RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Go, change the world®

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Project Report on

Face Recognition in Airport Management Systems

Submitted in partial fulfilment of the requirements for the V Semester ***ARTIFICIAL NEURAL NETWORK AND DEEP LEARNING***

AI253IA

By

Ashwin Ajoy Dharmavaram	1RV22AI011
Keerti Patil	1RV22AI022

Department of Artificial Intelligence and Machine Learning
RV College of Engineering®
Bengaluru – 560059
January 2025

RV College of Engineering®, Bengaluru

(Autonomous institution affiliated to VTU, Belagavi)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Bengaluru-560059

CERTIFICATE

This is to certify that the project titled ***Face Recognition in Airport Management Systems*** submitted in partial fulfillment of Artificial Neural Networks and Deep Learning (AI253IA) of V Semester BE is a result of the bonafide work carried out by **Ashwin Ajoy Dharmavaram** (**1RV22AI011**) and **Keerti Patil** (**1RV22AI022**) during the Academic year 2024-25.

Date:

Dr. Somesh Nandi	**Dr. B. Sathish Babu**
Course Coordinator	**Professor & HOD**
Department of AI & ML	**Department of AI & ML**

RV College of Engineering®, Bengaluru

(Autonomous institution affiliated to VTU, Belagavi)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE
AND
MACHINE LEARNING

Bengaluru-560059

DECLARATION

We, **Ashwin Ajoy Dharmavaram** (**1RV22AI011**)and **Keerti Patil** (**1RV22AI022**), students of fifth Semester BE hereby declare that the Project titled ***Face Recognition in Airport Management Systems*** has been carried out and completed successfully by us and is our original work.

Date:

Ashwin Ajoy Dharmavaram ________________

Keerti Patil ________________

ACKNOWLEDGEMENT

We are indebted to our guide, **Dr. Somesh Nandi**, Assistant Professor, RV College of Engineering . for the wholehearted support, suggestions and invaluable advice throughout our project work and also helped in the preparation of this report.

We express our sincere gratitude to our Head of the Department Dr. B Satish Babu, our Associate Professor Dr. S Anupama Kumar, and our beloved Principal, Dr. K. N Subramanya for their guidance and appreciation towards this Experiential Learning Project undertaken.

Finally, we take this opportunity to thank our family members and friends who provided all the backup support throughout the project work.

ABSTRACT

The Airport Management System is a security-centric platform designed to redefine airport operations with advanced safety measures and seamless integration. At its core is a cutting-edge face recognition module that ensures robust passenger verification at critical checkpoints, including check-ins, boarding gates, and restricted areas. By analyzing facial data and cross-referencing it with a secure database, the system proactively identifies individuals, verifies their identities, and prevents unauthorized access or suspicious activities. These measures significantly enhance airport security, creating a safe and trustworthy environment for passengers and staff.

The system prioritizes security while maintaining operational efficiency and streamlining processes without compromising safety. Its real-time monitoring and threat detection capabilities allow for quick responses to potential risks, further strengthening the protective measures in place. With a scalable and user-friendly design, the Airport Management System ensures ease of integration into existing infrastructures and adapts to the dynamic needs of modern airports. By focusing on advanced security technologies, this platform sets a new standard for safeguarding airport environments while delivering a seamless and secure travel experience.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Description

The airport management system today is often inefficient and lacks coordination, leading to frequent delays, confusion, and uncertainty for passengers, which disrupts their schedules. To address these issues, we propose a streamlined Airport Management System that integrates and organizes data, allowing for efficient management of flight schedules, passenger information, and real-time updates primarily focuses on the surveillance at the airport.

Face recognition is implemented using dlib, a part of the face recognition library, and face recognition models. The system leverages CNN-based modeling to identify and verify individuals in real time accurately. This technology is embedded into an edge device, specifically the Raspberry Pi 4B, allowing for seamless integration with a terrorist database to increase surveillance at the airport. The system monitors not just passengers, but everyone present at the airport, enhancing overall security.

The edge device processes data locally, ensuring faster recognition and response times, and integrates with airport security systems for a more comprehensive surveillance solution. This approach minimizes the risk of potential security threats and provides a proactive solution to identify and track individuals of interest.

1.2 Report Organization

The report is organized into seven main sections. It begins with an introduction that outlines the project description and the organization of the report. The literature review includes the literature survey, and existing and proposed systems and highlights the research gaps. Next is the software requirements specification, which details the general, hardware, and software requirements, functional and non-functional requirements, external interfaces, and design constraints. The system design section describes the architectural design, various levels of data flow diagrams, and the ANN/DL architecture or algorithm used. The implementation section provides code snippets and discusses the results. The report also concludes with a conclusion summarizing the work and a section on future enhancements emphasizing on potential improvements in the upcoming time.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Survey

The paper [1] 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, [2] 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, [3] proposes a system for encryption for facial images. [4] 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 [5] 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”. [6] 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. [7] 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. [8] 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. [9] 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.

[10] 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. [11] 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 [12] 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.

[13] 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. [14] states that due to their dependence on deterministic embeddings, noisy images are mapped into poorly learned regions of space which leads to inaccuracies. [15] 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.

[16] 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 passengers at the airport are focused, research has been undertaken for the implementation of the respective technologies.

[17] describes the use of Contactless Smart Cards (CSCs) at airports in conjunction with fingerprints. [18] describes the scope of smart airports with an IoT paradigm and hence, proposes to implement an IoT-integrated smart airport management system focusing on automating the passengers' baggage checking, improved flight services, etc.

[19] 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. [20] highlights the importance of facial recognition for the safety of passengers at airports to reduce crimes and increase surveillance taking GDPR (General Data Protection Regulations) and data acquisition into consideration.

2.2 Existing and Proposed System (Research Gaps)

Historical methods of facial recognition, such as those based on facial geometry, were explored but not implemented before the advent of deep learning. Modern models like FaceNet have achieved remarkable accuracy, with FaceNet attaining the highest at 99.63% among 26 evaluated models. Advancements in generative AI underscore the importance of robust data security, particularly when dealing with facial embeddings. Research into encryption methods for facial images and improvements in recognizing faces under challenging conditions, such as poor lighting or occlusion, has led to innovations like Local and Global Feature Attention Fusion networks. Synthetic dataset generation, addressed as an optimization problem by models like HyperFace, tackles the issue of limited training data. Efforts to recognize masked faces during the COVID-19 pandemic have included cropping techniques combined with Convolutional Block Attention Modules (CBAM). MagFace introduces a loss function where embedding magnitude reflects the quality of face recognition, enhancing system performance. Lightweight neural networks demonstrate close-to-state-of-the-art accuracy for facial expression recognition with fewer layers, while Ghost modules extract comprehensive features through linear transformations. Adjustments to traditional methods, such as modifying the Haar_Cascade algorithm for RGB images, significantly boost accuracy and recall. Foundation Models trained on diverse, large-scale datasets show comparable versatility and performance to smaller custom models. Innovations in facial recognition also explore spherical embedding spaces to reduce inaccuracies in noisy data and train models to predict occluded regions, improving robustness in real-world applications. At airports, CT technology and behavioral biometrics optimize processes like baggage verification, integrating facial and retinal scans. Contactless Smart Cards combined with fingerprints enhance security, while IoT-integrated systems automate baggage checks and passenger services. During the COVID-19 pandemic, contactless boarding ensured hygiene and security through advanced fraud detection systems. Facial recognition has become a cornerstone of airport security, balancing enhanced surveillance with compliance to GDPR and privacy considerations.

2.3 Objectives

The objectives of the proposed work is to:

1. Design and implement a streamlined Airport Management System that aims at integrating and organizing data, allowing for efficient management of flight schedules, passenger information, and real-time updates.

2. Enhance efficiency in coordination and scheduling of the flights.

3. Achieve minimum delays, enhancing passenger experience and improving overall operational efficiency at airports.

4. Integrate face navigation of the passengers using drones to verify their identification to ensure the security of passengers.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Introduction

The proposed Airport Management System is designed to enhance overall airport operations by integrating key functionalities. At its core, the system focuses on improving security through advanced face recognition technology. By incorporating a robust face recognition module, the system ensures the safety and protection of passengers and staff. It captures and analyzes facial data to proactively identify individuals and verify their identities against a secure database, thereby preventing unauthorized access and detecting suspicious activities. This innovative approach enhances the airport environment by streamlining passenger verification processes at check-ins, boarding gates, and restricted areas. With a user-friendly interface, the system ensures seamless interaction while maintaining a high level of security, establishing itself as an essential, security-centric platform for modern airport management.

3.2 General Requirement

3.2.1 Hardware Requirement

1. Raspberry Pi 4 B

2. Raspberry PI 5MP Camera

3. Server (for DBMS, Application, and Web Interface)
   - CPU: Dual-core processor
   - RAM: 8 GB
   - Storage: 50 GB
   - Network: Standard Ethernet or Wi-Fi connection
   - Operating System: Linux(Ubuntu 18.04LTS+) or Windows 10+

4. Testing Machine (Client side)
   - CPU: Single-core processor
   - RAM: 2 GB
   - Storage: 100 MB of free space (browser cache).
   - Operating System: Any OS that supports Python, MySQL, and web frameworks (Linux or Windows)
   - Browsers: Chromium 49+, Microsoft Edge v90+

3.3 Software Requirement

1. Python: Version 3.12.4
2. Flask: 3.0.3
3. face-recognition==1.3.0
4. opencv-python-headless
5. numpy==1.21.2
6. dlib==19.24.2
7. Psycopg2: 2.9.10
8. PostgreSQL: Version 16+

3.4 Functional Requirement

1. **Security and Surveillance:**

   Input: The system implements advanced face recognition technology using the dlib from face recognition models of the face recognition library. This system is deployed on an edge device, specifically, the Raspberry Pi 5B, integrated with a comprehensive terrorist database.

   Process:

   - Continuous monitoring and surveillance of every member present in the airport, including passengers, staff, and visitors.
   - Real-time face recognition to identify individuals flagged in the terrorist database or individuals exhibiting suspicious behavior.
   - Integration with AI-based anomaly detection systems for enhanced threat identification.
   - Alerts are generated and relayed to security personnel immediately upon detecting a match with the terrorist database or other security risks.

   Output:

   - Improved security through rapid identification of threats.
   - Proactive prevention of potential terrorist activities at the airport.
   - Comprehensive surveillance that ensures the safety of all individuals present at the airport.

3.5 Non-Functional Requirements

1. **Usability** - Focus to build an intuitive interface to ensure ease of usage for airport staff and minimal training for new users/viewers.
2. **Scalability** - Manages huge increase in data and load enabling for ease in maintenance, even when the operations like bookings like hotels near airports, etc increase.
3. **Security** - Prevents unauthorized access and increases surveillance, by keeping check on people at the airports.
4. **Reliability** - Consistency in services like canceling, booking, delaying etc. , managing and updating data efficiently and keeping data sensitivity intact as promised to reduce downtime.
5. **Compatibility** - Designed for smooth coordination with the other third - party organizations, collaborations etc. or with the airport infrastructure.
6. **Availability** - High availability and backup systems ensured to keep critical operations like booking flights, etc. open almost all the time.
7. **Reduced Latency** - Overall Efficiency improved due to better responsiveness, the average response time of the website being less than 3 seconds.

3.6 Design Constraints

Different possible constraints possible in implementing the proposed system with respect to design constraints could be of different types as follows:

1. Hardware design constraints:
    - Face recognition processing must be optimized for edge devices with restricted power and memory capacities.
    - Cameras used for capturing images must provide sufficient resolution and frame rates to ensure accuracy in different lighting and environmental conditions.
2. Performance design constraints:
    - To Maintain the accuracy of the model.

- To Update the terrorist data on which the model is trained.

3. Cost design constraints:

   - Hardware and the respective system deployment could be expensive and not budget-friendly.

4. Regulatory and Compliance design constraints:

   - Of Adhering and abiding by aviation security protocols and airport standards set at the airport.
   - Compliance with international and regional existing biometric data usage policies.

CHAPTER 4

SYSTEM DESIGN

4.1 Architectural Design

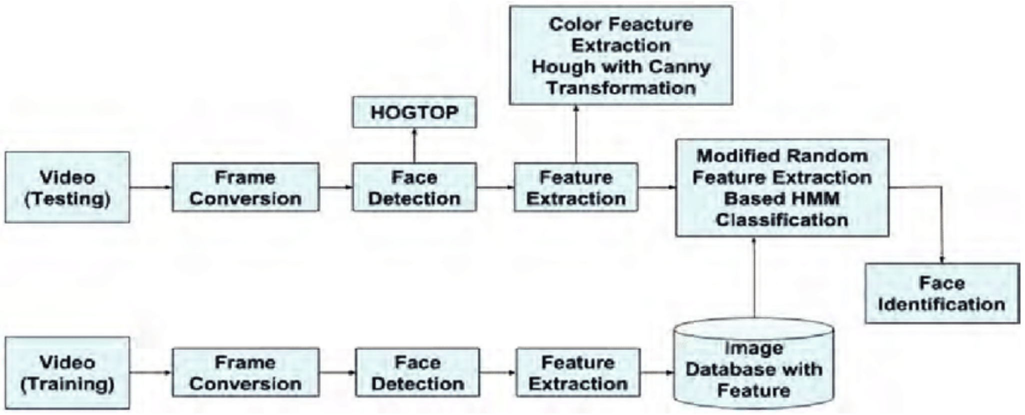


Figure 4.1: Dlib architecture

Dlib is a cutting-edge C++ toolkit that provides a large selection of machine-learning tools and methods for developing intricate software solutions. It is used in many different fields, including as high-performance computer settings, mobile phones, embedded devices, and robots. Any program can utilize Dlib for free because of its open-source licensing. Dlib's architecture is designed to be high-quality and portable, with comprehensive documentation and extensive unit test coverage. It is regularly tested on MS Windows, Linux, and macOS systems, ensuring broad compatibility. The library is implemented in pure ISO standard C++ and requires no additional packages, making it straightforward to integrate into various projects.

Key features of Dlib include:

- Dlib provides a range of machine learning algorithms, such as deep learning tools, support vector machines, clustering algorithms, etc.
- It offers a fast matrix object capable of using BLAS and LAPACK libraries when available, along with numerous linear algebra and mathematical operations.
- Dlib includes algorithms for exact and approximate inference in Bayesian networks, as well as routines for MAP inference in various graph structures.

- The library provides tools for reading and writing common image formats, automatic color space conversion, and common image operations such as edge finding and morphological operations.

- Dlib offers a portable and simple threading API, a message passing pipe for inter-thread and inter-process communication, and a portable and simple TCP sockets API.

- It provides a portable and simple core GUI API, implemented on top of which are numerous widgets.

- Dlib includes a CRC 32 object, MD5 functions, and various abstracted objects representing parts of data compression algorithms.

- The library offers a thread-safe logger object, a modular unit testing framework, and various assert macros useful for testing preconditions.

4.2 Data Flow Diagram

4.2.1 Data Flow Diagram Level 0

4.2 includes an overview of the entire system proposed.

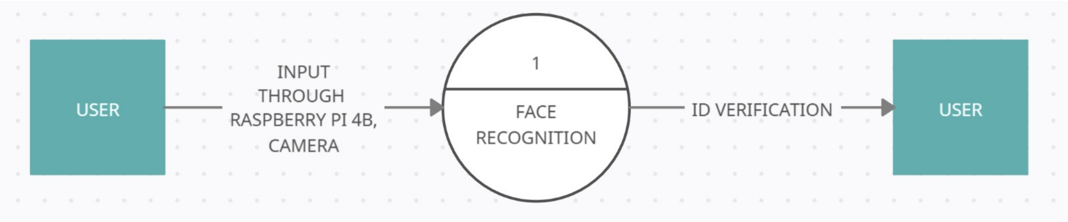


Figure 4.2: Level 0 of Data Flow Diagram

This explains that the faces of the people are taken as inputs, encoded, and vector embeddings are made accordingly, which are compared to the embeddings of the terrorist database through the dlib deep-learning model from face recognition models of the face recognition library, to successfully verify the identification of the passengers in the airport.

4.2.2 Data Flow Diagram Level 1

4.3 gives information on different processes possible in the system. Possible functionalities:

1. Face Recognition: Explains how the identity verification process is carried out in

real time.

2. Vector Embeddings: Explains how deep learning is implemented through dlib of the face recognition library.

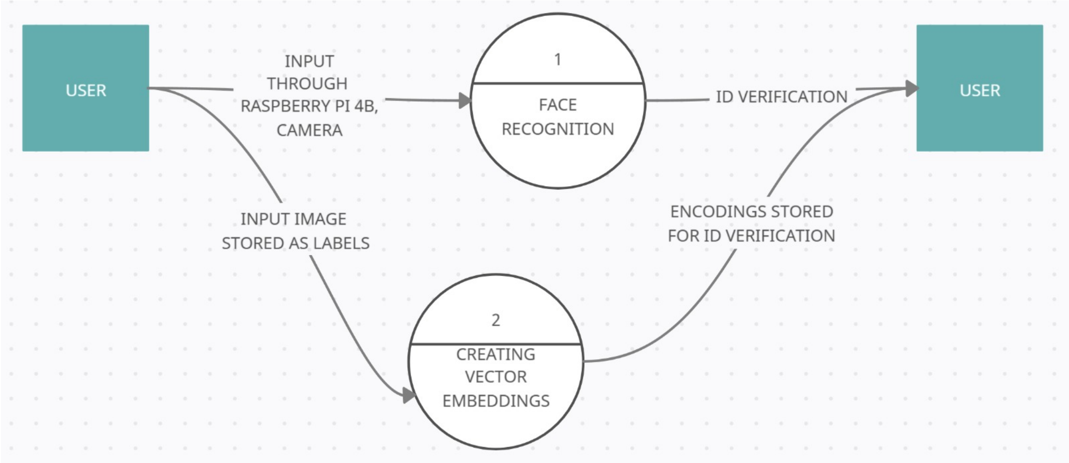

USER
INPUT THROUGH RASPBERRY PI 4B, CAMERA
1
FACE RECOGNITION
ID VERIFICATION
USER
INPUT IMAGE STORED AS LABELS
ENCODINGS STORED FOR ID VERIFICATION
2
CREATING VECTOR EMBEDDINGS

Figure 4.3: Level 1 of Data Flow Diagram

4.2.3 Data Flow Diagram Level 2

Each functionality or process mentioned in the figure 4.4 is broken down into several sub-processes or sub-functionalities in order as follows:

1. Face Recognition:

   - Input Image of the Person Through Camera (1.1): The process begins with capturing an image of the individual through a camera. This image is then preprocessed to ensure it meets the requirements of the recognition model, such as resizing or normalization, before being loaded into the system.

   - Comparing to Existing Encodings (1.2): The embeddings generated from the input image is compared against a database of pre-stored embeddings. This step involves calculating the similarity between the input embedding and stored ones, using a predefined threshold to identify potential matches.

   - Face Recognition (1.3): If a stored embedding matches the input embedding with sufficient similarity, the system successfully recognizes the face and gives an alert. If no match is found, it concludes that the face is unrecognized.

2. Vector Embeddings:

- Images Collected of Terrorists with Constraints of Dimension 250 × 250 (2.1): The system starts by gathering a dataset of images of individuals. These images are processed to meet specific constraints, such as resizing them to a uniform dimension of 250 × 250 pixels, ensuring consistency for further processing.

- Stored Encodings (2.2): The generated embeddings are stored in a structured format, as encodings, using a reliable storage mechanism. These stored encodings form the database against which input images will be compared for recognition.

- Stored in Encodings.npy: The face encodings are saved in a file named encodings.npy. This file acts as a centralized database of facial embeddings, enabling quick and efficient access during the recognition and verification processes.

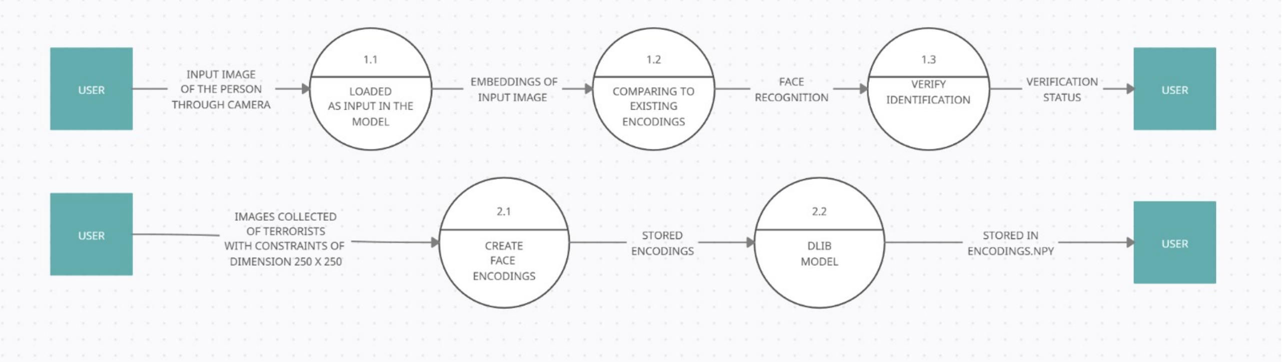

USER
INPUT IMAGE OF THE PERSON THROUGH CAMERA
1.1
LOADED AS INPUT IN THE MODEL
EMBEDDINGS OF INPUT IMAGE
1.2
COMPARING TO EXISTING ENCODINGS
FACE RECOGNITION
1.3
VERIFY IDENTIFICATION
VERIFICATION STATUS
USER
USER
IMAGES COLLECTED OF TERRORISTS WITH CONSTRAINTS OF DIMENSION 250 X 250
2.1
CREATE FACE ENCODINGS
STORED ENCODINGS
2.2
DLIB MODEL
STORED IN ENCODINGS.NPY
USER

Figure 4.4: Level 2 of Data Flow Diagram

4.3 Description of the ANN/DL Architecture/Algorithm Used

A pre-trained deep learning model from the face recognition library is used here. It leverages the HOG (Histogram of Oriented Gradients) and CNN (Convolutional Neural Network) models, both of which are part of the face recognition library.

- The system uses HOG (Histogram of Oriented Gradients) for efficient face detection by analyzing edges and gradients. For better accuracy, it can switch to CNN (Convolutional Neural Network), which performs better under challenging conditions like varying angles or lighting.

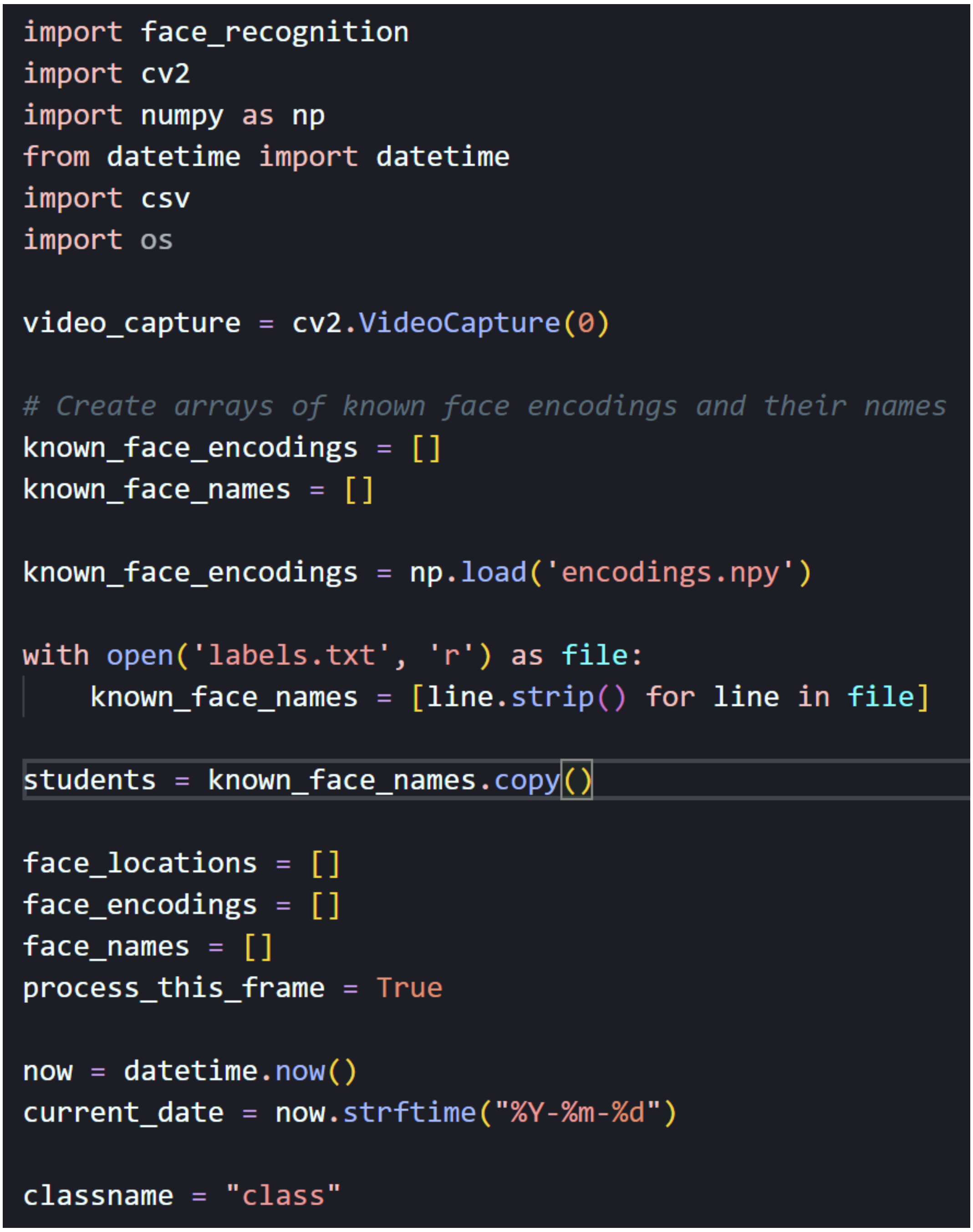
- Detected faces are converted into a 128-dimensional vector (embedding) using dlib. These embeddings capture unique facial features and are stored for comparison with future faces.
- The system compares the generated embeddings with stored ones using Euclidean distance or cosine similarity. A match is identified if the distance is minimal, and the associated name is retrieved for recognition.
- The system captures and processes video frames in real time. Detected faces are recognized by comparing embeddings, and attendance is logged with the person's name and entry time when a match is found.

CHAPTER 5

IMPLEMENTATION

This section gives the explanantion of the implementation of the deep learning model dlib in the proposed project.

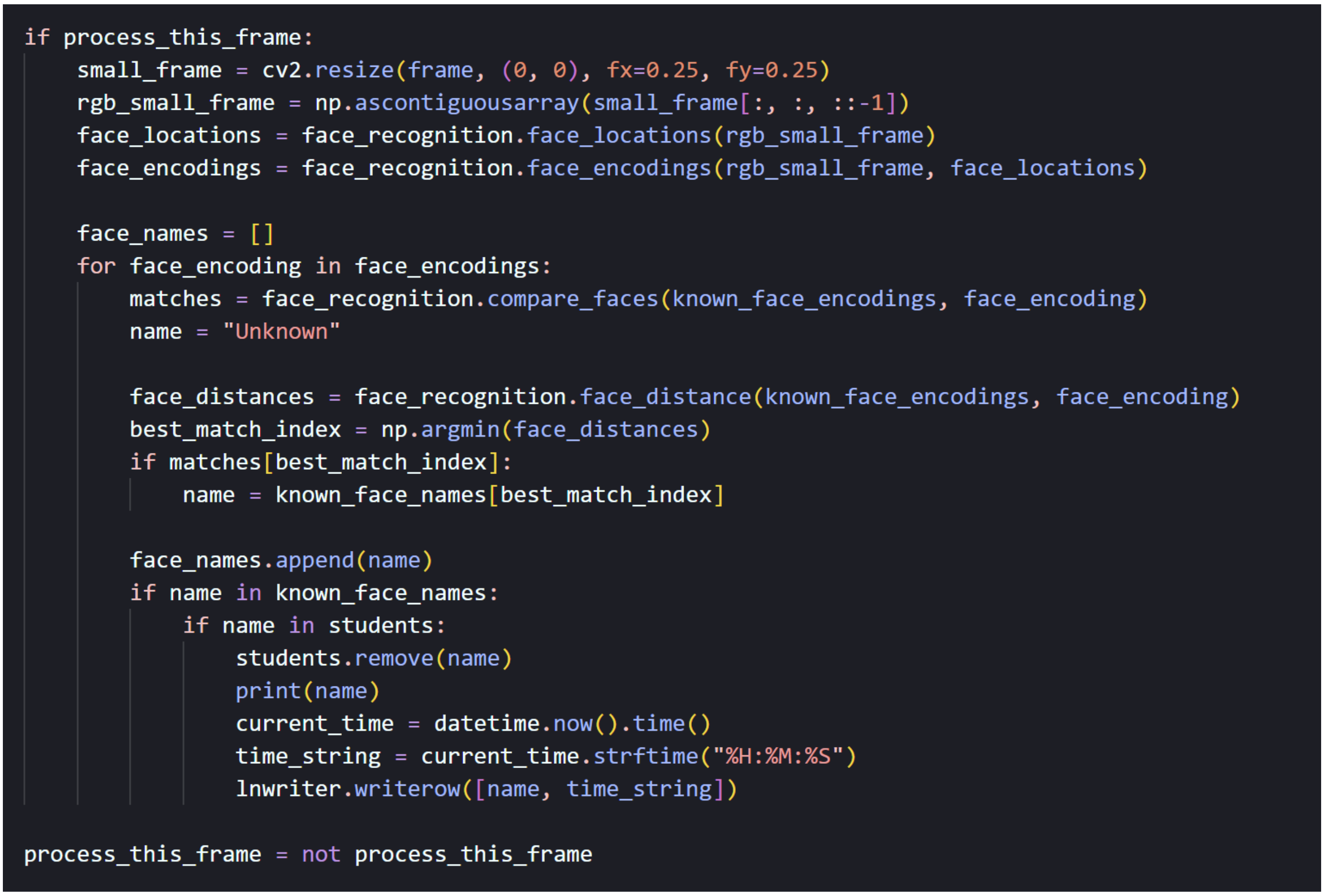
5.1 Code Snippets



```
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"
```

Figure 5.1: Code for Image Encodings

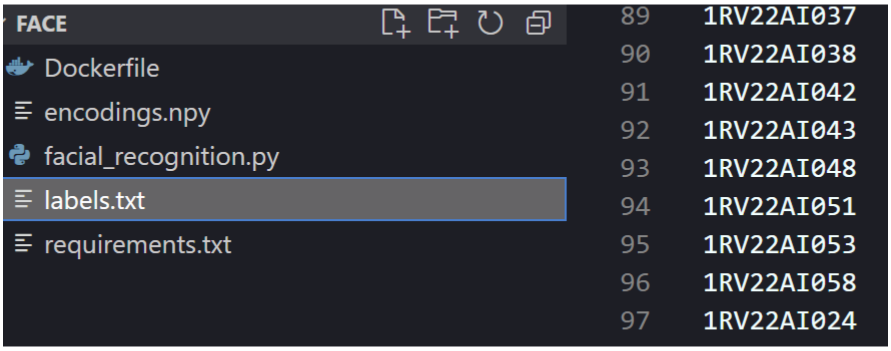
5.1 creates arrays of known face encodings (after creating the encodings) of the terrorist images with the names (ID's) accordingly in the labels.txt file.



```
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
```

Figure 5.2: Comparison of Encodings

5.2 compares the encodings existing of 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.



FACE

Dockerfile

encodings.npy

facial_recognition.py

labels.txt

requirements.txt

```
1RV22AI037
1RV22AI038
1RV22AI042
1RV22AI043
1RV22AI048
1RV22AI051
1RV22AI053
1RV22AI058
1RV22AI024
```

Figure 5.3: Labels for Encodings

5.3 shows the labels.txt file which stores the IDs of all the terrorists present in the database accordingly.

5.2 Results & Discussion

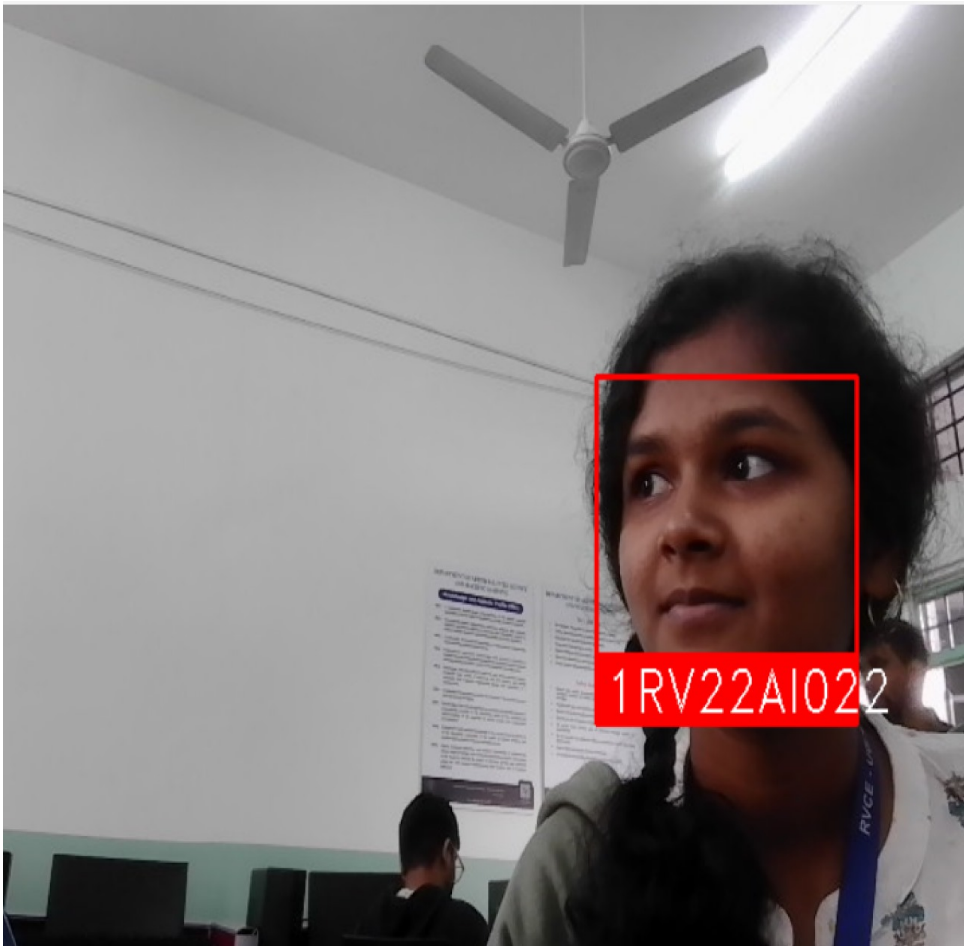
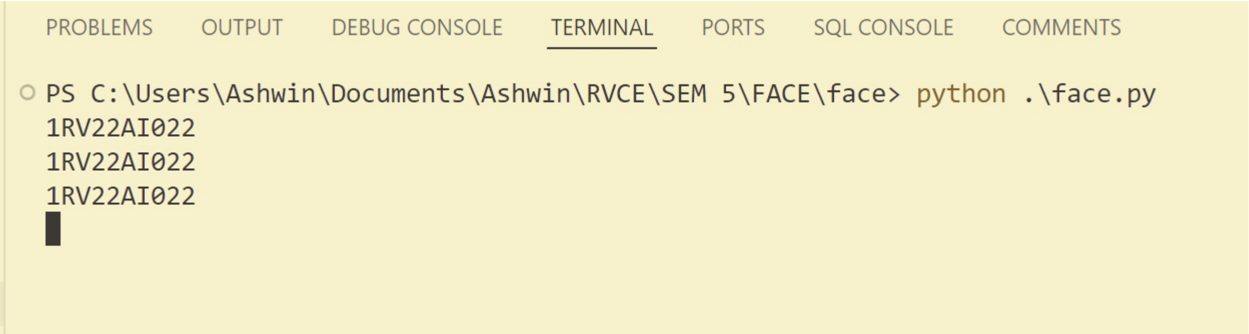


Figure 5.4: Detection of the Face



```
PS C:\Users\Ashwin\Documents\Ashwin\RVCE\SEM 5\FACE\face> python .\face.py
1RV22AI022
1RV22AI022
1RV22AI022
```

Figure 5.5: Face Identified

5.4 and 5.5 figures show how the faces are detected which is followed by matching the face encodings with the existing encodings in the terrorist database to return the ID of the terrorist in the terminal, which can be further stored in the form of csv with the relative timestamps.

CHAPTER 6

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 identities against a secure database, preventing unauthorized access and suspicious activities. This advanced security measure creates a safer airport environment by streamlining passenger 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.

CHAPTER 7

FUTURE ENHANCEMENTS

Future Enhancements of the proposed work include:

- Embedding the face recognition model in an **UAV** (Unmanned Aerial Vehicle) like a drone, to enhance surveillance at the airport by verifying the identification of every individual present at the airport besides passengers.
- UAVs equipped with face recognition provide real-time surveillance over large airport areas, identifying and tracking individuals of interest from elevated positions, especially in remote or crowded zones.
- They autonomously detect and match faces against terrorist databases, triggering alerts to security teams for immediate intervention in case of suspicious behavior or potential threats.
- The system ensures privacy by focusing on public areas, using encrypted data transmission, and complying with ethical standards and regulations to prevent misuse of biometric data.
- Multiple UAVs can collaborate for enhanced coverage, sharing data with central security systems and coordinating responses to potential threats, ensuring comprehensive airport surveillance.

BIBLIOGRAPHY

[1] L. Li, X. Mu, S. Li, and H. Peng, "A review of face recognition technology," *IEEE access*, vol. 8, pp. 139 110–139 120, 2020.

[2] D. Han, Y. Li, and J. Denzler, *Kan see your face*, 2024. arXiv: `2411.18165 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2411.18165`.

[3] 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, ISSN: 0020-0255. DOI: `https://doi.org/10.1016/j.ins.2022.11.121`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0020025522014323`.

[4] W. Yu and W. Wei, *Local and global feature attention fusion network for face recognition*, 2024. arXiv: `2411.16169 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2411.16169`.

[5] H. O. Shahreza and S. Marcel, *Hyperface: Generating synthetic face recognition datasets by exploring face embedding hypersphere*, 2024. arXiv: `2411.08470 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2411.08470`.

[6] 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. 108 308, 2022, ISSN: 0031-3203. DOI: `https://doi.org/10.1016/j.patcog.2021.108308`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S003132032100488X`.

[7] 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, ISSN: 1573-7497. DOI: `10.1007/s10489-020-02100-9`. [Online]. Available: `https://doi.org/10.1007/s10489-020-02100-9`.

[8] 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)*, 2021, pp. 14 225–14 234.

[9] 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: Con-

vnet," *Journal of Healthcare Engineering*, vol. 2021, no. 1, p. 5 196 000, 2021. DOI: https://doi.org/10.1155/2021/5196000. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/5196000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/5196000.

[10] 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.

[11] 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, pp. 35 429–35 446, 2023. DOI: 10.1109/ACCESS.2023.3266068.

[12] 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*, 2024. arXiv: 2411.03831 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2411.03831.

[13] T. Chettaoui, N. Damer, and F. Boutros, *Froundation: Are foundation models ready for face recognition?* 2024. arXiv: 2410.23831 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2410.23831.

[14] 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)*, 2021, pp. 15 629–15 637.

[15] 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.

[16] 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.

[17] M. David, G. Hussein, and K. Sakurai, "Secure identity authentication and logical access control for airport information systems," English, Proceedings: 37th Annual 2003 International Carnahan Conference on Security Technology ; Conference date: 14-10-2003 Through 16-10-2003, 2003, pp. 314–320.

[18] 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*, ser. ICFNDS '17, Cambridge, United Kingdom: Association for Computing Machinery, 2017, ISBN: 9781450348447. DOI: `10.1145/3102304.3105572`. [Online]. Available: `https://doi.org/10.1145/3102304.3105572`.

[19] 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`.

[20] 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, ISSN: 2730-5961. DOI: `10.1007/s43681-021-00077-w`. [Online]. Available: `https://doi.org/10.1007/s43681-021-00077-w`.