

**SYNOPSIS REPORT ON
REAL-TIME FACE ATTENDANCE SYSTEM
USING COMPUTER VISION**

A Synopsis Report Submitted in the partial fulfilment of the Requirements for the Degree
Of

MASTER IN COMPUTER APPLICATION

By

AKASH NAUTIYAL (2201042)

MCA III SEM

Under the Guidance of

Mr. Neeraj Panwar

Associate Professor



**DEPARTMENT OF COMPUTER APPLICATION
GRAPHIC ERA HILL UNIVERSITY**

ACKNOWLEDGMENT

I wish to thank our parents for their continuing support and encouragement. I also wish to thank them for providing us with the opportunity to reach this far in studies.

I would like to thank particularly our project Co-ordinator and our Project Guide **Mr. NEERAJ PANWAR** for his patience, support, and encouragement throughout the completion of this project and having faith in me.

At last, but not the least I greatly indebted to all other persons who directly or indirectly helped me during this work.

AKASH NAUTIYAL
Roll No.- 2201042
Class Roll No.-10
MCA III Sem (Sec-F)
Session: 2023-2024
GEHU Dehradun

TABLE OF CONTENTS

1. ABSTRACT
2. PROJECT INTRODUCTION
3. OBJECTIVE
4. DATABSE
5. METHODOLOGY
6. TOOLS/SOFTWARE USED
7. REFERENCES

ABSTRACT

A face recognition attendance system is a biometric technology that uses artificial intelligence to identify and verify people based on their facial characteristics. It's a fast, high-accuracy system that's used in various sectors, including finance, retail, government, and industry.

Attendance systems have evolved significantly over the years, with the advent of technology providing innovative solutions to streamline the process. Facial recognition has emerged as a promising technology in this domain due to its accuracy, efficiency, and non-intrusiveness. This paper presents an efficient real-time attendance system employing facial recognition technology. Traditional attendance systems often suffer from inaccuracies, buddy punching, and time-consuming manual processes. Leveraging advancements in computer vision and machine learning, the proposed system aims to address these challenges by automating attendance management through facial recognition.

The system operates by capturing facial images using a camera and employing Machine learning algorithms to detect, extract, and analyze facial features. Utilizing Machine Learning Algorithm, we develop Face encoding of the captured images and then compare it with encoding present in database of the stored images. The system's architecture encompasses both hardware and software components, including camera modules, processing units, and a centralized database.

The proposed system has been evaluated through extensive testing and validation, demonstrating promising results in terms of accuracy, speed, and reliability. Furthermore, the system's adaptability and scalability make it suitable for various industries, including education, corporate environments, and public institutions.

In conclusion, the real-time attendance system utilizing facial recognition technology presents a significant advancement in attendance management, offering a reliable, efficient, and secure solution for organizations seeking to streamline their attendance tracking processes.

INTRODUCTION

The system utilizes computer vision, a field of artificial intelligence, to automate the attendance process by recognizing and verifying individuals based on their facial features in real-time. The primary components and functionalities of the system include:

Face Detection: Employing robust algorithms to identify and locate human faces within images or video streams. This initial step is crucial for subsequent processing.

Facial Feature Extraction: Utilizing computer vision techniques to extract unique facial features, such as the distance between eyes, shape of the nose, and contours of the face. These features are translated into mathematical representations for recognition.

Real-Time Processing: The system operates in real-time, enabling immediate recognition and verification of individuals as they enter or interact with the system. This ensures swift attendance recording without causing delays or disruptions.

Database Management: Storing and managing a database of known faces or templates against which incoming faces are compared for identification and attendance tracking.

User Interface and Integration: Designing a user-friendly interface for administrators and users to interact with the system seamlessly. Integration with existing attendance management systems or APIs enhances its usability and compatibility.

Accuracy and Performance Evaluation: Conducting rigorous testing and evaluation to assess the system's accuracy, reliability, and performance under various conditions, including varying lighting, angles, and facial expressions.

The synopsis emphasizes the system's potential to streamline attendance tracking processes in educational institutions, corporate settings, and other organizations. Its benefits include increased accuracy, reduced manual effort, and real-time monitoring capabilities.

The proposed workflow for Real-Time Face Attendance System encompasses: -

- 1) Data Preprocessing
- 2) Face Detection
- 3) Features Extraction
- 4) Generate Face Encodings
- 5) Compare Encodings
- 6) Update Attendance in Database

OBJECTIVE

The objectives of a real-time face attendance system using computer vision typically revolve around improving efficiency, accuracy, and convenience in attendance management processes. Here are the primary objectives:

1. **Automated Attendance Tracking:** To automate the attendance recording process by using facial recognition technology. This eliminates the need for manual attendance marking, reducing errors and time spent on traditional methods.
2. **Real-Time Monitoring:** To provide instantaneous attendance updates. As individuals are recognized in real-time, the system can immediately log their attendance status without delays.
3. **Accuracy and Reliability:** To ensure a high level of accuracy in identifying individuals. The system aims to minimize false positives or negatives, accurately associating individuals with their attendance records.
4. **Efficiency and Timesaving:** To streamline attendance management for both administrators and attendees. By eliminating manual processes, the system saves time and effort, allowing staff to focus on other essential tasks.
5. **Integration and Compatibility:** To integrate seamlessly with existing attendance management systems or databases. Compatibility ensures smooth implementation without disrupting established workflows.
6. **User-Friendly Interface:** To offer an intuitive and user-friendly interface for both administrators and users interacting with the system. This promotes ease of use and acceptance among users.
7. **Security and Access Control:** To enhance security by monitoring entry points and ensuring that only authorized individuals are granted access based on their recognition in the system.
8. **Scalability and Adaptability:** To design a system that can be easily scaled to accommodate varying numbers of users and locations. This adaptability allows the system to be implemented across different environments.
9. **Compliance and Ethical Considerations:** To adhere to ethical standards and legal requirements concerning data privacy, consent, and the responsible use of facial recognition technology.

DATABASE

Firebase, developed by Google, is a comprehensive platform offering various services to build and scale web and mobile applications. Firebase includes several tools and services, with Firebase Realtime Database being one of its key offerings:

Real-time Data Storage: Firebase Realtime Database is a cloud-hosted NoSQL database that stores data in JSON format. It offers real-time synchronization, allowing data to be synced across clients in milliseconds whenever changes occur.

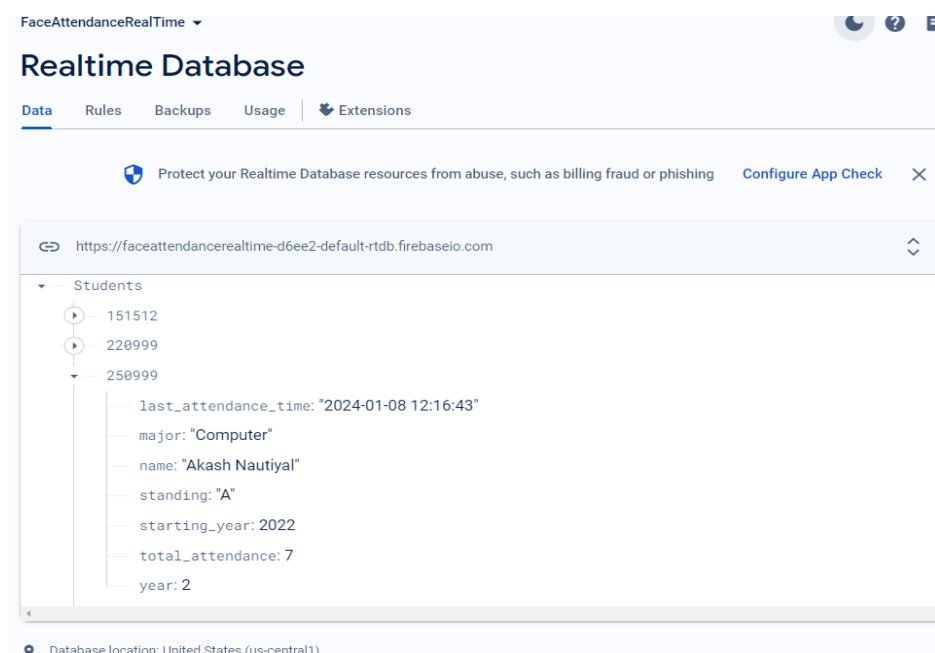
Synchronization Across Platforms: The database is accessible from various platforms, including web, iOS, Android, and server environments. Changes made by one client are instantly reflected on all connected devices, making it ideal for collaborative or multi-user applications.

Offline Support: Firebase provides offline capabilities, enabling applications to remain functional even without an internet connection. Once reconnected, the database syncs any pending changes automatically.

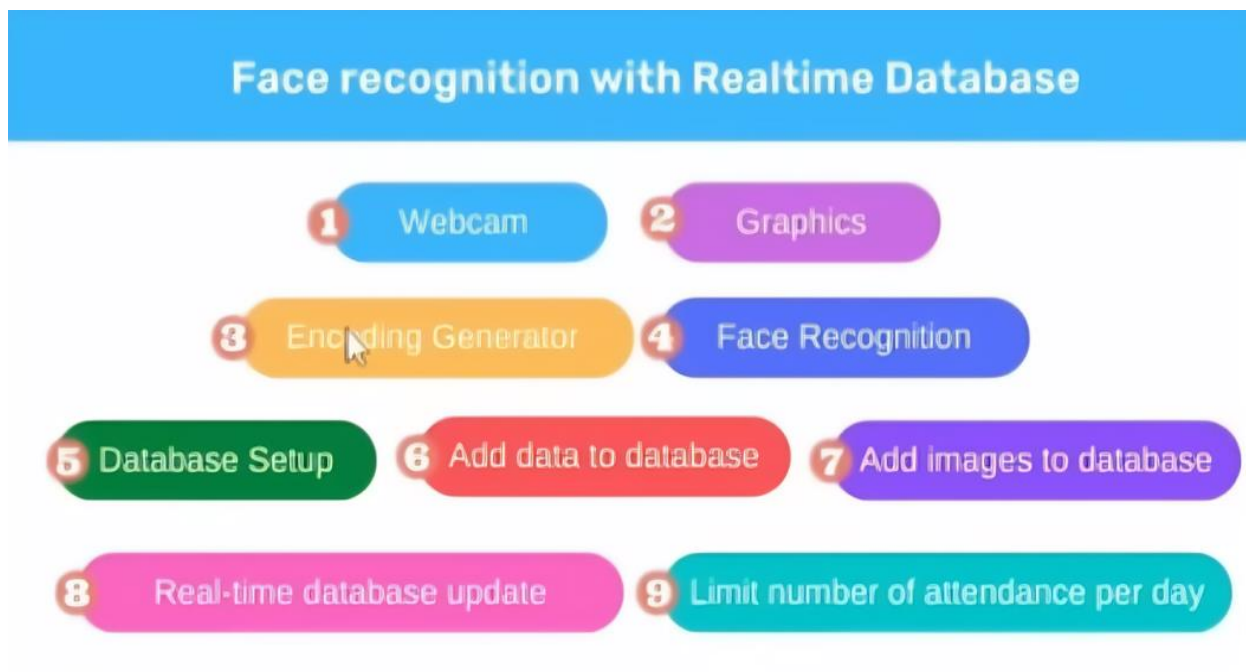
Security Rules: Firebase offers robust security rules to control access to the database. Developers can define rules based on authentication, user roles, and data structures to ensure data security and privacy.

Integration with Other Firebase Services: It seamlessly integrates with other Firebase services such as Authentication, Cloud Functions, Analytics, and Storage, allowing developers to create comprehensive and scalable applications.

Scalability: Firebase Realtime Database is designed to handle scalability requirements, automatically scaling to accommodate increasing data and user loads.



METHODOLOGY

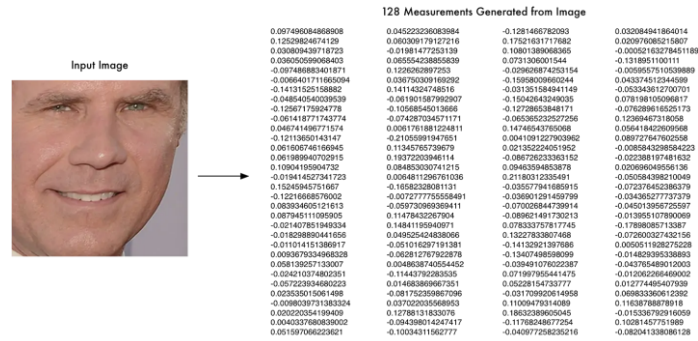


Data Preprocessing: Data preprocessing is a crucial step in data analysis and machine learning that involves transforming raw data into a more usable format. It aims to clean, normalize, and prepare the data to enhance its quality and make it suitable for further analysis or model training. It also involves storing of pre-processed data in database so that it can be used further.

Face Detection: Face detection is a computer vision technology used to identify and locate human faces within images or video frames. It's a foundational step in various applications, including facial recognition, security systems, photo editing software, and more.

Features Extraction: In face detection, feature extraction involves identifying and capturing discriminative information or patterns specifically related to facial characteristics within an image. Instead of extracting generic features from an image, the goal is to pinpoint facial attributes that enable the recognition or localization of faces.

Generate Face Encoding: Generating face encodings involves converting facial images into numerical representations, enabling easy comparison and recognition of faces. One popular method for face encoding is using deep learning models, particularly face recognition models, to extract facial embeddings.



Compare Encoding: In this process we generally compare both encoding of the database images and the captured images and the matched result is stored in list. The least value in the list is then identified by ID and is shown in the desired Graphics.

Update Attendance in Database: After showing desired image in the graphics, the attendance of the identified ID is then updated in the database in real time. If the attendance is already marked so it doesn't affect the database and the result "Already marked" is shown in the Output.

TOOLS/ SOFTWARE USED

Python: Make sure you have Python 3.7.6 installed on your system.

OpenCV: This code uses OpenCV for video capture, face detection, and image manipulation.

Cmake : cmake is used for simplifying and managing the build process of software projects across different platforms by generating build scripts, supporting various compilers, handling dependencies, and providing flexibility in choosing development environments.

Cvzone : cvzone is a Python library used for computer vision tasks. It simplifies complex vision-related processes like face detection, object tracking, and image processing by providing pre-built functions and tools, making computer vision development faster and more accessible.

Face recognition: Face recognition libraries in Python are used to identify and authenticate individuals based on facial features. These libraries provide pre-built functions and algorithms to detect faces, analyse facial characteristics, and compare them against known faces.

Camera: The code uses the default camera (Camera index 0). Ensure that you have a working camera connected to your system. If you're using an external camera, make sure it's properly configured and connected (camera index 1).

PyCharm: PyCharm is a popular Integrated Development Environment (IDE) specifically

designed for Python programming. Developed by JetBrains, it offers a range of features to facilitate Python development.

Firestore: Firestore, developed by Google, is a comprehensive platform offering various services to build and scale web and mobile applications. Firestore includes several tools and services, with Firestore Realtime

Operating System: This code should work on various operating systems (Windows, Linux, macOS) as long as Python, OpenCV are compatible with the chosen platform.

Running the Code:

Save the code in a Python file, e.g., main.py.

Make sure you have all the required dependencies installed.

Run the script. The webcam will open, and the facial recognition will be displayed, and the attendance will be updated in real time.

REFERENCES

1. [ageitgey/face_recognition: The world's simplest facial recognition api for Python and the command line](#)
2. <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffe121d78>
3. <https://firebase.google.com/>
4. <https://www.computervision.zone/>