Facial Recognition Systems in Edge Devices

Ashwin Ajoy Dharmavaram
5th Semester AIML
RV College of Engineering
Bangalore, India
ashwinajoyd.ai22@rvce.edu.in

Keerti Patil
5th Semester AIML
RV College of Engineering
Bangalore, India
keertipatil.ai22@rvce.edu.in

***Abstract*—Facial recognition (FR) is the ability of a system/ model to categorize a person's image into an ID, or. FR research has had several breakthroughs in recent years thanks to the increasing popularity and accuracy of Convolutional Neural Networks (CNN). However, FR systems tend to be quite large in terms of bytes with a large number of parameters. This makes them difficult to deploy on memory-constrained devices. The proposed work creates a lightweight Facial Recognition model that can be containerized and deployed in edge devices with little RAM. This paper shows deploying of a complex model such as dlib in an environment with little computation resources. Furthermore, this is mainly intended to be used in the airports to increase their surveillance by implementing the face recognition model to uniquely verify the identification of passengers by comparing with the existing terrorist database.**

***Index Terms*—Facial Recognition, FR, Face Embeddings, dlib, surveillance**

I. Introduction

Recently, the rise of Multilayer Perceptron-based Convolutional Neural Networks created several breakthroughs in Facial Recognition. However, before 2014, non convolutional ways were used to solve the problem. The first technique to solve the problem, was using Eigenface [1]. Eigenface converted the input image into a 1D grayscale vector, and then used Principal Component Analysis to group together faces that were closer. Fisherface pointed out that PCA used in Eigenface for dimensionality reduction maximizes the variance of all samples. Eigenface did not take sample category into account. Therefore, Fisherface [2] considered face labels in the training set and used Linear Discriminant Analysis (LDA) for dimensionality reduction, which maximized the ratios of inter-class and intra-class variance.
When it comes to deep learning frameworks, several steps are in common across frameworks, especially preprocessing [3].

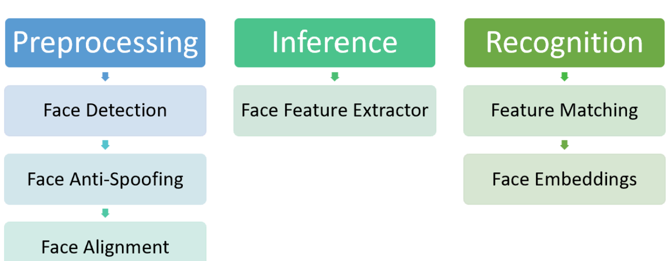

Fig. 1: Deep Neural Networks Image Processing Methodology

The main steps of preprocessing are:

1) Face Detection: Face Detection can be done in a couple of ways, some popular techniques are object detection [4], [5], and treating the face like a special entity, which results in higher accuracy.
2) Face Anti-Spoofing: Face anti-spoofing (FAS) is a security technique used to prevent biometric authentication systems from being fooled by fake face representations, such as printed photos, videos, 3D masks, or deepfake attacks. The major methods for face anti-spoofing usually input signals from RGB cameras.
3) Face Alignment: In images, the subject may look at the camera from different facial angles, making it difficult to create embeddings, or detect features. So, face alignment is used to keep uniformity.

In this project we have used dlib framework, and deployed it on a Raspberry Pi 5. Dlib is based on the ERT algorithm and uses 68 feature points on the face to perform facial recognition [6]. Dlib's face detector uses the HOG gradient histogram. The Histogram of Oriented Gradients (HOG) is a feature descriptor used in computer vision and image processing for object detection, particularly in facial recognition and pedestrian detection [7]. HOG works by analyzing the gradients (using a kernel such as the sobel operator) in an image and computing a histogram of gradient orientations.

Dlib provides 2 methods of performing face detection, HOG and CNN based. We use the HOG method as the CNN method requires a lot of computation resources.

The organization of the paper after the abstract and introduction is as follows. Section II has a Literature Survey of 27 papers in total and inference from the same, which helped in identifying the research gaps in the current works. Section III has the Methodology utilized in the proposed work which is followed by Section IV describing the implementation of the proposed methodology. This is followed by Section V with Results and Discussion describing results obtained with accuracy accordingly, and Section VI with the conclusion of the project.

II. Literature Survey

The paper [8] is a survey on historical methods of facial recognition, before deep learning such as using facial geometry for facial recognition purposes, and the paper states that it had not been implemented. Finally it examines the accuracies of various current models such as FaceNet, Deepface, etc, and it shows that among the 26 selected models, FaceNet has the highest accuracy of 99.63%. Meanwhile, [9] goes in the opposite direction and attempts to generate facial images given the embeddings. While not relevant to our work, we are reminded about the constant improvements to both generative AI, and computer hacking techniques and that we must improve data security on our system. Similarly, [10] proposes a system for encryption fo facial images. [11] aims to improve low resolution image face recognition. The authors created a Local and Global Feature Attention Fusion network that performs facial recognition by adaptively allocating attention between local and global information complementarily. They do this so that common issues such as poor lighting, facial occlusion, etc, make little difference.

To combat problems such as poor amount of images for training [12] explores the problem of synthetic dataset generation by converting the problem to an optimization problem solved through gradient descent, and proposing a new model called "HyperFace". [13] analyses literature on facial recognition on people wearing masks in light of the COVID-19 pandemic. The paper considers the models that their cited papers have used for solving the problem and optimizes their parameters. [14] also tries to solve this problem. The authors use a cropping based approach while combining it with a Convolutional Block Attention Module. The optimal cropping is explored for each case, while the CBAM module is adopted to focus on the regions around the eyes.

The performance of face recognition system degrades when the variability of the acquired faces increases. Prior work alleviates this issue by either monitoring the face quality in pre-processing or predicting the data uncertainty along with the face feature. [15] proposes MagFace, a category of losses that learn a universal feature embedding whose magnitude can measure the quality of the given face. Under the new loss, it can be proven that the magnitude of the feature embedding monotonically increases if the subject is more likely to be recognized. [16] demonstrates use of neural networks for patient facial expression recognition. The paper reaches an accuracy of 70%, which is close to the state of the are with lesser number of layers.

[17] creates a lightweight facial recognition model using facial feature alignment, and creates a framework wrapping the most popular facial recognition models such as DeepFace. The development of deep learning-based biometric models that can be deployed on devices with constrained memory and computational resources has proven to be a significant challenge. [18] Ghost modules use a series of inexpensive linear transformations to extract additional feature maps from a set of intrinsic features, allowing for a more comprehensive representation of the underlying information. Ghost modules use a series of inexpensive linear transformations to extract additional feature maps from a set of intrinsic features, allowing for a more comprehensive representation of the underlying information.

Most commonly used method for facial detection is using Haar_Cascades, and [19] used the Haar_Cascade Algorithm's 128 dimension vector that it uses for facial encoding. Instead of converting it to grayscale, it uses a subprocess that converts the grayscale image to RGB. This greatly improved the accuracy to 98.39% (20% increase) along with 63.59% precision and 98.3% recall.

[20] analyses the impact of Foundation models, models that are trained on highly diverse, and large scale datasets. Foundation Models are highly versatile and broadly applicable to a variety of different tasks. The authors find that results are comparable to the models that are created from scratch using small datasets.

Emerging research has found that spherical spaces better match the underlying geometry of facial images. [21] states that due to their dependence on deterministic embeddings, noisy images are mapped into poorly learned regions of space which leads to inaccuracies. [22] states that most facial recognition models only work with clean data, while real world is always occluded, which is a main cause of low validation accuracy. To solve this, the authors intentionally mask areas of images to train the model to predict the occluded parts, and hence generalize better.

[23] gives description on how to optimize the checking of the customer baggages using the CT technology, the conventional technology that is used in order to carry on document verification, etc. The behavioral biometric technology is the research domain in which key concepts like facial recognition and retinal scans of the passengers at the airport are focused on to, research has been undertaken for the implementation of the respective technologies.

[24] Face recognition has evolved significantly with deep learning, surpassing traditional methods like Eigenfaces and Fisherfaces. Models such as DeepFace and FaceNet use convolutional neural networks (CNNs) to learn hierarchical features, achieving high accuracy. However, challenges like balancing accuracy and efficiency in real-world scenarios remain. Hybrid frameworks like LightFace address these by combining lightweight architectures with robust feature extraction, enabling high performance in resource-constrained environments.

[25] focuses on contactless passengers' boarding without physical barriers ensuring the hygiene during COVID 19 pandemic. The security at the airport was ensured based on contactless scans using the technology. Technology to ensure fraud detection was also implemented. [26] highlights the importance of the facial recognition for the safety of the passengers at airports to reduce the crimes and increase the surveillance keeping GDPR (General Data Protection Regulations) and data acquisition into consideration.

[27] The paper reviews face recognition techniques, focusing on challenges like illumination and pose variation, categorizing methods, and addressing system evaluation and psychophysical studies.

From the literature survey of twenty-seven papers, as mentioned, it was quite clear that, current works do not focus much on the real-world real-time application of surveillance as an application of the implementation of facial recognition. It is quite clear that the `Facenet` system architecture is capable of being the highest accuracy-providing model to date, but is already being implemented in Facial Recognition. The study highlights that among different deep learning models used, `dlib` from `face_recognition_models` of the `face_recognition` library is unexplored and has scope for higher accuracy as it simplifies the creation of facial vector embeddings which is essential for the project's efficiency.

III. Methodology

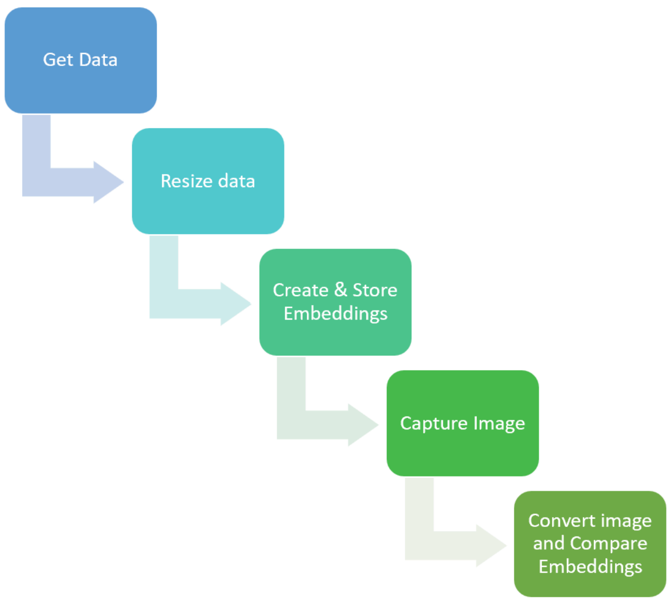

Fig. 2: Project Methodology

The methodology with which the proposed system proceeds:

1) Data Collection: Data Collection needs to be done carefully, as the the DLib model creates the embeddings by focusing on the image features. Therefore, it is required that the image of the person needs to be taken:
   - Front facing
   - White background
   - Without glasses
   - Minimal amount of makeup
2) Resizing data: The image is then resized to be 25% of it's original size.
   `small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)`
3) Creating and Storing Embeddings: After resizing to 250 x 250 pixels for consistency in further processing, vector embeddings are generated for the images in the database and converted to a structured format and are named as encodings. These encodings are stored in a file encodings.npy, which acts as a centralized database of facial embeddings.
4) Capturing Image: The image of the passenger is taken as an input, through the live webcam, the embeddings of which are calculated for further processing.
5) Converting Image and Comparing Embeddings: After creating embeddings of the input image, they are compared with the encodings, and the suspected ID is returned as the result, if there exist similarities between the passenger input image and the existing terrorists' database's encodings. This helps in increasing the surveillance at the airports by helping in uniquely verifying the identification of each passenger at the airport.

IV. Implementation

Through our experimentation, it was found that there exists a threshold distance of 0.45 between present and predicted face embeddings is sufficient to provide accurate predictions on Face Recognition. This distance may differ between people of different regions.

The facial recognition system was implemented using the `face_recognition` and `OpenCV` libraries in Python. The system operates in real-time using a webcam or processes a prerecorded video file for recognizing and logging individuals. Creating bounding boxes for image results possible creates a slow system, and may not work while using `Docker`, so we have elected to use a headless version of the `OpenCV` library.

```python
import face_recognition
import cv2
import numpy as np
from datetime import datetime
import csv
import os

video_capture = cv2.VideoCapture(0)

# Create arrays of known face encodings and their names
known_face_encodings = []
known_face_names = []

known_face_encodings = np.load('encodings.npy')

with open('labels.txt', 'r') as file:
    known_face_names = [line.strip() for line in file]

students = known_face_names.copy()

face_locations = []
face_encodings = []
face_names = []
process_this_frame = True

now = datetime.now()
current_date = now.strftime("%Y-%m-%d")

classname = "class"
```

Fig. 3: Code for Image Encodings

Referring to Fig. 3, the code snippet here creates arrays of known face encodings (after making the encodings) of the terrorist images with the names (IDs) accordingly in the labels.txt file, which will be used for verification of the identification of the passengers later.

```python
if process_this_frame:
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
    rgb_small_frame = np.ascontiguousarray(small_frame[:, :, ::-1])
    face_locations = face_recognition.face_locations(rgb_small_frame)
    face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

    face_names = []
    for face_encoding in face_encodings:
        matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
        name = "Unknown"

        face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
        best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            name = known_face_names[best_match_index]

        face_names.append(name)
        if name in known_face_names:
            if name in students:
                students.remove(name)
                print(name)
                current_time = datetime.now().time()
                time_string = current_time.strftime("%H:%M:%S")
                lnwriter.writerow([name, time_string])

process_this_frame = not process_this_frame
```

Fig. 4: Comparison of the Encodings

Referring to Fig. 4, the code here compares the encodings existing in the image database with the encodings of the input image taken through the webcam, followed by their comparison which helps in the detection of the terrorists with the help of the existing database.

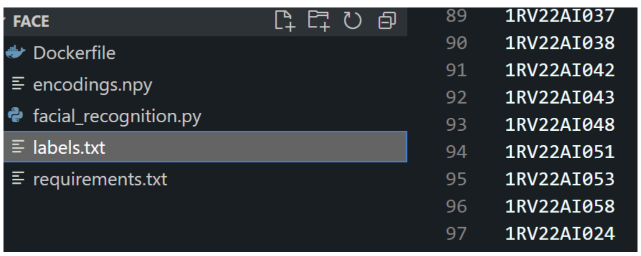

Fig. 5: Labels for Encodings

Referring to Fig. 5, this figure shows the labels.txt file, which stores the IDs of all the terrorists present in the database accordingly.

V. Results

After implementation, we observed the following results. Fig. 6 shows that the sample input image taken matches the existing image database and hence, the relative ID of the image stored as a label, is returned as a result.

Correct Match	Unknown Match	Wrong Match
114	8	4

Based on the above results, we can see that choice of 0.45 as an embedding threshold is a good estimate, as we achieve an accuracy of 90.47% in a constrained system. Before this, the accuracy of the model was quite low due to images collected in the database not matching or following up with the constraints (for example, resolution of the image, dimensions of the image, background, angle, mode(portrait, landscape), etc.) as mentioned. Accuracy improved from 54.43% to 85.67% and then when the underfitting issue was resolved, accuracy jumped to the current accuracy of 90.47% even un the constrained system.

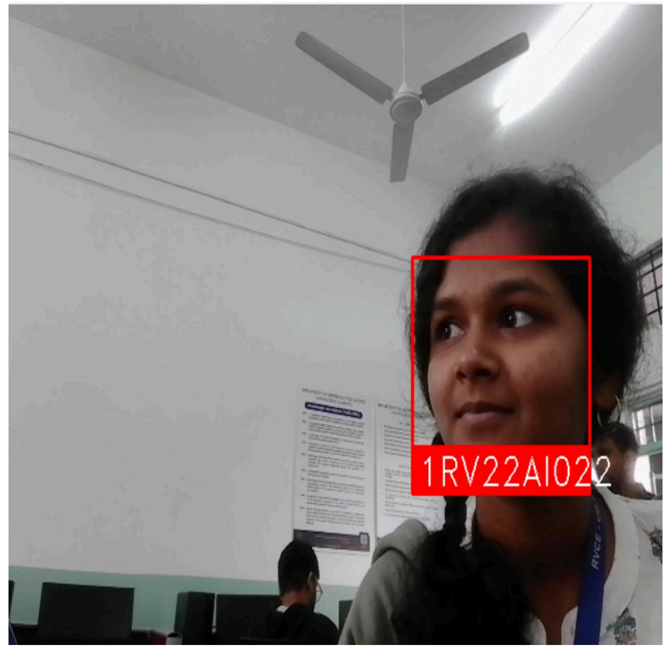

Fig. 6: Facial Recognition at work

VI. Conclusion

The developed Airport Management System is a highly efficient and scalable platform that integrates key functions to improve overall airport operations. Not just that, the proposed Airport Management System focuses centrally on security enhancement through advanced face recognition technology. This system integrates a robust face recognition module to ensure the safety and protection of passengers and staff. By capturing and analyzing facial data, the system proactively identifies individuals and verifies their iden- tities against a secure database, preventing unauthorized access and suspicious activities. This advanced security measure creates a safer airport environment by streamlining pas- senger verification at check-ins, boarding gates, and restricted areas. The user-friendly interface ensures seamless interaction while maintaining a high level of security, making the Airport Management System an effective, security-centric platform.

References

[1] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991, doi: 10.1162/jocn.1991.3.1.71.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997, doi: 10.1109/34.598228.

[3] X. Wang, J. Peng, S. Zhang, B. Chen, Y. Wang, and Y. Guo, "A Survey of Face Recognition." [Online]. Available: https://arxiv.org/abs/2212.13038

[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[5] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, Springer International Publishing, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[6] D. Zhang, J. Li, and Z. Shan, “Implementation of Dlib Deep Learning Face Recognition Technology,” in *2020 International Conference on Robots & Intelligent System (ICRIS)*, 2020, pp. 88–91. doi: 10.1109/ICRIS52159.2020.00030.

[7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, pp. 886–893. doi: 10.1109/CVPR.2005.177.

[8] L. Li, X. Mu, S. Li, and H. Peng, “A review of face recognition technology,” *IEEE access*, vol. 8, pp. 139110–139120, 2020.

[9] D. Han, Y. Li, and J. Denzler, “KAN See Your Face.” [Online]. Available: https://arxiv.org/abs/2411.18165

[10] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li, and X. Tang, “EFR-CSTP: Encryption for face recognition based on the chaos and semi-tensor product theory,” *Information Sciences*, vol. 621, pp. 766–781, 2023, doi: https://doi.org/10.1016/j.ins.2022.11.121.

[11] W. Yu and W. Wei, “Local and Global Feature Attention Fusion Network for Face Recognition.” [Online]. Available: https://arxiv.org/abs/2411.16169

[12] H. O. Shahreza and S. Marcel, “HyperFace: Generating Synthetic Face Recognition Datasets by Exploring Face Embedding Hypersphere.” [Online]. Available: https://arxiv.org/abs/2411.08470

[13] G. Jeevan, G. C. Zacharias, M. S. Nair, and J. Rajan, “An empirical study of the impact of masks on face recognition,” *Pattern Recognition*, vol. 122, p. 108308, 2022, doi: https://doi.org/10.1016/j.patcog.2021.108308.

[14] Y. Li, K. Guo, Y. Lu, and L. Liu, “Cropping and attention based approach for masked face recognition,” *Applied Intelligence*, vol. 51, no. 5, pp. 3012–3025, 2021, doi: 10.1007/s10489-020-02100-9.

[15] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, “MagFace: A Universal Representation for Face Recognition and Quality Assessment,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 14225–14234.

[16] E. M. Onyema, P. K. Shukla, S. Dalal, M. N. Mathur, M. Zakariah, and B. Tiwari, “Enhancement of Patient Facial Recognition through Deep Learning Algorithm: ConvNet,” *Journal of Healthcare Engineering*, vol. 2021, no. 1, p. 5196000, 2021, doi: https://doi.org/10.1155/2021/5196000.

[17] S. I. Serengil and A. Ozpinar, “LightFace: A Hybrid Deep Face Recognition Framework,” in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2020, pp. 1–5. doi: 10.1109/ASYU50717.2020.9259802.

[18] M. Alansari, O. A. Hay, S. Javed, A. Shoufan, Y. Zweiri, and N. Werghi, “GhostFaceNets: Lightweight Face Recognition Model From Cheap Operations,” *IEEE Access*, vol. 11, no. , pp. 35429–35446, 2023, doi: 10.1109/ACCESS.2023.3266068.

[19] C. A. Antipona, R. R. Magsino, R. M. Dioses, and K. E. Mata, “An Enhancement of Haar Cascade Algorithm Applied to Face Recognition for Gate Pass Security.” [Online]. Available: https://arxiv.org/abs/2411.03831

[20] T. Chettaoui, N. Damer, and F. Boutros, “FRoundation: Are Foundation Models Ready for Face Recognition?.” [Online]. Available: https://arxiv.org/abs/2410.23831

[21] S. Li, J. Xu, X. Xu, P. Shen, S. Li, and B. Hooi, “Spherical Confidence Learning for Face Recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 15629–15637.

[22] H. Qiu, D. Gong, Z. Li, W. Liu, and D. Tao, “End2End Occluded Face Recognition by Masking Corrupted Features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6939–6952, 2022, doi: 10.1109/TPAMI.2021.3098962.

[23] Z. Zhang, “Technologies Raise the Effectiveness of Airport Security Control,” in *2019 IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 2019, pp. 431–434. doi: 10.1109/ICCASIT48058.2019.8973152.

[24] S. Bouyakoub, A. Belkhir, F. M. Bouyakoub, and W. Guebli, “Smart airport: an IoT-based Airport Management System,” in *Proceedings of the International Conference on Future Networks and Distributed Systems*, in ICFNDS '17. Cambridge, United Kingdom: Association for Computing Machinery, 2017. doi: 10.1145/3102304.3105572.

[25] D. Semedo, D. Carmo, R. Padnevych, and J. Magalhaes, “Contact-free Airport Borders with Biometrics-on-the-Move,” in *2021 IEEE International Workshop on Biometrics and Forensics (IWBF)*, 2021, pp. 1–2. doi: 10.1109/IWBF50991.2021.9465075.

[26] D. Almeida, K. Shmarko, and E. Lomas, “The ethics of facial recognition technologies, surveillance, and accountability in an age of artificial intelligence: a comparative analysis of US, EU, and UK regulatory frameworks,” *AI and Ethics*, vol. 2, no. 3, pp. 377–387, 2022, doi: 10.1007/s43681-021-00077-w.

[27] L. Li, X. Mu, S. Li, and H. Peng, “A Review of Face Recognition Technology,” *IEEE Access*, vol. 8, no. , pp. 139110–139120, 2020, doi: 10.1109/ACCESS.2020.3011028.