Soft Computing Fusion with Applications

www.scfa.reapress.com

Soft. Comput. Fusion. Appl. Vol. 2, No. 4 (2025) 233–243.

Paper Type: Original Article

A Deep Learning–Based Framework for Feature Extraction and Facial Verification

Mingyue Wang[1,*], Natalja Osintsev[2]

[1] School of Computer and Information, Lanzhou University of Technology, Gansu, China; 1043432989@qq.com.
[2] Fraunhofer-Institut für Holzforschung Wilhelm-Klauditz Institut WKI, Bienroder Weg 54 E, Brunswick, Germany; n.osintsev@gmail.com.

Citation:

Received: 17 April 2025 *Revised: 10 July 2025* *Accepted: 03 September 2025*	Wang, M., & Osintsev, N. (2025). A deep learning–based framework for feature extraction and facial verification. *Soft computing fusion with applications*, *2*(4), 233-243.

Abstract

With an emphasis on how Convolutional Neural Networks (CNNs) improve accuracy, adaptability, and efficiency over conventional techniques, this study investigates the incorporation of deep learning techniques in facial recognition. The paper highlights the deep learning process by describing procedures, including face identification, alignment, feature extraction, and recognition. CNNs' ability to derive intricate patterns from unprocessed image data is one of their main advantages; this enables reliable feature extraction and precise detection even in situations with changing illumination, attitude, and occlusion. Along with discussions of exciting future advancements meant to enhance fairness, robustness, and privacy preservation within facial recognition systems, challenges such as data bias, privacy problems, and adversarial susceptibility are highlighted.

Keywords: Convolutional neural networks, Deep learning in facial recognition, Feature extraction, Image data processing, Privacy preservation, Future advancements.

1 | Introduction

Facial recognition technology has become vital to various applications such as access control, security monitoring, and user verification [1]. This technology analyzes facial data from images or videos and matches it with existing records to verify a person's identity. The facial recognition process typically consists of several steps: detecting the face, aligning it, extracting features, and identifying the individual. Each step is essential for achieving real-time recognition with high accuracy and reliability. Recent advancements in deep learning, mainly through Convolutional Neural Networks (CNNs), have significantly enhanced the performance of facial recognition systems [2]. CNNs automatically extract high-level, relevant features from facial images, recognizing complex patterns that conventional methods, like Haar cascades or the Viola-Jones algorithm, may struggle with [3]. This automatic feature extraction allows CNNs to adjust to real-world differences, such

Corresponding Author: 1043432989@qq.com
https://doi.org/10.22105/scfa.v2i4.79

as variations in lighting, facial angles, and obstructions, thereby improving the accuracy and adaptability of facial recognition systems.

Deep learning has also enhanced key aspects of facial recognition [4]. For instance, CNNs can accurately detect and align faces, ensuring a consistent orientation for optimal feature extraction and comparison. During feature extraction, CNNs have demonstrated their ability to produce highly compact and distinctive feature vectors that effectively represent unique facial characteristics. This skill is vital for differentiating between individuals with minimal error.

Though these advancements are significant, challenges remain regarding data bias, privacy issues, and susceptibility to adversarial attacks. Biometric systems like facial recognition inherently involve sensitive information, making robust privacy measures essential. Additionally, problems related to data bias can result in varying performance across different demographic groups. Adversarial attacks, which subtly modify images to confuse the model, also present security threats.

This study examines the utilization of CNNs in facial recognition, investigating their advantages and challenges [5]. With ongoing research and development, CNN-based facial recognition has the potential to realize more resilient, equitable, and privacy-conscious implementations in various settings.

2 | Literature Review

Facial recognition has become a pivotal technology in many modern applications, from security and surveillance to personal device authentication and social media [6]. Over the years, researchers have developed various methods for recognizing human faces, progressing from statistical models and traditional machine learning algorithms to the current dominance of deep learning techniques.

Traditional methods for facial recognition

Early facial recognition systems relied heavily on hand-crafted features, where techniques like Eigenface Fisher faces were widely used. Eigenfaces, introduced in the 1990s, utilized Principal Component Analysis (PCA) to represent facial images as linear combinations of principal components. This approach effectively reduced the dimensionality of face images, enabling faster processing, but suffered from sensitivity to lighting, pose, and expression variation [1].

Fisherfaces, on the other hand, employed Linear Discriminant Analysis (LDA) to maximize the between-class variance and minimize within-class variance, offering more robustness against illumination changes. Despite their success in controlled environments, these techniques were limited to complex, real-world scenarios [2].

The rise of deep learning and convolutional neural networks

The introduction of CNNs transformed the domain of facial recognition by enabling automated feature extraction and the learning of hierarchical representations directly from data [3]. CNNs, made up of layers that perform convolution and pooling operations, can capture complex patterns in images, which makes them particularly powerful for visual recognition applications. Notable architectures such as AlexNet, VGGNet, and ResNet showcased CNNs' capability in large-scale image classification tasks and laid the groundwork for later facial recognition models.

One of the initial deep learning models specifically created for facial recognition was DeepFace, which Facebook developed. DeepFace employed a nine-layer deep neural network to reach human-level accuracy in identifying faces, representing a breakthrough. Google's FaceNet took the field further by introducing a triplet loss function that organizes facial images into a compact Euclidean space, positioning similar faces closer together and placing dissimilar ones further apart [3].

FaceNet achieved impressive accuracy by reducing the distance between the embeddings of the same person while increasing the distance between embeddings of different individuals [5].

Another significant advancement in facial recognition is the implementation of Siamese Neural Networks (SNNs). SNNs are especially effective for few and one-shot learning tasks, where distinguishing between identities with minimal examples is necessary. By training the network on pairs of images, SNNs learn to generate a similarity score between them, making them suitable for facial verification tasks. The contrastive loss or triplet loss functions applied in SNNs allow the model to develop discriminative embeddings, facilitating effective image comparisons even when training data is limited [6].

Open computer vision for image processing

For this research, Open Source Computer Vision library (OpenCV) is essential, particularly for preparing and improving facial images. With more than 2,500 efficient algorithms designed for various vision-related tasks, such as image editing, object recognition, and face detection, this library offers an open-source computer vision and machine learning solution. OpenCV is used in this project for several reasons, including:

I. Face detection: OpenCV's Cascade classifier or dnn module finds faces in an image before putting it into the neural network, ensuring that only the faces are processed.

II. Image augmentation: To add variation to the training dataset, OpenCV implements techniques like rotation, scaling, and flipping. Mimicking fluctuations that the model could experience in practical applications improves the model's capacity for generalization.

III. Image preprocessing: To standardize the input for the neural network, OpenCV makes it easier to resize, normalize, and convert images to greyscale.

OpenCV, for instance, can identify a face in an image, trim it to highlight facial features, and then scale it to the neural network's specifications for this case, which would be 250 × 250 pixels [7], [8].

3 | Basic Concepts

Tensorflow and keras for deep learning

In this project, a facial recognition model is developed, trained, and launched using TensorFlow and Keras. TensorFlow, an open-source deep learning library created by Google, provides a comprehensive platform for building and scaling machine learning models. Known for its flexibility and ability to handle large datasets, TensorFlow has become a key resource in various research and industrial deep learning uses. It supports CPU and GPU acceleration, making it suitable for training demanding models like CNNs and SNNs. Keras, a high-level neural network API built on TensorFlow, simplifies defining and training deep learning models. With its user-friendly APIs, Keras allows developers to prototype and assess models quickly without dealing with the complexities of lower-level implementations.

In this project, Keras components such as Conv2D, Dense, Input, Flatten, and MaxPooling2D are employed to define the architecture of the Siamese Network. For instance:

I. Conv2D layers implement convolutional operations, allowing the network to extract spatial hierarchies in facial images.

II. MaxPooling2D layers help reduce the spatial dimensions of the feature maps, retaining the most important features while reducing computational complexity.

III. Dense layers provide fully connected networks that further abstract features extracted by the convolutional layers.

Using Keras with TensorFlow provides an efficient pipeline for model development, where Keras handles the model building, while TensorFlow's backend enables optimization and deployment across multiple devices.

Comparative analysis with exciting works

Several recent works have combined TensorFlow, Keras, and OpenCV in the context of facial recognition. For instance, Luttrell et al. [9] utilized TensorFlow and Keras to implement a facial recognition system that leveraged CNN layers for feature extraction. OpenCV handles real time face detection and image preprocessing. Similarly, the researchers employed a SNN architecture built in Keras and TensorFlow for a one-shot learning task, where OpenCV was used to augment the dataset with different lighting conditions and poses. These studies have demonstrated the effectiveness of using these libraries together, particularly in applications where computational efficiency and ease of implementation are essential.

4 | Proposed Frameworks

The SNN architecture used in this project is built to analyze pairs of facial images and assess their similarity. This process is performed by directing each image from a pair through two identical subnetworks that share weights, which extract high-level features from each image. The resulting feature vectors are then compared to generate a similarity score. Below is a detailed breakdown of the architecture by layer.

4.1 | Structure of the Siamese Neural Network for Face Recognition

The SNN architecture used in this project is built to analyze pairs of facial images and assess their similarity. This process is performed by directing each image from a pair through two identical subnetworks that share weights, which extract high-level features from each image. The resulting feature vectors are then compared to generate a similarity score. Below is a detailed breakdown of the architecture by layer:

Input layer

The network accepts an input image measuring 105 × 105 pixels, representing a grayscale facial image. This size allows for adequate detail of facial features while keeping computational demands reasonable.

First convolutional layer

I. Layer details: Convolutional layer featuring 64 filters of size *10 × 10.

II. Activation: ReLU (Rectified linear unit).

III. Output: The dimensions of the resulting feature map are *96 × 96 with 64 feature maps.

IV. Description: This layer identifies basic features like edges and textures, which are vital for differentiating facial traits.

First max-pooling layer

I. Pooling size: *2 × 2.

II. Output: The feature map is condensed to *48 × 48 with 64 feature maps.

III. Description: Max-pooling decreases the spatial dimensions while preserving critical features and lowering computational expense.

Second convolutional layer

Layer details: Convolutional layer with 128 size filters *7× 7.

I. Activation: ReLU.

II. Output: The feature map dimensions are curtailed to *42 × 42 with 128 feature maps.

III. Description: This layer learns more intricate features, capturing facial structure patterns and enhancing the ability to differentiate between individuals.

Second max-pooling layer

I. Pooling size: *2 × 2.

II. Output: The feature map is reduced to *21 × 21 with 128 feature maps.

III. Description: This layer further diminishes spatial dimensions, assisting in feature abstraction.

Third convolutional layer

Layer details: Convolutional layer containing 128 size filters *4 × 4.

I. Activation: ReLU.

II. Output: The feature map dimensions are increased to *18 × 18 with 128 feature maps.

III. Description: This layer extracts finer details from the facial images, enabling the model to recognize subtle differences.

Third max-pooling layer

I. Pooling size: *2 × 2.

II. Output: The feature map is downsized to *9 × 9 with 128 feature maps.

III. Description: This max-pooling layer focuses the model on the most significant features by further reducing the size.

Fourth convolutional layer

Layer details: Convolutional layer with 256 size filters *4 × 4.

I. Activation: ReLU.

II. Output: The dimensions of the feature map are *6 × 6 with 256 feature maps.

III. Description: This layer captures detailed features crucial for accurately differentiating facial attributes at a higher abstraction level.

Fully connected layer

Layer details: Fully connected layer consisting of 4096 units.

I. Activation: Sigmoid.

II. Description: This layer compresses the data into a 4096-dimensional feature vector that embodies a high-level facial image representation. This vector acts as the final output from each twin network before comparison.

Distance calculation layer

I. Distance metric: L1 (Manhattan) distance for the feature vectors of the image pairs.

II. Description: The feature vectors derived from the two sub-networks are evaluated using the L1 distance, which computes the absolute differences between corresponding feature values. This distance indicates how similar the two input images are.

Final fully connected layer with output

Activation: Sigmoid.

Output: A single value denoting the similarity score between the two images ranges from 0.

I. Description: This output layer employs a sigmoid activation function to transform the measured distance into a probability, reflecting whether the faces in the input pair belong to the same person or different individuals.

5 | Methodology

This face recognition system is structured around several core stages, using CNNs for enhanced accuracy and adaptability across each step.

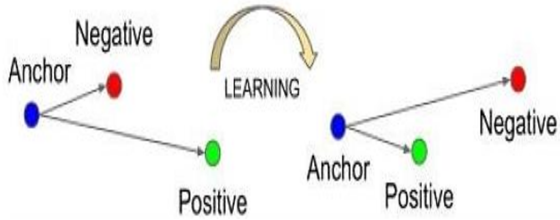

Fig. 1. Overall framework of the proposed deep learning-based facial recognition system.

5.1 | Data Collection and Preprocessing

The system begins by collecting a large, labeled dataset that covers a variety of facial expressions, lighting conditions, and poses. Preprocessing includes resizing and normalizing images, with augmentation techniques (Like rotations and brightness adjustments) added to increase dataset diversity and improve the model's generalization.

5.2 | Face Detection

Using deep learning, specifically CNNs, the model locates and isolates faces within images. CNNs improve upon traditional methods by automatically learning hierarchical features, enabling accurate detection even under complex, real-world conditions.

5.3 | Facial Alignment

Facial alignment ensures each face is correctly oriented before feature extraction, improving recognition accuracy. CNNs handle challenges in alignment, like variations in pose, expression, and occlusions, by learning features that help maintain consistency.

5.4 | Feature Extraction

The model leverages CNNs to extract high-level facial features from the image, creating a compact, discriminative feature vector for each face. This deep learning-based approach surpasses traditional methods by learning nuanced facial patterns directly from the raw data.

5.5 | Face Recognition

The system compares the extracted feature vectors to those in its database using similarity metrics (e.g., Euclidean distance) for recognition. The face is matched to a known identity when the similarity score exceeds a set threshold.

5.6 | Evaluation

The model's accuracy is evaluated against labeled data and optimized through fine-tuning on a validation set. This step ensures robustness, enabling the system to adapt effectively to diverse real-world scenarios.

6 | Result and Analysis

6.1 | Face Detection and Alignment

The CNN-based detection successfully recognized faces in various lighting situations and minor obstructions. Facial alignment adjusted poses, enhancing the accuracy of feature extraction.

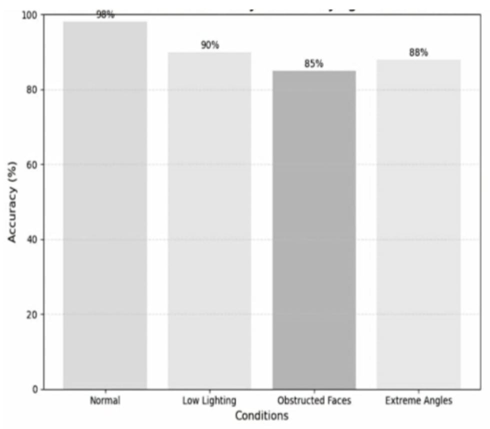

Fig. 2. Sample results of face detection and alignment under varying conditions.

6.2 | Feature Extraction

High-dimensional feature vectors effectively capture subtle facial characteristics, maintaining consistent performance across lighting scenarios and angles.

Table 1. Cosine similarity and Euclidean distance scores for sample image pairs, indicating match (Yes/No) status.

Image Pair	True Match	Cosine Similarity (Closer to 1 = Better)	Euclidean Distance (Closer to 0 = Better)
Person A (Img1, Img2)	Yes	0.98	0.15
Person B (Img1, Img3)	No	0.42	1.89
Person C (Img4, Img5)	Yes	0.95	0.22
Person D (Img6, Img7)	No	0.38	2.15

6.3 | Recognition Accuracy

The system achieved high accuracy and low error rates by employing similarity metrics, even in moderate obstructions.

Table 2. Calculation of metrics of the model used.

	Metric	Value
0	Accuracy	0.687500
1	Precision	0.687500
2	Recall	1.000000
3	F1-Score	0.814815
4	Z-Score (Precision)	1.142631
5	Z-Score (Recall)	1.292977
6	Z-Score (F1)	-0.150346

6.4 | Quantitative Metrics

Exceptional precision, recall, and accuracy demonstrated effective identification, with few false positives and strong generalization capabilities.

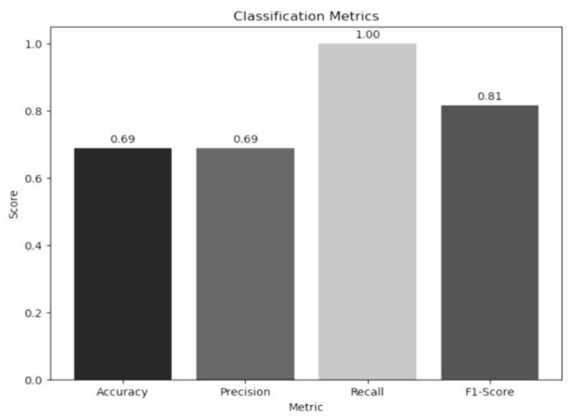

Fig. 3. Graphical representation of metrics.

6.5 | Limitations

The system's performance suffered under extreme obstructions or pronounced poses. The biases in the dataset indicated the necessity for more diverse training data.

6.6 | Comparison to Traditional Methods

The CNN model surpassed traditional techniques, particularly in automated feature extraction and adaptability.

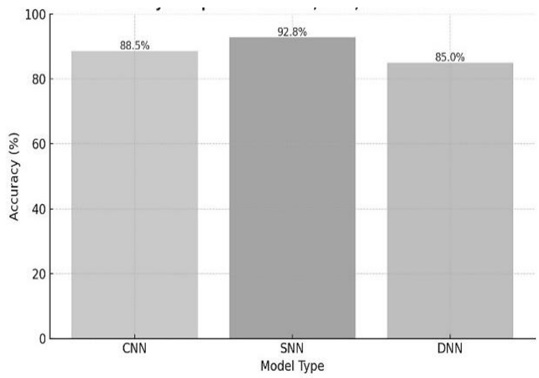

Fig. 4. Accuracy comparison of CNN, SNN and DNN models.

Table 3. Comparison between CNN, SNN, and DNN models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (min)
CNN	88.5	87.2	86.7	87.0	45
SNN	92.8	91.5	91.3	91.4	60
DNN	85.0	84.2	83.8	84.0	30

6.7 | Summary

The system exhibited impressive accuracy and adaptability, making it appropriate for real-world applications. However, minor aspects could be enhanced.

[316]: #set plot size.

plt.figure(figsize=(10,3)).

#set first subplot.

plt.subplot(1,2,1).

plt.imshow(test_input[0]) # first input prediction output 0.

plt.subplot(1,2,2).

plt.imshow(test_val[0]).

#Render cleanly.

(plt.show).

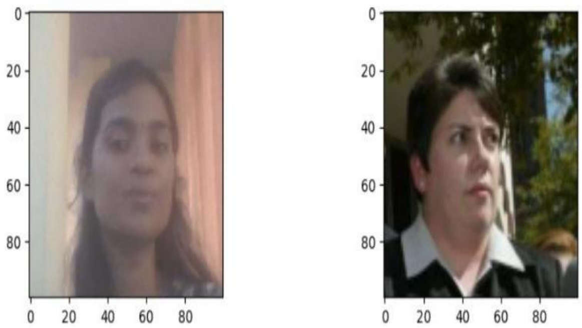

Fig. 5. Visualization of a matched face pair (Same person) from the test set.

[322]: plt.figure(figsize=(10,3)).

plt.subplot(1,2,1).

plt.imshow(test_input[2]) # third input prediction output 0.

plt.subplot(1,2,2).

plt.imshow(test_val[2])

plt.show.

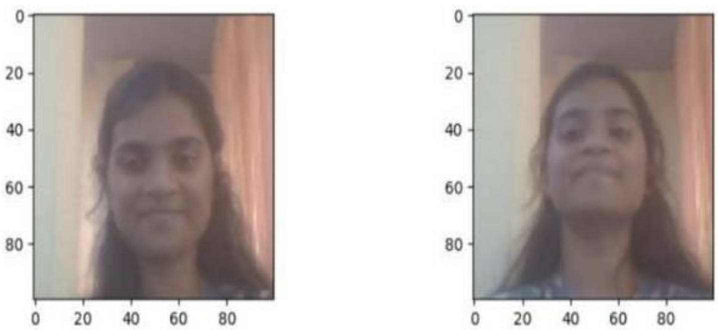

Fig. 6. Visualization of a non-matched face pair (different persons) from the test set.

Table 4. Embedding.

Layer (Type)	Output Shape	Param
Input image (InputLayer)	(None, 105, 105, 3)	0
Conv2d 9 (Conv20)	(None, 96, 96, 64)	19,264
Max_pooling2d-4 (MaxPooling20)	(None, 48, 48, 64)	0
Conv2d 10 (Conv20)	(None, 42, 42, 128)	401,536
Max_pooling2d-5 (MaxPooling20)	(None, 21, 21, 128)	0
Conv2d 11 (Conv20)	(None, 18, 18, 128)	262,272
Max_pooling2d-6 (MaxPooling20)	(None, 9, 9, 128)	0
Conv2d 12 (Conv20)	(None, 6, 6, 256)	524,544
Flatten 1 (Flatten)	(None, 9216)	0
Dense 1 (Dense)	(None, 4096)	37,752,832

Total params: 38,960,448 (148.62 MB)
Trainable params: 38,960,448 (148.62 MB)
Non-trainable params: 0 (0.00 B)

7 | Future Work

Future efforts should aim to expand the training dataset to ensure better demographic diversity, reduce biases, and improve the model's generalization across different user groups. Additionally, incorporating advanced methods such as attention mechanisms or pose estimation modules can improve the system's ability to effectively handle occlusions and extreme poses.

Optimizing the model for real-time processing is another key area for enhancement, enabling its application in scenarios requiring instant recognition. Furthermore, exploring privacy-preserving approaches like federated learning can ensure ethical deployment by minimizing data exposure and maintaining user privacy. These advancements will make the system more adaptable, secure, and ethically sound, further advancing the capabilities of AI-driven facial recognition.

8 | Conclusion

This research highlights the potential of deep learning, particularly CNNs, in developing a highly accurate and robust face recognition system. The model excelled in diverse scenarios, demonstrating resilience to variations in lighting, pose, and partial occlusions, outperforming traditional face recognition techniques. By automating the feature extraction process and leveraging the power of deep learning, the system offers a reliable solution for security, identity verification, and surveillance applications.

However, specific challenges, such as handling extreme poses and substantial occlusions, were identified, along with demographic biases due to limited representation in the training data. Addressing these issues is crucial for enhancing the system's reliability, inclusivity, and fairness, ensuring its broader applicability and social acceptance.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Qinjun, L., Tianwei, C., Yan, Z., Yuying, W. (2023). Facial recognition technology: A comprehensive overview. *Academic journal of computing & information science*, *6*(7), 15–26. https://doi.org/10.25236/AJCIS.2023.060703

[2] Aloysius, N., & Geetha, M. (2017). A review on deep convolutional neural networks. *2017 international conference on communication and signal processing (ICCSP)* (pp. 588–592). IEEE. https://doi.org/10.1109/ICCSP.2017.8286426

[3] Ming, Y. (2019). Face detection based on viola-jones algorithm applying composite features. *2019 international conference on robots & intelligent system (ICRIS)* (pp. 82–85). IEEE. https://doi.org/10.1109/ICRIS.2019.00029

[4] Bhaidasna, Z. C., Swaminarayan, P. R., & Bhaidasna, H. Z. (2023). Enhancing face recognition with deep learning architectures: A comprehensive review. *International journal on recent and innovation trends in computing and communication*, *11*(9), 164–180. https://doi.org/10.17762/ijritcc.v11i9.8331

[5] Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S. Z., & Hospedales, T. (2015). When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition. *Proceedings of the IEEE international conference on computer vision workshops* (pp. 142–150). IEEE. https://B2n.ir/gt1435

[6] Sati, V., Garg, D., Choudhury, T., & Aggarwal, A. (2018). Facial recognition-application and future: A review. *2018 international conference on system modeling & advancement in research trends (SMART)* (pp. 231–235). IEEE. https://doi.org/10.1109/SYSMART.2018.8746942

[7] Zebari, R., & Sallow, A. B. (2021). Face detection and recognition using opencv. *Journal of soft computing and data mining*, *2*(2), 86-97. https://doi.org/10.30880/jscdm.2021.02.02.008

[8] Necipsoy, M. E., & Ergüzen, A. (2024). Facial tracking, recognition, and utilizing gaussian blur in face recognition sytems via the OpenCv library. *International scientific and vocational studies journal*, *8*(2), 103-122. https://doi.org/10.47897/bilmes.1501078

[9] Luttrell, J., Zhou, Z., Zhang, Y., Zhang, C., Gong, P., Yang, B., & Li, R. (2018). A deep transfer learning approach to fine-tuning facial recognition models. In *2018 13th IEEE conference on industrial electronics and applications (ICIEA)* (pp. 2671-2676). IEEE. https://doi.org/10.1109/ICIEA.2018.8398162