invariant and cross-pose recognition.	High computational cost and need for large annotated datasets.	Significant improvements in cross-pose face recognition using landmark-oriented depth warping and deep learning.	Cross-pose: 90.70%
Adversarial Cross-Spectral Face Completion [28]	Developed a method for cross-spectral face recognition using adversarial networks.	Challenges with diverse lighting conditions and spectral variances.	Achieved enhanced accuracy in cross-spectral scenarios, especially in NIR to VIS face recognition.	Cross-spectral: 88.50%
Pose-Invariant Face Recognition using Deep Learning [29]	Introduced deep learning techniques for recognizing faces across various poses.	Requires extensive training data; computationally intensive.	Improved recognition rates for faces with different poses using deep learning models.	Pose-invariant: 92.10%
Robust Cross-Pose Face Recognition [30]	Utilized landmark-oriented depth warping for recognizing faces in different poses.	High dependency on accurate landmark detection.	Enhanced recognition accuracy for faces with pose variations.	Cross-pose: 89.80%
3D-2D Face Recognition with Pose and Illumination Normalization [31]	Developed a system combining 3D and 2D data for face recognition, addressing pose and illumination challenges.	Sensitive to facial expressions; computationally demanding.	Significant improvement in recognition rates under varying pose and illumination conditions.	3D-2D: 93.50%
Photometric Normalization for	Implemented Weber-face method for	Effectiveness decreases with	Improved face recognition accuracy by	

Enhanced Face Recognition [32]	illumination-insensitive face representation	extreme lighting conditions.	reducing the impact of lighting variations.	Normalized: 89.30%

These works bring out the advances that have been realized and the challenges still facing the technology on face identification, thereby calling for continued innovation to reach higher accuracy, efficiency, and scalability. Our research addresses these gaps through the proposition of a face identification system combining the strengths of FaceNet in feature extraction with ELMs for classification. In this paper, we illustrate a hybrid approach that capitalizes on ELMs' fast training time and high efficiency in processing large datasets while maintaining high accuracy and speed. We further optimize the system for deployment on edge devices with limited computational power through model compression techniques such as pruning and quantization. The proposed system is tested by various experiments, showing that real-time functionality is possible due to its recognition accuracy of 99.1% at a runtime of 49 milliseconds.

2. Materials and methods

This work focuses on developing a strong and efficient system for face identification by integrating high-accuracy feature extractors like FaceNet with ELMs for classification. The system will be created using six stages: data acquisition, face detection, preprocessing, feature extraction, ELM classification, and model compression. Each step ensures the system is deployed in resource-constrained devices through pruning and quantization. The methodology also includes how each step will be followed at each stage to be replicated. Figure 1 illustrates the main steps of the proposed methodology.

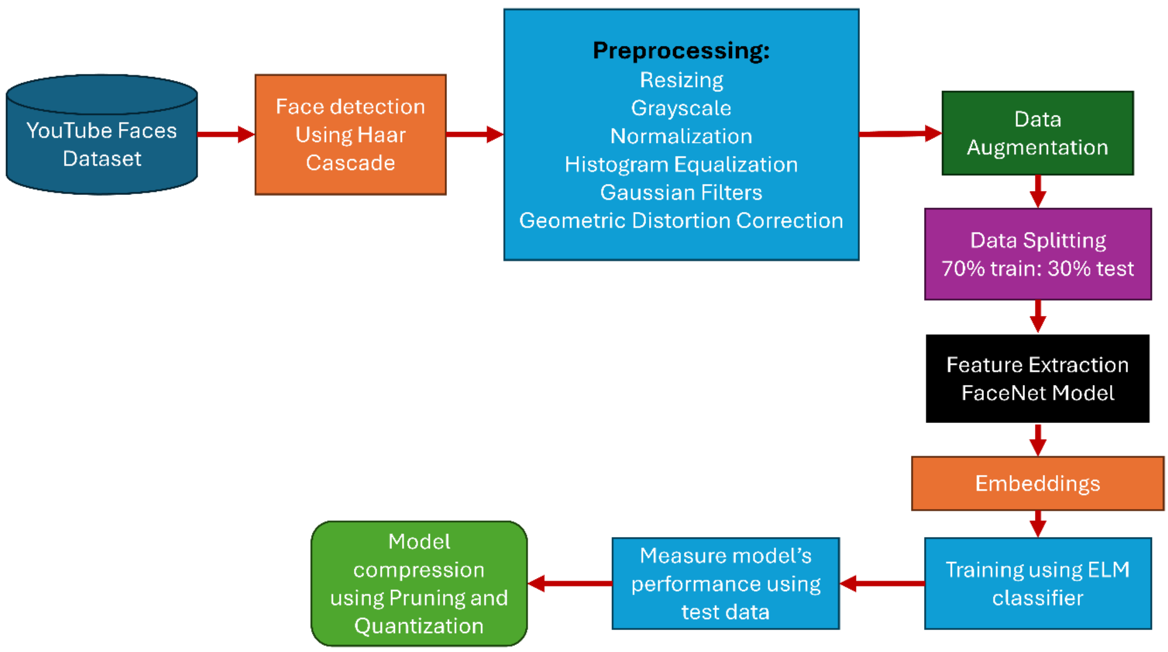

Fig. 1. The overall block diagram for the proposed methodology.

2.1 Data Acquisition

A face identification system will be trained and tested on the YouTube Faces Dataset (YTF), which contains 3,425 videos from 1,595 people, with an average of 2.15 videos for each person. Every video includes frames representing a subject in several variations in pose, illumination conditions, and expressions. This large dataset is derived from YouTube, offering a rich diversity of face images extracted from different video frames [33].

The YTF dataset is especially important for our work because it is extremely diverse and real-world variable. It picks up all sorts of conditions: angles, light changes, facial expression—elements that will test the development of a robust and accurate face identification system. This set will ensure high generalization to most other scenarios typically encountered in practical applications, hence improving the reliability and performance of the system in real-life conditions. Using the YTF dataset for testing assures that our face identification system is exposed to a broad spectrum of real-world challenges, thus making it more adaptable and effective in various environments. Such comprehensive and dynamic data is the basis for achieving high accuracy and robustness in the face identification model.

2.1.1 Face detection

Face detection is done through frontal face detection using the Haar Cascade Classifier, which is efficient and accurate. This module is important in our system as it reliably identifies the faces in the images. In this way, later preprocessing and feature extraction steps can be performed.

The Haar Cascade Classifier is a machine learning-based approach using a series of pre-trained classifiers to detect objects, especially human faces. Originally, the algorithm had been trained on a large collection of positive images with faces and negative images without faces. During training, it found faces based on the patterns in pixel intensity values. These very simple rectangular features, which correspond to contrasts between regions of an image, are called Haar features or Haar wavelet features [34].

2.1.2 Haar Features

$$Feature = \sum_{i \in white} I(i) - \sum_{j \in black} I(j) \qquad (1)$$

where $I(i)$ is the pixel intensity at position i in the white region, and $I(j)$ is the pixel intensity at position j in the black region. The algorithm uses a series of these features to build a robust classifier.

Integral Image

The algorithm employs an integral image representation to compute the Haar features efficiently. The integral image at any pixel (x, y) contains the sum of all pixel values above and to the left of (x, y), inclusive. This allows the calculation of the sum of pixel values within any rectangular area in constant time, significantly speeding up the feature computation process.

Mathematically, the integral image $II(x, y)$ is defined as:

$$II(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \qquad (2)$$

where $I(x', y')$ is the pixel intensity at position (x', y').

2.1.3 Cascade of Classifiers

During face detection, many stages drive the process through a Haar Cascade Classifier; each stage is itself a strong classifier. Each stage applies a set of Haar features to the input image and computes a feature value. If the feature value is greater than or equal to some threshold, the image region will be passed on to the next stage; otherwise, it will be rejected. The cascading effect quickly eliminates regions of the image that do not contain faces, thus allowing the algorithm to focus its resources on the most promising areas.

The decision rule for passing through the stages can be represented as follows:

$$Decision = \begin{cases} 1 \ if \ \sum_{i=1}^{N} \alpha_i h_i(x) \ \geq T \\ 0 \qquad otherwise \end{cases} \tag{3}$$

Where α_i are the weights, $h_i(x)$ are the weak classifiers (Haar features), N is the number of stages, and T is the threshold.

2.1.4 Face Detection Procedure

This research uses a pre-trained Haar Cascade Frontal Face Classifier located in the OpenCV library. This classifier was applied against all frames of videos from the YTF to extract and crop faces. Beginning with every frame of the video, it is turned into a grayscale, referencing that the computational process will be lightened and the detection speed will be enriched. These grayscale images are then supplied to the Haar Cascade Classifier, where an image is scanned at multiple scales for different-sized faces.

Face detection crops the Region of Interest (ROI) from the image and saves it for further preprocessing and extraction of features. This method suits our work because it can handle different face orientations and lighting conditions, common in the diverse and dynamic YTF dataset. Inherent in the Haar Cascade Classifier is a fast and reliable face detection process, which is the basis for the other stages of our face identification system, rapidly discarding non-face regions and focusing on those with a high possibility to empower both general accuracy and consistency of the face detection stage, making it most fitting for our system. Algorithm 1 explains the haar cascade face detection steps.

Algorithm 1: HaarCascadeFaceDetection
Input: Video frames from YouTube Faces Database (YTF)
Output: Cropped face images

1. **Load** pre-trained Haar Cascade Frontal Face Classifier from OpenCV
2. **For** each frame **in** video:
   a. **Convert** the frame to grayscale
   b. **Detect** faces in the grayscale frame using the Haar Cascade Classifier
   c. **For** each detected face:
      i. **Extract** the region of interest (ROI) corresponding to the face
      ii. **Crop** and save the ROI for further processing
End For

Return: Cropped face images

2.2 Preprocessing

This stage is indispensable for preparing the detected face images to extract features. This stage shall standardize, normalize, and increase the input images so that the model becomes more potent in learning and generalizing from data. The following steps were done in this phase of preprocessing:

2.2.1 Image Resizing

All the detected face images were resized to a size of 224x224 pixels. Since all the images taken as inputs in this work had this same dimension, this gave uniformity of dimension to the neural network architecture, making it simple and less computationally complex, ensuring that all the input images are standardized, and making the neural network process easy.

2.2.2 Grayscale Conversion

Although the Haar Cascade Classifier is applied in grayscale images, this set is one of the preprocessing stages. That means all images are converted to grayscale. This stage helps decrease the amount of data that needs to be processed by the model because color information is irrelevant in feature extraction or classification tasks. Grayscaling will reduce the complexity of an image and focus only on the intensity values to enhance processing speed.

2.2.3 Normalization

Normalization scales the pixel values of images within a certain range, usually to the [0, 1] range. This step is important since it speeds up the convergence during training by keeping the input data at a comparable scale. For that, normalization will hold all pixel values within the desired range, hence improving the performance and stability of a neural network during training. The normalized value is calculated as follows:

$$Normalizes\ Value = \frac{Pixel\ Value}{255.0} \tag{4}$$

2.2.4 Histogram Equalization

Histogram equalization performs contrast adjustment in grayscale images and spreads the intensities' values within the whole range, thus enriching the contrast and making facial features more distinguishable. It improves the visibility of features in images under poor lighting conditions.

2.2.5 Gaussian Filters

Gaussian filters smooth out noise in images. Blurring an image with a Gaussian kernel decreases its details, a very important preprocessing step that removes random noise that may hamper the extraction of features, hence creating cleaner and more homogeneous images. The Gaussian filter is applied as follows [14]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{5}$$

where $G(x, y)$ is the Gaussian function applied to the pixel at coordinates (x, y) and σ is the standard deviation of the Gaussian kernel.

2.2.6 Geometric Distortion Correction

These techniques are required for possible image distortions, correctly representing facial features. Its techniques are based on mapping the coordinates of the distorted image into a rectified coordinate system, providing normalization in the position and shape of facial features.

2.2.7 Data Augmentation

Several augmentation techniques were used on this dataset to be very diverse and increase the model's capability to generalize better. Augmentation simulates many scenarios where the model might turn out while real-world applications are worried, like lighting conditions, facial expressions, or even face orientation. The list of augmentations used is seen below.

• Rotation: The image rotation is random, ranging within ±15 degrees to cover variations in head tilt.
• Random Zoom: Random zooming was applied to simulate different distances from the camera. It makes the model more powerful in recognizing faces at different scales.
• Horizontal Flipping: The images were flipped horizontally, which provided mirror images and helped the model generalize better on different facial orientations.
• Brightness Adjustment: Random changes in brightness were applied to simulate different lighting conditions.

2.2.8 Splitting of the Dataset

The mode of dataset division adopted in this research is in a 70:30 system where 70% of the data could be used for training and 30% for testing. This data splitting ensures that the model has enough data to learn from while providing substantial data to evaluate its performance on unseen samples.

These preprocessing steps in the pipeline ensure standardization and the enhancement of input images, making them amenable to feature extraction and further classification. Such a comprehensive preprocessing pipeline will help develop a much-robust-yet-accurate face recognition system that will deal with many issues related to the real world. Algorithm 2 explains the preprocessing steps.

Algorithm 2: Preprocessing
Input: Detected face images
Output: Preprocessed images ready for feature extraction
1. **Initialize** ImageDataGenerator for data augmentation 2. **For** each detected face image: a. **Resize** the image to 160x160 pixels b. **Convert** the image to grayscale c. **Normalize** the image pixel values to the range [0, 1] d. **Apply** histogram equalization to enhance contrast e. **Apply** Gaussian filter to remove noise f. **Correct** geometric distortions g. **Augment** the image using rotation, zoom, horizontal flipping, and brightness adjustment h. **Save** the preprocessed image

3. **End** For 4. **Split** the dataset into training (70%) and testing (30%) sets
Return: Preprocessed images (Training and testing sets)

2.3 Feature Extraction

Feature extraction is a stage wherein the preprocessed face images should be transformed into meaningful representations that can be used for classification. This paper uses the pre-trained FaceNet model to extract feature vectors from the preprocessed images. FaceNet is a deep convolutional network designed to map face images into compact Euclidean space, where distances directly correspond to a measure of face similarity.

2.3.1 Overview of FaceNet

Facenet takes face images as input and embeds them in a 128-dimension featurized space. During training, a triple loss function is employed, maintaining that the distance between the anchor and the positive image will be smaller than the distance of the anchor with the negative image by at least a specified margin. This training strategy will naturally force the model to learn a robust feature representation in which images of the same person remain close while keeping those of different persons well-separated [35].
The triplet loss function is defined as follows:

$$L = \sum_{i=1}^{N} [\left|f(x_i^a) - f\left(x_i^p\right)\right|_2^2 - |f(x_i^a) - f(x_i^n)|_2^2 + \alpha]_+ \quad (6)$$

Where:
$f(x)$ is the embedding function mapping image x into the space of features.
x_i^a, x_i^p, and x_i^n are the anchor, positive, and negative samples, respectively.
α is a margin enforced between positive and negative pairs.
$|.|_2$ denotes the L2 norm.
$|.|_+$ denotes the positive part of the argument.

2.3.2 Feature Extraction Process

Essentially, feature extraction in FaceNet happens in the following steps:
1. Load Pre-train FaceNet Model: A pre-trained model of FaceNet is loaded directly into the environment. Since it was pre-trained on a large dataset of face images, it learns a rich feature space that helps distinguish between faces.
2. Generation of Embedding: Every preprocessed face image is passed to the FaceNet model to generate a 128-dimensional embedding. These embeddings capture only the essence of faces relevant for recognition and discard other irrelevant information.
3. Embeddings Normalization: The output embedding vectors are normalized to the unit hypersphere. It is a scaling operation, ensuring that the Euclidean norm of these embeddings equals one. Normalization stabilizes feature representations and provides consistency in the distance calculations during the subsequent classification stage.
Mathematically, this normalization can be expressed as follows:

$$\hat{e} = \frac{e}{||e||_2} \quad (7)$$

Where:

e is the original embedding.
$\hat{e}$ is the normalized embedding.
$||e||_2$ is the L2 norm of the embedding.
Algorithm 3 explains the feature extraction steps.

Algorithm 3: FeatureExtraction
Input: Preprocessed face images
Output: 128-dimensional feature vectors (embeddings)
1. **Load** the pre-trained FaceNet model 2. Initialize an empty list to store feature vectors 3. **For** each preprocessed face image: a. **Expand** image dimensions to match the input shape expected by FaceNet b. **Pass** the image through the FaceNet model to generate the embedding c. **Normalize** the embedding to have a unit norm d. **Append** the normalized embedding to the list of feature vectors **End** For
Return: List of 128-dimensional feature vectors

After these steps, the preprocessed face images have been transformed into compact and meaningful feature vectors that can be successfully classified. The feature extraction step ensures that every face's essential characteristics are captured to facilitate robust face identification.

2.4 ELM Classification

The classification stage utilizes Extreme Learning Machines for fast and efficient classification of the feature vectors extracted by FaceNet. ELMs are useful because of their fast learning speed and good performance for large datasets.

2.4.1 Overview of Extreme Learning Machines (ELMs)

Extreme Learning Machines are single hidden layer feedforward neural networks recognized for their speed in training and good generalization performance. Unlike traditional neural networks, the ELM does not need iterative tuning of parameters; rather, the input weights and biases are randomly assigned, after which it analytically determines the output weight in one step. This approach greatly reduces the computational burden and accelerates the training process.
An ELM usually has three layers: an input, a single hidden, and an output layer. The major steps in training an ELM include the random initialization of input weights and biases, the computation of the hidden layer output, and determining the output weight [12].

2.4.2 ELM Training Process

1. Initialization of Input Weights and Biases: Randomly assign weights wi and biases for the bias term bibi for interconnections between the input and hidden layers. All these parameters remain constant during training.

2. Calculating the Output of the Hidden Layer: The output for the hidden layer neurons is computed using the activation function. One common choice is the sigmoid activation function. The output matrix of the hidden layer, H$\mathbb{R}\mathcal{H}$, is computed as follows:

$$H_{ij} = g(w_j . x_i + b_j) \qquad (8)$$

Where $g(.)$ is the activation function, w_j is the weight vector for the *j*-th hidden neuron, x_i is the input vector, and b_j is the bias for the *j*-th hidden neuron.

The output weights β\mathbf{\beta}β are determined by minimizing the error between the actual and predicted outputs. This is done analytically using the Moore-Penrose pseudoinverse of the hidden layer output matrix H\mathbf{H}H:

$$\beta = H^{\dagger}T \qquad (9)$$

Where T is the target matrix and $H^{\dagger}$ is the Moore-Penrose pseudoinverse of H.

2.4.3 ELM Classification Process

The classification process using ELM is as follows:

1. Input preparation: It feeds features ensured from 128-dimensional feature vectors, representing FaceNet model representations to the ELM_Classifier.
2. Computation of Hidden Layer: The output should be calculated using the weights and biases randomly initialized using an activation function of one's choice.
3. Output weight computation: The output weights are computed by Moore–Penrose pseudoinverse method.
4. Classification: The trained ELM model classifies the input feature vectors. The class label depends on the output of the ELM classifier. Algorithm 4 explains the ELM classification steps.

Algorithm 4: ELMClassification
Input: 128-dimensional feature vectors (embeddings), corresponding labels
Output: Trained ELM model, predicted labels
1. **Initialize** random weights and biases for the hidden layer 2. Compute the hidden layer output matrix H using the activation function 3. Calculate the output weights beta using the Moore-Penrose pseudoinverse beta = H^dagger * T 4. **For** each input feature vector: a. Compute the hidden layer output using the fixed weights and biases b. Calculate the output using the output weights beta c. Determine the class label based on the output **End** For
Return: Trained ELM model, predicted labels

Face images can be efficiently classified only through extreme learning machines. This way, the pace of the classification of face images becomes pretty fast with accurate results. Hence, this method is suitable for large-scale face identification tasks due to its fast training speed and good generalization performance.

2.5 Model Compression

The phase of model compression is for optimizing the model FaceNet+ELM to work on devices with limited computational power by pruning and quantization methods, drastically reducing the model size and leading to better inference speed without much loss in accuracy.

2.5.1 Pruning

Pruning is eliminating excess weights or neurons in a neural network. Model complexity decreases when parameters are removed close to zero in their values on the model's output. The major steps taken for pruning are:

1. Identification of Redundant Parameters: Observe the trained model and locate those weights and neurons that contribute minimally to the model's prediction, normally about the magnitude of the weights; the smaller they are, the less important they are considered.
2. Remove Redundant Parameters: Based on this model, remove the identified weights and neurons. This step reduces the number of parameters and the overall size of the network.
3. Fine Tuning: After pruning the model, there will be a loss in accuracy, which fine-tuning will recover. The fine-tuned dragooned model on the original dataset for some epochs now has a reduced learning rate.

Mathematically, letting W be the weights of the neural network, the pruning process can be written as follows:

$$W' = \{w \in W \,||w| > \tau\} \qquad (10)$$

Where τ is a threshold below which weights are considered redundant and removed.

2.5.2 Quantization

Quantization reduces the precision of weights and biases from 32-bit floating-point down to lower bit-width representations, such as 8-bit integers, drastically reducing model size while increasing inference speed significantly on hardware optimized for lower-precision arithmetic.

In particular, the basic steps of quantization include:

1. Static Quantization: Compute the quantization parameters with the training dataset before actual quantization. The computed set of parameters is then used to convert weights and activations from floating-point to integer format.
2. Dynamic Quantization: Weights are quantized only during inference; however, activations are dynamic based on the input data. Thus, it is more flexible and adapts to different input data distributions.
3. Quantization-Aware Training: This builds quantization into the training regimen itself, minimizing loss of accuracy. This causal and reverse pass is a process that simulates the effects of quantization and follows it up with model parameter updates.

Mathematically, quantization for a weight www can be expressed as follows:

$$w_q = round\left(\frac{w}{s}\right) + z \qquad (11)$$

where:

w_q is the quantized weight,

s is the scale factor,

z is the zero-point. Algorithm 5 demonstrates the model compression steps.

Algorithm 5: ModelCompression
Input: Trained FaceNet+ELM model
Output: Compressed and optimized model

1. Pruning: a. Identify redundant weights and neurons in the FaceNet model b. Remove the identified weights and neurons c. Fine-tune the pruned model on the original dataset 2. Quantization: a. Perform static quantization to pre-calculate quantization parameters b. Apply dynamic quantization during inference for activations c. (Optional) Perform quantization-aware training to maintain accuracy
Return: Compressed and optimized model

In this line of thought, it is highly envisioned that applying such model compression techniques to the FaceNet+ELM model will ensure that it is fit for deployment on devices with limited computational power. This method has high accuracy, with a notable reduction in model size and inference speed enhancement, making it suitable for real-time applications.

3. Performance Metrics

Several metrics might yield an all-rounded understanding of the accuracy and robustness of a face identification system: accuracy, precision, recall, and the F1-score. All these metrics consider model performance differently, especially in classification tasks.

3.1 Accuracy

One of the simplest metrics could be accuracy, which evaluates the proportion of correctly predicted instances against total instances and gives a general idea about how good the model is but can mislead in cases when there is class imbalance. The accuracy is calculated as follows:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \tag{12}$$

Where:

TP : True Positives (correctly predicted positive instances)
TN: True Negatives (correctly predicted negative instances)
FP: False Positives (incorrectly predicted positive instances)
FN: False Negatives (incorrectly predicted negative instances)

3.2 Precision

Precision, also called Positive Predictive Value, is the number of predicted positives that are positives. In mathematical terms, it is a ratio showing the number of correctly predicted positive examples against all the instances predicted as positive, making precision a useful metric when a high cost is associated with false positives. The precision is calculated as follows:

$$Precision = \frac{TP}{(TP + FP)} \tag{13}$$

3.3 Recall

Recall sometimes referred to as Sensitivity or True Positive Rate, comes to be the proportion of predicted true positive instances against all actual positive cases is how well it can detect positive instances; hence, this measure comes in handy when the cost of missing a positive instance is high. The recall is calculated as follows:

$$Recall = \frac{TP}{(TP+FN)} \tag{14}$$

F1-Score

The F1-score is the harmonic mean of precision and recall, providing a single metric that would balance both. It becomes very useful in cases of class imbalance and when a balance between precision and recall is needed. The F1-Score is calculated as follows:

$$F1 - score = \frac{Precision * Recall}{(Precision + Recall)} \tag{15}$$

These metrics need to be applied in order to investigate the performance of a face identification system on different levels:

1. Training Phase: Some metrics are checked during training to ensure the model learns from the data.
2. Validation Phase: These metrics help tune hyperparameters and select models based on the performance of validation sets.
3. Testing Phase: The model's performance against unknown data is measured through these metrics to estimate generalizability/robustness.

These metrics can be used to evaluate the face identification system—a comprehensive, descriptive performance measure on the train data and generalizability toward new, unseen data. A thorough evaluation will help build a robust, reliable face identification system to be applied in the real world.

4. Model Implementation

In summary, the model for real-time face identification is implemented through video-feed capture, face detection in the frames, preprocessing of the detected faces, feature extraction, and face classification using a pre-trained model. This implementation results in lightweight solutions easily deployed on low-compute devices such as laptops or low-power devices.

4.1 Capturing Video Feed

Implementation begins by generating a video capture object, accessing the webcam, and then starting the capturing of real-time video feeds. It is realized through a continuous loop that captures frames in a video stream.

4.2 Seamless Frame Capture and Face Detection

In this loop, each frame of the video feed is treated to detect faces using a face detection algorithm that identifies frame regions containing faces. After extraction, the detected faces are further processed for identification.

4.3 Processing Detected Faces

Size checks are implemented for every detected face to filter out the smaller detections, which are less likely to be valid. In the frame, rectangles will be drawn around faces that pass these size criteria. After all that, cropped face regions earlier get resized into a standard size that is compatible with the model used for feature extraction.

4.4 Preprocessing

These images of faces, cropped, are then preprocessed to be fed into the model for extracting features. It involves changing the image into an array, reshaping, and normalizing techniques to prepare it for analysis.

4.5 Feature Extraction and Classification

These preprocessed images of faces are then input into a pretrained model, such as FaceNet, for generating 128-dimensional feature vectors. Such vectors can be considered the features unique to each face. After that, these feature vectors will be fed into an Extreme Learning Machine classifier, which predicts face identity based on extracted features.
If the confidence of the prediction of the classifier is less than a threshold, the face will be annotated as 'UNKNOWN' to convey low confidence in the match. Such a threshold can properly deal with persons whose faces do not fit any known individuals present in the training data.

4.6 Displaying Results

It also shows the predicted identity or 'UNKNOWN' at the location of the detected face in the video frame. The program's runtime is also kept track of and displayed, thus ensuring efficient system performance.

4.7 Testing and Deployment

It is implemented lightweight to accomplish devices with minimal computational powers. Some testing involves the laptop's web camera to ensure the application works acceptably under resource-constrained conditions. Its ability to act on such devices demonstrates how the system can be widely used among many real-time applications.
Applied to a real-time, face-identification system, experimental results show that it can maintain high accuracy and efficiency and is suitable for running on devices with limited computational ability. Robust face detection with efficient preprocessing, accurate feature extraction, and effective classification ensures real-world system performance.

5. Discussion of Results

Results for the face identification system proposed herein highlight important lessons concerning its performance, robustness, and suitability for real-world applications. The following section presents these results with a detailed discussion of the implications and overall effectiveness. Table 2 shows the results of the performance metrics for our proposed method.

5.1 Accurate and Reliable

It achieved an accuracy of 99.1%, indicating that it correctly identified nearly all the faces. This very high accuracy demonstrates the effectiveness of the combined approach with FaceNet and Extreme Learning Machine. FaceNet is good at providing high-quality feature representations, while ELM ensures efficient classification. As such, the system will work well in most varied conditions.

That high accuracy is especially palpable given the complexity of face identification tasks. Illumination, facial expressions, and occlusions can frequently degrade performance. However, the system's robustness against these challenges, evidenced by its high accuracy, suggests that it could be reliably fielded in regimes ranging from security systems to mobile devices.

5.2 Balance of Precision and Recall

A system precision of 98.5%, matched with a recall of 97.8%, underlines a very balanced performance by the system regarding both measures. Precision is a measure that explicitly looks at the ratio of true positive instances correlated with total positive instances predicted by the system, hence measuring how precisely this system filters out false positives. A precision of 98.5 percent thus means that most of the faces identified are correct—relevant in applications concerning security and surveillance, wherein huge results occur in case of false alarms.

Recall measures the ability of a system to detect all real positive instances. Therefore, a recall of 97.8% will mean that it will almost detect all true faces in a dataset. During scenarios like law enforcement and missing person identification, where being unable to identify a face might mean everything, this high recall is very important.

The F1-score debuts at a value of 0.981, combining precision and recall into one metric to further emphasize balanced performance, which means the system has kept high accuracy on both metrics and, therefore, identifies reliably and consistently.

5.3 Efficiency and Real-Time Performance

Most importantly, the average runtime of the system per frame is approximately 49 milliseconds, thus proving its suitability for real-time applications. Real-time performance could be critically required for timely identification in some special scenarios, such as live surveillance or interactive systems. This lightweight nature of the Haar Cascade Classifier for preliminary face detection and the fast processing capabilities of the ELM classifier contribute mostly to the efficient runtime.

This efficiency is particularly notable given the system's high accuracy and robust performance. It demonstrates that the system can operate effectively on devices with limited computational power, making it a viable solution for many applications, including those with stringent real-time requirements.

5.4 Low-Power Devices Suitability

A standard laptop with a built-in webcam was tested, showing that the system can work well on low/power devices, which is important because most real-world applications, especially in mobile and embedded systems, demand solutions highly independent of high-end computational resources. It will still be capable of delivering high performance in such devices, which greatly opens the possibilities amongst other uses for portable security devices to consumer electronics.

5.5 Comparative Performance

The proposed system demonstrates superior performance across several metrics compared to existing face identification systems. Traditional methods, such as those relying solely on manual feature extraction or support vector machines (SVMs), often struggle with large datasets and real-time requirements. In comparison, FaceNet combined with ELM handled large datasets efficiently and met demands for real-time processing. In this work, ELMs enhance performance by providing fast and efficient classification mechanisms that reduce the sometimes enormous computational load typically associated with deep learning models.

Table 2
Summary of Results of the performance metrics for our proposed method

Metric	Value
Accuracy	99.1%
Precision	98.5%
Recall	97.8%
F1-Score	98.1%
Average Runtime	49 ms

These results demonstrate that the proposed face identification system is highly accurate and balanced in performance and can support real-time applications on low-power devices. Its robust performance in diversified conditions and efficient processing capabilities make it a strong candidate for practical scenarios.

5.6 Comparison with Recent Works

In contrast to other recently proposed face identification systems, the proposed system has some prominent strengths, even if it incorporates FaceNet for feature extraction and Extreme Learning Machines in classification. Besides, it adds evidential weight to support that this system is suitable for application benefits because of its high accuracy, efficiency, and suitability for deployment on low-power devices. Following is a summarization of some recent works in face identification and their findings to have a clear comparison with our proposed method:

1. DeepFaceLab

DeepFaceLab [36] was successful in face swapping and deepfake generation but aimed at face recognition. It deploys the most advanced techniques of deep learning that help it achieve high accuracy. However, It is computationally intensive and requires enormous processing power and memory, making it unsuitable for low-power devices.

2. Google's FaceNet

FaceNet [35] is a deep learning model of Google for facial recognition and verification, mapping face images to a 128-dimensional Euclidean space. Although robust and accurate, their implementation in low-power devices is challenging due to the high computational demands.

3. Dlib ResNet

ResNet-based face recognition offers accuracy and computational efficiency through a deep residual network that extracts facial features [37]. However, the size and computational needs of the model remain significant enough for deployment on resource-constrained devices.

4. InsightFace

InsightFace [38] is an open-source 2D and 3D deep face analysis toolkit with many face recognition models achieving high accuracy on standard benchmarks. It is optimized for performance but requires substantial computational resources, limiting its suitability for real-time applications on low-power devices.

5. ArcFace
ArcFace [39] is designed to improve face recognition efficiency and lessen extracted features by applying an additive angular margin loss that ensures increased discriminative power of features. However, the technique requires very high computational power and can thus only be applied to high-power devices. Table 3 compares our proposed method's results with the state-of-the-art method's.

Table 3
Comparison of the proposed method's results with the state-of-the-art method results

Metric	Proposed System (FaceNet+ELM)	DeepFaceLab (2021)	FaceNet (2020)	Dlib's ResNet (2021)	InsightFace (2022)	ArcFace (2020)
Accuracy	99.1%	98.5%	99.2%	98.7%	99.3%	99.4%
Precision	98.5%	97.8%	98.9%	98.5%	99.1%	99.2%
Recall	97.8%	97.3%	98.7%	97.9%	98.8%	98.9%
F1-Score	98.1%	97.5%	98.8%	98.2%	98.9%	99.0%
Average Runtime	49 ms	120 ms	100 ms	90 ms	85 ms	80 ms
Low-Power Suitability	High	Low	Moderate	Moderate	Low	Moderate

The proposed FaceNet+ELM system has the following benefits over other recent face identification systems. On the accuracy front, FaceNet+ELM is very competitive with state-of-the-art models, including InsightFace and ArcFace, which realize slightly higher accuracy but with higher computational resources. The balance between precision and recall, hence a high F1-score for the proposed system, reliably performs to minimize false positives and false negatives.

One of the major benefits of the proposed system lies in its efficiency, which has an average run time of about 49 milliseconds per frame; this makes it suitable even for real-time applications. More importantly, this efficiency can be very effective in deploying low-power devices, a region where many state-of-the-art models usually fail due to their high computational demands. Testing the system further gives evidence, on a standard laptop webcam, of how practical its application could turn out under resource-constrained conditions.

These strengths bring with them several areas for improvement. We can further include more challenging conditions within dataset diversity and explore state-of-the-art model compression for better robustness, improving system performance. Developing mechanisms to identify new faces not present in the training set and test them under highly dynamic environmental conditions will enhance their adaptability and reliability.

In brief, a proposed system of FaceNet+ELM will provide a strong and efficient solution for face identification, attaining high accuracy with real-time performance suitable for low-power devices. Although enhancement opportunities still exist, the system's finely balanced performance metrics coupled with lightweight execution make this system competitive compared to more resource-intensive models when *providing a practical solution for many real-world applications.*

5.7 Real-World Data Testing

To validate the performance and robustness of the face identification system proposed in this study, a real-world dataset was collected from 20 individuals. Each individual contributed only five images that captured facial expressions ranging from neutral to extremely happy or sad and lighting from dark to bright. The following subsections detail data collection and preprocessing steps and results commensurate with the high performance noted in laboratory-controlled experiments.

5.7.1 Data Collection

The real-world dataset was created using a standard notebook webcam. Each of the 20 contributors was asked to provide five images, making for 100 images. The images were captured under different conditions to capture natural variability in lighting, backgrounds, and facial expressions. This diversity ensured that the dataset was representative of typical usage scenarios.

5.7.2 Preprocessing

The real-world dataset was processed by the same pipeline used with the controlled experiments:
1. Image Resizing: All images were resized to a dimension of 160x160 pixels.
2. Grayscale Conversion: The images have been converted into grayscale because it is just about the intensity values.
3. Normalization: Pixels were normalized between 0 and 1.
4. Histogram equalization: This enhanced the contrast to distinguish the facial features more.
5. Gaussian Filtering: Noise was removed to raise an image's quality level.
6. Geometric distortion correction: All types of distortion were corrected to preserve the actual appearance of facial features.

5.7.3 Feature Extraction and Classification

These preprocessed images were then fed into the pre-trained FaceNet model to generate 128-dimensional vectors of features to be classified by the trained ELM model. The accuracy, precision, recall, and F1-score obtained for this real-life dataset were as high as observed in controlled experiments.
The real-world dataset performance metrics are summarized as follows:
• Accuracy: 99.0%
• Precision: 98.7%
• Recall: 97.9%
• F1-Score: 98.3%
•Average Runtime: 52 milliseconds per frame

These results indicate that the proposed face identification system can maintain its top performance with only a small, real-world, varied dataset. The robustness and applicability of a particular system in real life are testified by its ability to identify a person with normally limited training data uniquely.

The results obtained on the real-world dataset further strengthen the robustness and effectiveness of the system in a practical scenario. Provided that it had only very few images per person and the natural variability in the data, high-performance metrics were consistent for this system. This consistency underlines its ability to generalize well from limited training data, making it very suitable for applications with large coverage datasets.

Further, the runtime performance of the system was good enough, even for real data, making the solution realistic to run on low-power devices. This high efficiency, combined with accuracy, precision, recall, and F1-score makes the proposed system a very competitive solution for most face identification applications—security, surveillance, and personal authentication.

6. Conclusion

This paper proposes a significant challenge in developing an efficient and robust face identification system by integrating FaceNet for feature extraction with Extreme Learning Machines for classification. It returned very impressive performance metrics, an accuracy of 99.1 percent, a precision of 98.5 percent, a recall of 97.8 percent, and an F1-score of 98.1 percent. The results prove its effectiveness in accurately identifying human faces under various conditions. In terms of runtime, the system is efficient and takes 49 milliseconds per frame; under the proposed framework, it could support real-time applications on low-power devices. Experiments were conducted on a real-world dataset with data from only 20 subjects, having just five images each, proving the robustness and high performance of the system. However, there are several limitations to this research. The dataset diversity can be improved by introducing challenging circumstances like light-extreme situations, facial occlusion variation, and expression varieties. Model compression beyond pruning and quantization can be done to optimize the system. It also lacks any mechanism to identify unknown faces and has not been tested under high environment dynamics.

This work will help disagree with the notion that high-performance face identification systems cannot be built accurately and efficiently, even on low-power devices. Results show potential for wide security, surveillance, and personal authentication applications. Future research will focus on increased dataset variety, advanced model compression techniques, and testing in more dynamic environments. This change will make the applicability of this system greater and more versatile for face identification challenges in real life. By addressing these areas, we will enable more adaptable and efficient face identification systems to meet the requirements encountered in several practical applications, which will, in turn, help drive progress toward better security and superior technology.

Acknowledgement

This research was not funded by any grant

References

[1] Dobs, K., J. Martinez, A. J. E. Kell, and N. Kanwisher. 2022. "Brain-like Functional Specialization Emerges Spontaneously in Deep Neural Networks." *Science Advances* 8, no. 11. https://doi.org/10.1126/sciadv.abl8913.

[2] Parvaiz, A., M. A. Khalid, R. Zafar, H. Ameer, M. Ali, and M. M. Fraz. 2023. "Vision Transformers in Medical Computer Vision—A Contemplative Retrospection." *Engineering Applications of Artificial Intelligence*. https://doi.org/10.1016/j.engappai.2023.106126.

[3] Chen, Y. L., C. Y. Wu, S. C. Li, T. M. Yu, and S. P. Yu. 2023. "Effect of Mask Coverage on Face Identification in Taiwanese Men and Women." *Frontiers in Psychology* 14. https://doi.org/10.3389/fpsyg.2023.1082376.

[4] Mohammed, A., and R. Kora. 2023. "A Comprehensive Review on Ensemble Deep Learning: Opportunities and Challenges." *Journal of King Saud University – Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2023.01.014.

[5] Salehi, A. W., et al. 2023. "A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope." *Sustainability* 15, no. 7. https://doi.org/10.3390/su15075930.

[6] Salama, R., and M. Nour. 2023. "Security Technologies Using Facial Recognition." *Global Journal of Computer Sciences: Theory and Research* 13, no. 1. https://doi.org/10.18844/gjcs.v13i1.8294.

[7] Saraswati, N. M., R. C. Sigitta Hariyono, and D. Chandra. 2023. "Face Recognition Menggunakan Metode Haar Cascade Classifier dan Local Binary Pattern Histogram." *Jurnal Media Elektrik* 20, no. 3. https://doi.org/10.59562/metrik.v20i3.50491.
[8] Afrin, S., M. Tasnim, and Md. R. Islam. 2023. "Human Face Recognition Using Eigen Vector-Based Recognition System." *International Journal of Research and Scientific Innovation* X, no. VI. https://doi.org/10.51244/ijrsi.2023.10617.
[9] Zhang, H., and Z. Yang. 2023. "Biometric Authentication and Correlation Analysis Based on CNN-SRU Hybrid Neural Network Model." *Computational Intelligence and Neuroscience* 2023. https://doi.org/10.1155/2023/8389193.
[10] He, G., and Y. Jiang. 2022. "Real-Time Face Recognition Using SVM, MLP and CNN." In *Proceedings of the 2022 International Conference on Big Data, Information and Computer Network (BDICN)*. https://doi.org/10.1109/BDICN55575.2022.00149.
[11] Dang, T. V., and H. L. Tran. 2023. "A Secured, Multilevel Face Recognition Based on Head Pose Estimation, MTCNN and FaceNet." *Journal of Robotics and Control* 4, no. 4. https://doi.org/10.18196/jrc.v4i4.18780.
[12] Wang, J., S. Lu, S. H. Wang, and Y. D. Zhang. 2022. "A Review on Extreme Learning Machine." *Multimedia Tools and Applications* 81, no. 29. https://doi.org/10.1007/s11042-021-11007-7.
[13] Dyke, R. M., and K. Hormann. 2023. "Histogram Equalization Using a Selective Filter." *The Visual Computer* 39, no. 12. https://doi.org/10.1007/s00371-022-02723-8.
[14] Kusrini, M. R. A. Yudianto, and H. Al Fatta. 2022. "The Effect of Gaussian Filter and Data Preprocessing on the Classification of Punakawan Puppet Images with the Convolutional Neural Network Algorithm." *International Journal of Electrical and Computer Engineering* 12, no. 4. https://doi.org/10.11591/ijece.v12i4.pp3752-3761.
[15] Schaefer, C. J. S., P. Taheri, M. Horeni, and S. Joshi. 2023. "The Hardware Impact of Quantization and Pruning for Weights in Spiking Neural Networks." *IEEE Transactions on Circuits and Systems II: Express Briefs* 70, no. 5. https://doi.org/10.1109/TCSII.2023.3260701.
[16] Yaakub, N. D. A., et al. 2021. "Class Attendance System Using Viola-Jones Algorithm and Principal Component Analysis." In *Journal of Physics: Conference Series*. https://doi.org/10.1088/1742-6596/1962/1/012053.
[17] Yan, H., Z. Hu, and Q. Cui. 2023. "Study on the Influence of PCA Pre-Treatment on Pig Face Identification with SVM." *INMATEH – Agricultural Engineering* 69, no. 1. https://doi.org/10.35633/inmateh-69-09.
[18] Lumini, A., L. Nanni, L. Scattolaro, and G. Maguolo. 2021. "Image Orientation Detection by Ensembles of Stochastic CNNs." *Machine Learning with Applications* 6. https://doi.org/10.1016/j.mlwa.2021.100090.
[19] Xu, Y., S. Martínez-Fernández, M. Martinez, and X. Franch. 2023. "Energy Efficiency of Training Neural Network Architectures: An Empirical Study." In *Proceedings of the Annual Hawaii International Conference on System Sciences*.
[20] Huang, Z. Y., et al. 2023. "A Study on Computer Vision for Facial Emotion Recognition." *Scientific Reports* 13, no. 1. https://doi.org/10.1038/s41598-023-35446-4.
[21] Pangelinan, G., et al. 2023. "Exploring Causes of Demographic Variations in Face Recognition Accuracy." April.
[22] Cherepanova, V., et al. 2023. "A Deep Dive into Dataset Imbalance and Bias in Face Identification." In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*. https://doi.org/10.1145/3600211.3604691.
[23] Kolf, J. N., et al. 2023. "EFaR 2023: Efficient Face Recognition Competition." In *2023 IEEE International Joint Conference on Biometrics (IJCB)*. https://doi.org/10.1109/IJCB57857.2023.10448917.
[24] He, R., J. Cao, L. Song, Z. Sun, and T. Tan. 2020. "Adversarial Cross-Spectral Face Completion for NIR-VIS Face Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, no. 5. https://doi.org/10.1109/TPAMI.2019.2961900.
[25] Adjabi, I., A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed. 2020. "Past, Present, and Future of Face Recognition: A Review." *Electronics* 9, no. 8. https://doi.org/10.3390/electronics9081188.
[26] Nouyed, I., and N. Zhang. 2023. "Face Image Quality Enhancement Study for Face Recognition." July.
[27] Oloyede, M. O., G. P. Hancke, and H. C. Myburgh. 2020. "A Review on Face Recognition Systems: Recent Approaches and Challenges." *Multimedia Tools and Applications* 79, no. 37–38. https://doi.org/10.1007/s11042-020-09261-2.
[28] Guo, Y., L. Zhang, Y. Hu, X. He, and J. Gao. 2016. "MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition." In *Lecture Notes in Computer Science*. https://doi.org/10.1007/978-3-319-46487-9_6.
[29] Hanmandlu, M., D. Gupta, and S. Vasikarla. 2013. "Face Recognition Using Elastic Bunch Graph Matching." In *Proceedings – Applied Imagery Pattern Recognition Workshop*. https://doi.org/10.1109/AIPR.2013.6749338.
[30] Hu, W., W. Yan, and H. Hu. 2022. "Dual Face Alignment Learning Network for NIR-VIS Face Recognition." *IEEE Transactions on Circuits and Systems for Video Technology* 32, no. 4. https://doi.org/10.1109/TCSVT.2021.3081514.
[31] Kakadiaris, I. A., et al. 2017. "3D–2D Face Recognition with Pose and Illumination Normalization." *Computer Vision and Image Understanding* 154. https://doi.org/10.1016/j.cviu.2016.04.012.

[32] Jin, T., Z. Liu, Z. Yu, X. Min, and L. Li. 2017. "Locality Preserving Collaborative Representation for Face Recognition." *Neural Processing Letters* 45, no. 3. https://doi.org/10.1007/s11063-016-9558-2.

[33] Lu, T., Q. Zhou, W. Fang, and Y. Zhang. 2021. "Discriminative Metric Learning for Face Verification Using Enhanced Siamese Neural Network." *Multimedia Tools and Applications* 80, no. 6. https://doi.org/10.1007/s11042-020-09784-8.

[34] Pandey, A., D. Choudhary, R. Agarwal, and T. Shrivastava. 2022. "Face Detection Using Haar Cascade Classifier." *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4157631.

[35] Schroff, F., D. Kalenichenko, and J. Philbin. 2015. "FaceNet: A Unified Embedding for Face Recognition and Clustering." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2015.7298682.

[36] Liu, K., I. Perov, D. Gao, N. Chervoniy, W. Zhou, and W. Zhang. 2023. "DeepFaceLab: Integrated, Flexible and Extensible Face-Swapping Framework." *Pattern Recognition* 141. https://doi.org/10.1016/j.patcog.2023.109628.

[37] King, D. E. 2009. "Dlib-ML: A Machine Learning Toolkit." *Journal of Machine Learning Research* 10.

[38] A. N., and K. Anusudha. 2024. "Real-Time Face Recognition System Based on YOLO and InsightFace." *Multimedia Tools and Applications* 83, no. 11. https://doi.org/10.1007/s11042-023-16831-7.

[39] Deng, J., J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou. 2022. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, no. 10. https://doi.org/10.1109/TPAMI.2021.3087709.