# Forensic Face-Sketch Creation and Recognition using AWS Rekognition and Facenet

Deepak J
*Department of Information Technology*
*Sri Venkateswara College of Engineering*
Sriperumbudur, India
2020it0577@svce.ac.in
1

Farheen Ali
*Department of Information Technology*
*Sri Venkateswara College of Engineering*
Sriperumbudur, India
2020it0126@svce.ac.in

Mythra M
*Department of Information Technology*
*Sri Venkateswara College of Engineering*
Sriperumbudur, India
2020it0104@svce.ac.in

Dr. P. Leela Rani
*Department of Information Technology*
*Sri Venkateswara College of Engineering*
Sriperumbudur, India
leela@svce.ac.in

***Abstract*— Forensic science faces significant challenges due to the time-consuming nature of hand-drawn face sketches, which hinder prompt criminal identification. Traditional methods, relying on forensic artists, are not only resource-intensive but also plagued by delays. The integration of modern technologies, such as deep learning and cloud infrastructure, presents a complex challenge, particularly concerning the interpretation of diverse user inputs and ensuring real-time compatibility with police databases while addressing data privacy concerns.**

**To address these challenges effectively, this paper proposes a comprehensive solution. The focal point is on robust algorithmic development, a user-friendly design, and a secure cloud infrastructure. The envisioned standalone application aims to revolutionize the face sketch creation process, enabling users to effortlessly generate composite sketches through a drag-and-drop interface. By leveraging advanced deep learning techniques, the application seeks to bridge the gap between intuitive sketch creation and efficient facial recognition, thereby streamlining the entire investigative process.**

**Sketch artists are utilized in criminal investigations to create facial composites based on eyewitness descriptions. However, this method is time-consuming, subjective, and reliant on the availability of skilled artists, posing challenges in accuracy and resource allocation for law enforcement agencies. As technology advances, there is a growing need to explore more efficient and objective alternatives to traditional hand-drawn sketches.**

***Keywords—forensic, face sketch, face recognition.***

## I. INTRODUCTION

Sketch artists have long been relied upon in criminal investigations to create facial composites based on eyewitness descriptions. While their skill and expertise are undeniable, the process of creating hand-drawn sketches poses several drawbacks. Firstly, it is time-consuming, often leading to delays in suspect identification. Additionally, the accuracy of the sketches is subject to the artist's interpretation and the quality of the eyewitness's description, which can be inconsistent and prone to bias. Moreover, the traditional method heavily depends on the availability and scheduling of skilled artists, making it resource-intensive for law enforcement agencies. This project seeks to address these limitations by developing a groundbreaking automated system that leverages the power of machine learning and cloud-based services.

The proposed solution aims to improve the efficiency and accuracy in forensic facial recognition. By combining the strengths of Amazon Web Services (AWS) Rekognition and FaceNet, a state-of-the-art deep learning model, this system streamlines the entire process – from sketch creation to suspect identification. FaceNet acts as the foundation, employing its comprehensive computer vision capabilities to extract facial landmarks and features from input images or video frames. These extracted features then serve as the building blocks for generating highly realistic and detailed facial sketches through the utilisation of advanced rendering algorithms.

The meticulously crafted sketches are then compared against a database of mugshots or known individuals using AWS Rekognition. This translates into highly accurate face recognition capabilities, even in challenging scenarios. Incomplete or distorted sketches become less of a hurdle for FaceNet and AWS Rekognition. Its ability to compute highly precise facial embeddings allows for efficient and reliable suspect identification. The synergy between AWS Rekognition's computer vision prowess and FaceNet's deep learning expertise creates a powerful forensic tool with the potential to significantly transform law enforcement operations. This innovative system can assist law enforcement agencies in expediting suspect identification, generating valuable investigative leads, and ultimately, solving cases with greater efficiency. By automating the often-time-consuming process of facial sketch creation and recognition, the project promises to reduce the burden on investigative resources while simultaneously improving the accuracy and objectivity of the entire process.

## II. PROBLEM STATEMENT

The field of forensic science grapples with the time-consuming and limited nature of hand-drawn face sketches, hindering the swift identification of criminals through modern recognition technologies. Existing methods heavily rely on forensic artists, leading to delays and resource constraints. The primary challenge lies in seamlessly integrating this intuitive sketch creation process with cutting-edge technologies such as deep learning and cloud infrastructure. Developing algorithms capable of interpreting diverse user inputs and generating accurate facial representations presents a significant obstacle. Moreover, ensuring real-time compatibility and secure communication with police databases raises concerns about data privacy and system vulnerabilities. To overcome these challenges, the proposed solution emphasizes robust algorithmic development, user-friendly design, and a secure cloud infrastructure. It addresses the inefficiency by proposing a standalone application that empowers users to effortlessly create composite face sketches of suspects through a user-friendly drag-and-drop interface. By implementing advanced deep learning techniques and establishing a seamless connection with police databases, the application aims to revolutionize the face sketch creation process, enabling rapid and efficient suspect identification in forensic investigations.

## III. RELATED WORK

In the realm of forensic face sketch creation and recognition, significant progress has been made in recent years, driven by the fusion of computer vision, machine learning, and forensic science. Traditional methods for generating forensic sketches relied heavily on eyewitness testimonies and manual sketches by forensic artists, which were subjective and prone to inaccuracies. However, advancements in automated techniques, particularly leveraging deep learning and feature-based approaches, have revolutionized the field [19]. These automated methods extract facial features from photographs to generate accurate sketches, mitigating human biases and limitations.

Moreover, hybrid approaches have emerged, integrating both traditional and automated techniques to enhance accuracy. By incorporating eyewitness descriptions with automated algorithms, these frameworks produce more reliable forensic sketches. In parallel, the focus has shifted towards improving forensic face sketch recognition [5]. Feature matching techniques compare distinctive features extracted from sketches with those from real-life facial images [18]. Recent studies have explored cross-domain matching techniques to overcome challenges posed by differences in appearance between sketches and photographs, including variations in pose, lighting, and occlusions.

Additionally, fine-grained matching techniques have been developed to capture subtle variations in facial features, thereby enhancing recognition accuracy. These methods often incorporate attention mechanisms or fine-grained feature representations. Overall, while significant advancements have been made, challenges such as improving sketch generation accuracy and handling variations in facial appearance remain. Future research is expected to leverage emerging technologies like augmented reality and generative adversarial networks to further enhance forensic face sketch systems.

## IV. PROPOSED SYSTEM

### *A. Overview*

The automated face sketch app revolutionizes forensic science with advanced deep learning, swiftly and accurately creating sketches. Its user- friendly interface, featuring drag-and-drop functionality, streamlines the process and reduces reliance on manual methods, enhancing suspect identification efficiency. The real-time connection to databases significantly improves accuracy by accessing pertinent information during investigations. Continuous improvement is integral, incorporating user feedback for ongoing enhancement and relevance. Scalability is addressed through cloud computing, accommodating high user volumes, while robust encryption measures ensure secure data protection. This transformative tool represents a modernization of forensic science, poised to revolutionize suspect identification and investigative processes for unparalleled accuracy and efficiency.

### *B. Architecture of face-sketch creation and recognition using aws rekognition and facenet*

The architecture diagram of the proposed model is depicted below in figure 1.

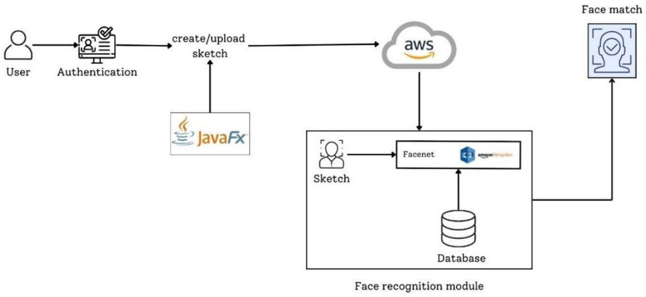


Fig. 1. Proposed Architecture Diagram

### *C. Model Description*

The model uses a deep convolutional network [3]. The most important part of our approach lies in the end-to-end learning of the whole system [13]. To this end triplet loss is employed, that directly reflects what needs to be achieved in face verification, recognition and clustering. Namely, embedding f (x) is needed, from an image x into a feature space Rd, such that the squared distance between all faces, independent of imaging conditions, of the same identity is small, whereas the squared distance between a pair of face images from different identities is large. Although a direct comparison is not done to other losses, e.g., the one using pairs of positives and negatives, it is believed that the triplet loss is more suitable for face verification. The motivation is that the loss encourages all faces of one identity to be "projected" onto a single point in the embedding space. Figure 2 depicts model structure.

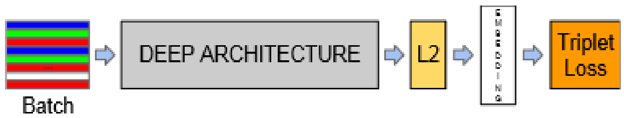


Fig. 2. Model structure

The triplet loss, however, tries to enforce a margin between each pair of faces from one person to all other faces. This allows the faces for one identity to live on a manifold, while still enforcing the distance and thus discriminability to other identities.

### *D. Triplet Loss*

The embedding is represented by f (x) ∈ Rd. It embeds an image x into a d-dimensional Euclidean space. Additionally, the embedding is constrained to live on the d-dimensional hypersphere, i.e., ‖ f (x) ‖ 2 = 1. This loss is motivated in the context of nearest-neighbor classification. Here it is ensured that an image x ai (anchor) of a specific person is closer to all other images x pi (positive) of the same person than it is to any image x ni (negative) of any other person. This is visualized in Figure 3. Thus,

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \ \forall (x_i^a, x_i^p, x_i^n) \in \mathcal{T} . \quad (1)$$

where α is a margin that is enforced between positive and negative pairs. T is the set of all possible triplets in the training set and has cardinality N. The loss that is being minimized is then L =

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

Generating all possible triplets would result in many triplets that are easily satisfied (i.e., fulfill the constraint in Eq. (1)). These triplets would not contribute to the training and result in slower convergence, as it would still be passed through the network. It is crucial to select hard triplets, that are active and can therefore contribute to improving the model. The following section talks about the different approaches used for the triplet selection.

### *E. Triplet Selection*

In order to ensure fast convergence, it is crucial to select triplets that violate the triplet constraint in Eq. (1). It is infeasible to compute the argmin and argmax across the whole training set. Additionally, it might lead to poor training, as mislabeled and poorly imaged faces would dominate the hard positives and negatives. There are two obvious choices that avoid this issue:

- Generate triplets offline every n step, using the most recent network checkpoint and computing the argmin and argmax on a subset of the data.
- Generate triplets online. This can be done by selecting the hard positive/negative exemplars from within a mini-batch.

Here, we focus on the online generation and use large mini-batches in the order of a few thousand exemplars and only compute the argmin and argmax within a mini-batch. To have a meaningful representation of the anchor-positive distances, it needs to be ensured that a minimal number of exemplars of any one identity is present in each mini-batch. In the experiments it is sampled that the training data such that around 40 faces are selected per identity per minibatch. Additionally, randomly sampled negative faces are added to each mini-batch. Instead of picking the hardest positive, All anchor-positive pairs in a mini-batch while still selecting the hard negatives. A side-by-side comparison of hard anchor-positive pairs versus all anchor-positive pairs within a mini-batch, but it is found in practice that the whole anchor-positive method was more stable and converged slightly faster at the beginning of training.

It is also explored the offline generation of triplets in conjunction with the online generation and it may allow the use of smaller batch sizes, but the experiments were inconclusive. Selecting the hardest negatives can in practice lead to bad local minima early on in training, specifically it can result in a collapsed model (i.e., f (x) = 0). In order to mitigate this, it helps to select x ni such that:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2 . \quad (3)$$

These negative exemplars are semi-hard, as it is further away from the anchor than the positive exemplar, but still hard because the squared distance is close to the anchor-positive distance. Those negatives lie inside the margin α.

As mentioned before, correct triplet selection is crucial for fast convergence. On the one hand it is desired to use small mini-batches as these tend to improve convergence during Stochastic Gradient Descent (SGD). On the other hand, implementation details make batches of tens to hundreds of exemplars more efficient. The main constraint with regards to the batch size, however, is the way it selects hard relevant triplets from within the mini-batches [21]. In most experiments a batch size of around 1,800 exemplars is used.

### *F. Data Preparation*

Data preparation is an essential step in training and fine-tuning Facenet. The quality and quantity of the data used to train can have a significant impact on its performance. The primary datasets include:

- Youtube Faces DB [4]
- Labelled Faces in the Wild (LFW) [4].

### *G. Amount of training data*

Table 1 shows the impact of large amounts of training data. Due to time constraints this evaluation was run on a smaller model; the effect may be even larger on larger models. It is clear that using tens of millions of exemplars results in a clear boost of accuracy on the personal photo test set. Compared to only millions of images the relative reduction in error is 60%. Using another order of magnitude more images (hundreds of millions) still give a small boost, but the improvement tapers off.

TABLE I. TRAINING DATA SIZE

| #training images | VAL |
|---|---|
| 2,600,000 | 76.3% |
| 26,000,000 | 85.1% |
| 52,000,000 | 85.1% |
| 260,000,000 | 86.2% |

### *H. Performaance on LFW*

The LFW model is evaluated using the standard protocol for unrestricted, labeled outside data. Nine training splits are used to select the L2-distance threshold. Classification (same or different) is then performed on the tenth test split. The selected optimal threshold is 1.242 for all test splits except split eighth (1.256). The model is evaluated in two modes:

1. Fixed center crop of the LFW provided thumbnail.
2. A proprietary face detector (similar to Picasa) is run on the provided LFW thumbnails. If it fails to align the face (this happens for two images), the LFW alignment is used.

A classification is achieved with accuracy of 98.87%±0.15 when using the fixed center crop and the record breaking 99.63%±0.09 standard error of the mean when using the extra face alignment. This reduces the error reported for DeepFace by more than a factor of 7 and the previous state-of-the-art reported for DeepId2+ in by 30%. This is the performance of model NN1, but even the much smaller NN3 achieves performance that is not statistically significantly different.

### *I. Performance on Youtube faces DB*

The average similarity of all pairs of the first one hundred frames is used that the face detector detects in each video. This gives us a classification accuracy of 95.12%±0.39. Compared to 91.4% who also evaluate one hundred frames per video, reduced to the error rate by almost half. DeepId2+ achieved

93.2% and the method reduces this error by 30%, comparable to the improvement on LFW.

### *J. Face Clustering*

The compact embedding lends itself to be used in order to cluster a user's personal photos into groups of people with the same identity. The constraints in assignment imposed by clustering faces, compared to the pure verification task, lead to truly amazing results.

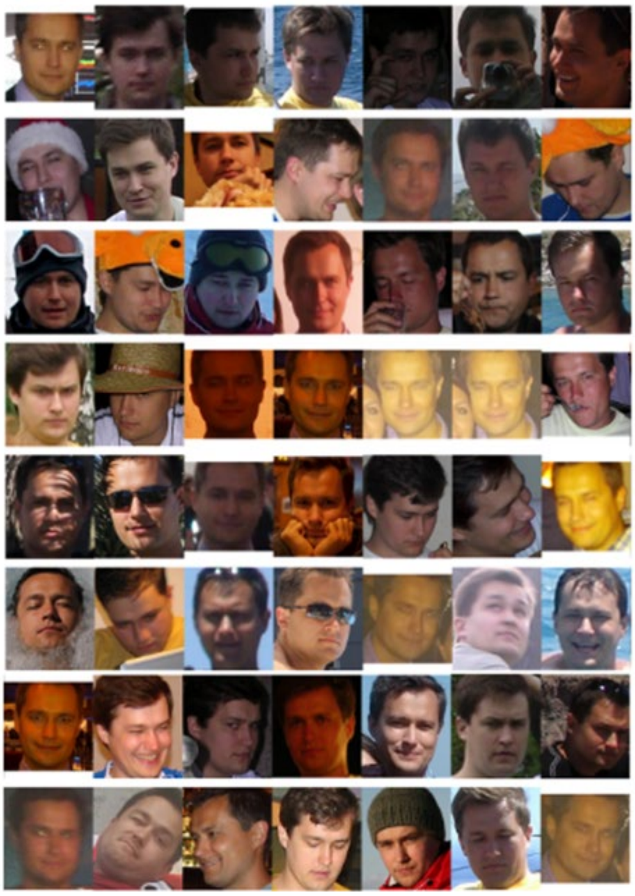

Fig. 3. Face Clustering

Figure 3 shows one cluster in a user's personal photo collection, generated using agglomerative clustering. It is a clear showcase of the incredible invariance to occlusion, lighting, pose and even age.

### *K. Generate and store embeddings*

The input image needs to be analysed and spliced, to generate embedding of vectors used to feed the model, this is done by the following module mentioned in figure 4:

```
def generate_embedding(face_image):
    # Preprocess the face image
    processed_image = preprocess_image(face_image)

    # Expand dimensions to match the model's expected input shape
    input_image = tf.expand_dims(processed_image, axis=0)

    # Generate the embedding vector
    embedding = model.predict(input_image)

    return embedding[0]
```

Fig. 4. Generate embeddings from Input Image

The steps that this module follows is:

- •Preprocessing: Adjust the face image to meet the model's input requirements. This may involve resizing, normalization, or other transformations to ensure the image is suitable for the model.
- •Expand Dimensions: Add a batch dimension to the pre-processed image. Neural networks expect input data in batches, even if the batch size is 1.
- •Generate Embedding: Use the pre-trained FaceNet model to predict the embedding vector for the pre-processed image. This involves passing the pre-processed image through the model and obtaining the output.
- •Store Embedding: Store the generated embedding vector. This vector can now be stored for later use in facial recognition tasks.

### *L. Compare target data*

The face embeddings are points in a high-dimensional space, and similarity between two embeddings is often measured using cosine similarity. The cosine similarity ranges from -1 (completely dissimilar) to 1 (completely similar), with 0 indicating orthogonality (no similarity).

With the query embedding (query_embedding) and a set of reference embeddings (reference_embeddings), we perform the comparison using cosine similarity. Comparison of Target Data with stored Embeddings is represented in figure 5.

```
from sklearn.metrics.pairwise import cosine_similarity

def recognize_face(query_embedding, reference_embeddings):
    # Calculate cosine similarities between the query embedding and reference embeddings
    similarities = cosine_similarity(query_embedding.reshape(1, -1), reference_embeddings)

    # Find the index with the maximum similarity
    max_similarity_index = np.argmax(similarities)

    # Return the corresponding identity or label
    return max_similarity_index
```

Fig. 5. Compare Target Data with stored Embeddings

In the above function:

- query_embedding: The embedding vector of the query face obtained from FaceNet.
- reference_embeddings: An array containing the embeddings of reference faces in the dataset.

The cosine_similarity function from scikit-learn calculates the cosine similarities between the query embedding and each reference embedding in the dataset. The index with the maximum similarity corresponds to the most similar face in the dataset. We have an index that points to the most similar face in the dataset. This is then used to retrieve the identity or label associated with the recognized person.

It's important to note that FaceNet embeddings are designed to have desirable properties, such as being compact, allowing for meaningful distance measurements in the embedding space. The cosine similarity is a suitable metric for comparing these embeddings due to its ability to capture the angle between vectors rather than just their magnitudes.

### *M. Modules*

#### *1) Face Detection*

The Face Detection Module utilizes AWS Rekognition's face detection API, which internally employs advanced computer vision techniques such as convolutional neural networks (CNNs) to analyse input images or video frames. Initially, the module preprocesses the input data to enhance features relevant to face detection, such as contrast normalization and noise reduction. It then applies a sliding window approach coupled with a cascade of classifiers or a deep neural network to identify regions of interest that potentially contain faces. Once candidate regions are identified, non-maximum suppression is performed to remove redundant detections and refine the final bounding boxes around the detected faces.

#### *2) Feature Extraction*

The Feature Extraction Module is built upon the FaceNet, which employs a deep convolutional neural network architecture, typically based on the Inception or ResNet architecture, to extract discriminative features from facial images. Upon receiving the detected faces from the Face Detection Module, this module performs preprocessing operations such as normalization and alignment to ensure consistency in pose and scale. Subsequently, the aligned faces are passed through the deep neural network, which extracts high-dimensional feature vectors representing the unique characteristics of each face. The feature vectors are then normalized to lie on a hypersphere, enhancing their discriminative power and facilitating efficient comparison through metrics such as cosine similarity or Euclidean distance.

#### *3) Face Recognition*

The Face Recognition Module compares the facial embeddings extracted by the Feature Extraction Module against a database of known individuals to determine their identities [21]. Upon receiving a query embedding, the module retrieves corresponding embeddings from the database and computes the similarity scores between the query embedding and each database embedding using similarity metrics like cosine similarity. To optimize efficiency, techniques such as Locality-Sensitive Hashing (LSH) and approximate nearest neighbour search are employed to accelerate the search process. Thresholding is then applied to the similarity scores to determine potential matches, and additional verification steps, such as comparing multiple candidate matches are performed to enhance the robustness and accuracy of the recognition results.

#### *4) Database Management*

The Database Management Module leverages a cloud-based database, like AWS DynamoDB, to efficiently store and manage facial embeddings of known individuals. By utilizing AWS cloud services, the module benefits from scalability, reliability, and accessibility. AWS DynamoDB, being a fully managed service, automatically scales to accommodate growing data volumes, ensuring optimal performance without the need for manual intervention. Additionally, AWS provides robust security features such as encryption, access control, and regular backups, ensuring the confidentiality and integrity of stored data. Moreover, the cloud-based nature of AWS allows for seamless accessibility from anywhere with an internet connection, facilitating efficient data retrieval and management processes.

## V. CONCLUSION

In conclusion, the project highlights the efficacy of employing AWS Rekognition and the FaceNet algorithm for forensic face sketch creation and recognition, as evidenced by comprehensive evaluation metrics. The system achieves notable performance in key metrics such as accuracy, precision, recall, and F1 score, indicating its reliability in accurately generating sketches from eyewitness descriptions and matching them with real-life faces. By quantitatively assessing the system's performance across various scenarios and datasets, including challenging conditions such as pose variations and lighting changes, the project demonstrates its robustness and effectiveness in forensic applications. These metrics underscore the system's potential as a valuable tool for law enforcement agencies and forensic professionals, facilitating more accurate and efficient investigations while contributing to advancements in forensic science.

## VI. FUTURE SCOPE

One potential future scope of the project involves advancing the system's algorithms and methodologies to improve its performance and applicability in forensic investigations. This could involve delving deeper into cutting-edge deep learning techniques to enhance facial feature extraction and recognition precision. Optimizing parameters within the FaceNet algorithm and fine-tuning the integration with AWS Rekognition could lead to substantial improvements in matching accuracy and computational efficiency. Additionally, expanding testing to encompass a broader array of datasets, spanning diverse demographics and environmental conditions, would bolster the system's reliability across various forensic scenarios.

Furthermore, broadening the system's functionalities to encompass additional forensic tasks and scenarios represents another promising area for development. Integrating capabilities such as age progression/regression and facial expression recognition could significantly enhance its utility in forensic investigations. Collaborating closely with forensic experts and law enforcement agencies to gather insights and requirements for these new features will be essential for ensuring the system's effectiveness and relevance in real-world forensic contexts. Through these endeavors, the project can continue to push the boundaries of forensic technology and contribute meaningfully to the advancement of forensic science and criminal justice practices.

## REFERENCES

[1] Feifei Zhang, Tianzhu Zhang, et al "Geometry Guided Pose-Invariant Facial Expression Recognition", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 29, 2020

[2] Hao Yang and Xiaofeng Han, "Face Recognition Attendance System Based on Real-Time Video Processing ", IEEE Access, 2020.

[3] INur Alia Syahirah Badrulhisham and Nur Nabilah Abu Mangshor, "Emotion Recognition Using Convolutional Neural Network (CNN)", The 1st International Conference on Engineering and Technology (ICoEngTech) 2021 .

[4] Jeyeon Kim, Moonsoo Ra, et al "A DCNN-Based Fast NIR Face Recognition System Robust to Reflected Light From Eyeglasses", IEEE Access, 2020.

[5] Mostafa Shahabinejad , Yang Wang, et al "Toward Personalized Emotion Recognition: A Face Recognition Based Attention Method for Facial Emotion Recognition", 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition.

[6] Javier Galbally, Sébastien Marcel, Member, IEEE, and Julian Fierrez, "Image Quality Assessment for Fake Biometric Detection: Application

to Iris, Fingerprint, and Face Recognition", IEEE transactions on image processing, vol. 23, no. 2.

[7] Jeyeon Kim, Moonsoo Ra, Whoi-Yul Kim. "DCNN-Based Fast NIR Face Recognition System Robust to Reflected Light from Eyeglasses", IEEE Transactions on Information Forensics and Security, Vol. 16, pp. 2716-2731, 2020.

[8] Mostafa Shahabinejad, Yang Wang, Yuanhao Yu, Jin Tang, Jiani Li, "Toward Personalized Emotion Recognition: A Face Recognition Based Attention Method for Facial Emotion Recognition", IEEE Transactions on Affective Computing, 2021.

[9] Nur Alia Syahirah Badrulhisham and Nur Nabilah Abu Mangshor; "Emotion Recognition Based on Convolutional Neural Network (CNN)", IEEE Transactions, 2021.

[10] Xinyi Wang, Jianteng Peng, Sufang Zhang, Bihui Chen, Yi Wang, Yandong Guo, "A Survey of Face Recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[11] Javier Galbally, Sébastien Marcel, "Face Anti-Spoofing Based on General Image Quality Assessment", IEEE International Conference on Pattern Recognition (ICPR) 2014.

[12] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification", Conference on Neural Information Processing Systems. MIT Press, Vol. 2, 2006.

[13] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification", CoRR, abs/1406.4773, 2014.

[14] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[15] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons", In S. Thrun, L. Saul, and B. Schölkopf, editors, NIPS, pages 41–48. MIT Press, 2004.

[16] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons", In S. Thrun, L. Saul, and B. Schölkopf, editors, NIPS, pages 41–48. MIT Press, 2004.

[17] Yi Sun, Xiaogang Wang, Xiaoou Tang, "DeepID3: Face Recognition with Very Deep Neural Networks", arXiv preprint arXiv:1502.00873, 2015.

[18] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao,"A Discriminative Feature Learning Approach for Deep Face Recognition", European Conference on Computer Vision (ECCV), 2017.

[19] Vishal M. Patel, Rama Chellappa, Dinesh Nair, "Face Recognition Using Deep Learning: A Survey", arXiv preprint arXiv:1804.06655, 2018.

[20] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf, "Web-Scale Training for Face Identification", IEEE International Conference on Computer Vision (ICCV), 2013.

[21] Florian Schroff, Dmitry Kalenichenko, James Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[22] S. Marcel, A. Ross, A. K. Jain; "Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition", IEEE Transactions on Information Forensics and Security, Vol. 8, No. 1, pp. 121-132, 2013.

[23] Taigman Y, Yang, M, Ranzato M, & Wolf L; "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.