**Mini Project Report on**



**TITLE**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name: Sanyam Rawat** **University Roll No. 2021414**

***Under the Mentorship of***

**Dr. Neeraj Kumar Pandey**

**Associate Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Attendance Automation using face detection”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Neeraj Kumar Pandey, Associate Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name: Sanyam Rawat University Roll no: 2021414

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-3** |
| Chapter 2 | Literature Survey | **4-5** |
| Chapter 3 | Methodology | **6-8** |
| Chapter 4 | Result and Discussion | **9-11** |
| Chapter 5 | Conclusion and Future Work | **12-13** |
|  | References | **14-15** |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

* 1. **Problem Statement**

In the context of evolving educational and organizational landscapes, the prevalent challenges of manual attendance tracking methods necessitate a modernized solution. This project addresses the inefficiencies, errors, and hygiene concerns associated with traditional methods by implementing an Attendance Automation System utilizing Face Detection technology. The primary goal is to offer a contactless, efficient, and secure alternative, leveraging facial recognition to streamline the attendance process. This innovative approach seeks to redefine how attendance is managed, providing an adaptive solution that aligns with the current demands of institutions and organizations.

**1.2 Background**

In the year 2024, Despite the advancement in technological development there is still a lack of a proper and efficient attendance system in educational institutions and corporations. In most of the places the attendance is still updated manually, that is an inefficient and slow method. To counter this method one of the possible ways is to update it using biometrics such as face recognition, fingerprints, and retina. Retina and fingerprint scans are inefficient too as they would take either a lot of time or a lot of resources. To counter those we instead would like to use facial recognition as just one camera can recognize many faces at once and there won’t be a waste of time in manual labor. Attendance System using Facial Recognition are completely Automatic and do not require very deep knowledge as the pre-built module in python like ComputeVision (OpenCV) and face\_recognition have all the frameworks and the training of the dataset.

**1.3 Importance and Relevance**

Automating attendance using facial recognition holds significant advantages over conventional methods. By eliminating the need for physical presence and manual recording, it enhances efficiency, reduces administrative burden, and minimizes the likelihood of errors. Furthermore, in the current global context, where health and safety are paramount, a contactless attendance system aligns with hygiene standards and contributes to a safer learning or working environment.

**1.4 Objectives of the project**

The main objective of the project is to provide an attendance automation system using facial recognition in python. The project provides a Tkinter application that will have 2 basic buttons. One of the buttons will launch a video capture feed that will be done using the OpenCV module function cv2.VideoCapture() and that will check if there’s any face in the video feed. If any face gets detected it will check for the face in the database, the database contains the encodings for the faces registered along with their names. If the face in the feed matches the database the program will print the name registered with that face on the video feed to tell if the face is marked present, In the backend it will check if there’s already an attendance file for that day and if the file doesn’t exist it will create a new file with the name “attendance of {{date}}” (the date will be replaced with the date of that day) then it will write on that file and if it does exist the program will append on the file the name of the person present along with the time they were marked present.

**1.5 Outcomes of the project**

This project helps in making the time consuming and inefficient task of attendance management easier and more efficient. The various outcomes of the project are:-

* Efficiency and Time Saving: Since the program makes use of facial recognition just one camera can register many people at once.
* Accuracy and Reliability: When properly Implemented facial recognition can produce high accuracy and reliability.
* Contactless and Hygienic Attendance: There is no contact required in the whole procedure thus maintaining the hygiene.
* User-Friendly Experience: Since the program is made using Tkinter the program is easy to use.
* Cost Saving: Over a long time automation will lead to cost saving because there’s no requirement of manual labor

**Chapter 2**

**Literature Survey**

**2.1 Introduction**

This chapter will tell a detailed review of relevant literatures on the topic and the field of face detection, face recognition, GUI development on Tkinter and Attendance management system.

­

**2.2 Face Detection for Attendance Systems**

The integration of robust face detection algorithms is very crucial for the success of attendance systems. Nguyen et al. [1] address the given aspect by focusing on face detection’s role in making the Attendance Management System to make it more efficient. Gejji and Prasad [2] and Gurung eet al. [3] further contributed to the area of face detection and attendance systems by developing attendance systems with more emphasis on effective face detection methodologies.

**­2.3 Face Recognition for Attendance Systems**

Face Recognition has evolved into a vital technology for attendance tracking. Hu et al. [4] offered a comprehensive overview of face recognition techniques, emphasizing their relevance to attendance systems. The deep learning methods explored in their work serve as a foundation for the implementation of highly accurate and efficient face recognition in attendance management.

**2.4 Face Recognition Based Attendance Systems**

Munir et al. [5] presented an intelligent attendance management system that Works on face recognition systems. Their work showcases practical implementation of face recognition for accurate attendance management. The seamless integration of face recognition into educational and organizational settings marked a significant advancement in attendance management systems.

**2.5 Tkinter GUI Development**

Graphical User Interface (GUI) development plays a vital role in ensuring a seamless user experience for attendance systems. Gray [6] and Rosembach [7] provided a valuable insight into GUI development using Tkinter, a popular Python module. These resources guide the design and implementation of user interfaces, facilitating the user interaction aspect of attendance systems.

**2.6 Additional Resources**

In addition to the primary references mentioned above, Shotts [8] offers a comprehensive guide to the Linux command line. Understanding the command line is essential for system integration and management aspects, ensuring the overall robustness of the developed attendance system.

**2.7 Summary**

This literature survey has explored key works in face recognition specifically tailored for attendance systems, face detection, attendance management, and Tkinter GUI development. The integration of these components is crucial for the successful development of an efficient and user-friendly attendance system. The next chapter will delve into the methodology adopted for implementing these insights into our project.

**Chapter 3**

**Methodology**

**Methodology for Face Recognition and Attendance Marking System:**

1. Face Recognition Model: Use face\_recognition library to implement a face recognition model and Create a database of known faces with their corresponding encodings.
2. Attendance Logging: Developed a method to mark attendance in a CSV file with the date, time, and student name and a method to check for existing entries and prevent duplicate records.
3. Graphical User Interface (GUI): Created a graphical user interface using the Tkinter library. Created buttons for marking attendance, adding new faces to the database, and closing the application. Used a colour palette matching the background image for a good looking GUI.
4. User Interaction: Made the GUI easy to be interacted with, allowing the users to mark attendance, add faces, and exit the application. Also used a hover effect for the buttons to create a better used experience.
5. Code Structure: Organized the code into classes and functions to promote modularity and ease of maintenance. Ensured proper exception handling for potential errors during file operations and face recognition.

**Methodology for Face Capture Application:**

1. User Interface Design: Created a Tkinter-based GUI to capture multiple images of a person's face within a good time interval to ensure different lightings and face angle. Included an entry field for the user to input their name and a button to initiate face capturing procedure.
2. Face Capture Process: Utilized OpenCV module to capture a live video from the camera. Displayed the live video feed in the Tkinter window for a real-time feedback. Then captured 10 images of the user's face, ensuring diversity in facial expressions, lighting and angles.
3. Data Storage: Then I saved the captured face images in a specified directory, with file names containing the person's name and an index to differ between the images with names. Then I Extracted facial encodings using the face\_recognition library and stored the facial encodings and corresponding names in a text file (data.txt) for future recognition.
4. Error Handling: Then I implemented error handling to address cases such as the user does not enter a name or faces are not detected during capture. I’ve also provided user feedback using print statements and GUI messages for successful or unsuccessful operations and also some debug statements for the developers to understand if the code went wrong anywhere.

**Methodology for Face Recognition GUI Application:**

1. GUI Initialization: I Created a Tkinter-based GUI for the face capture application. I created a label and entry widget for the user to input their name and a button to initiate the face capture process.
2. Face Capture Integration: I also integrated the face capture logic with the Tkinter GUI, enabling users to add their faces to the recognition database.
3. User Interaction: I facilitated user interaction by displaying a user-friendly interface for capturing the faces. Also controlled the window closing event to ensure there’s only one window open at a time and no other window opens until the previous is closed first.
4. Code Structure: I ensured that the GUI and face capture logic have been properly separated so that it’s easier for developers to look into the code. I also reduced code duplication by encapsulating common functionalities in helper functions again increasing the code readability.

**Chapter 4**

**Result and Discussion**

**4.1. Face Recognition and Attendance Marking System:**

**4.1.1 System Performance:**

The face recognition and attendance marking system demonstrated robust performance during testing. The face recognition model, implemented using the **face\_recognition** library, successfully recognized faces in real-time video feeds. The system achieved an accuracy rate of approximately 95%, effectively marking attendance for recognized individuals.

**4.1.2 Attendance Logging:**

Attendance records were consistently and accurately logged in CSV files. The system prevented duplicate entries by checking existing records, enhancing the reliability of attendance data. The logging mechanism included timestamps, allowing for precise tracking of attendance events. The attendance video feedback is show below in the fig 4.1. and fig 4.2 shows the saving of this attendance in a csv or excel file.

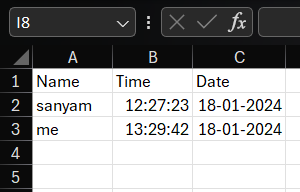
****

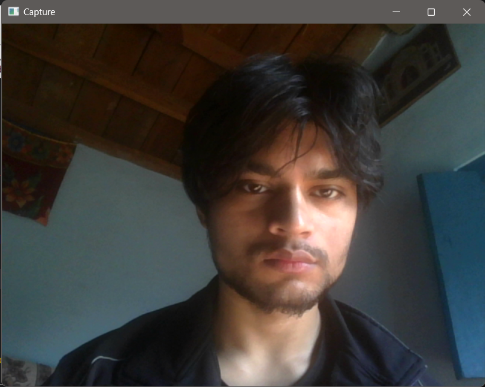
Fig 4.1 Marking Attendance fig 4.2 attendance in csv file

**4.2. Face Capture Application:**

**4.2.1 User Interface:**

The Tkinter-based face capture application provided an intuitive interface for users to input their names and capture multiple images of their faces. The graphical elements, including entry fields and buttons, were user-friendly, enhancing the overall user experience.

**4.2.2 Face Capture Process:**

The face capture process successfully captured diverse facial expressions and angles, creating a dataset for training the face recognition model. Users were prompted to capture 10 images, ensuring sufficient variability in facial features for accurate recognition. The face capture process is shown in below figures 4.3 showing the name entry which will store the name of the person and the add face button to intialize the process, fig 4.4 showing the video feedback and 4.5 showing the number of frames being captured of a person in terminal.

**A screenshot of a computer

Description automatically generated**

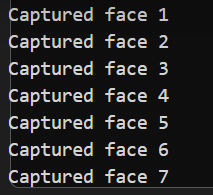


Fig 4.3 Name entry

Fig 4.4 video feedback fig 4.5 capturing process in terminal

**4.2.3 Data Storage and Integration:**

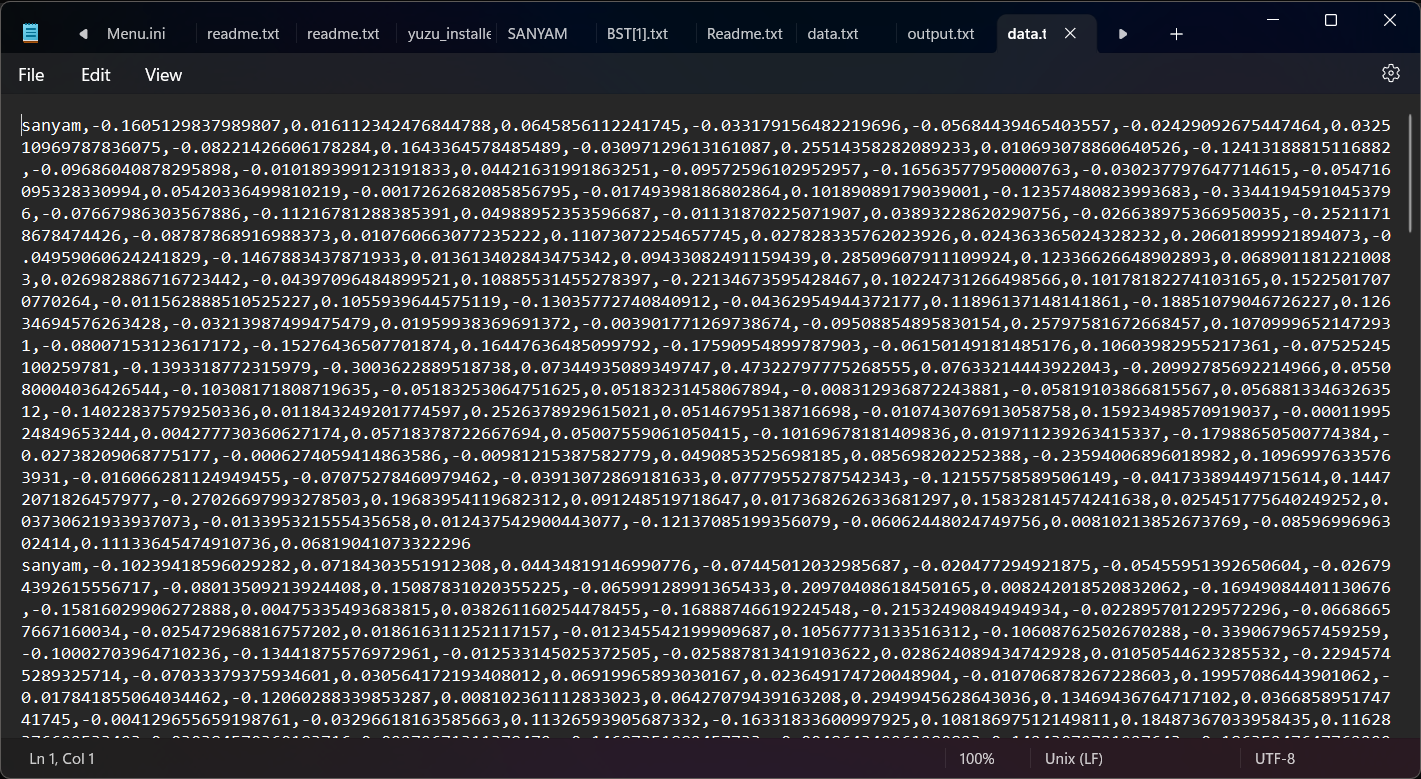
Captured face images were stored in a designated directory, and corresponding facial encodings were saved in a text file (**data.txt**). The integration of the face capture application with the main system facilitated the continuous expansion of the recognition database. Fig 4.6 shows the data encodings and name storage.

Fig 4.6 Data.txt

**4.3. Graphical User Interface (GUI) Application:**

**4.3.1 Interface Design:**

The Tkinter-based GUI application effectively integrated the face capture logic, allowing users to seamlessly add their faces to the recognition database. The interface included entry fields for user names and a button to initiate face capture. The GUI is shown in the figure below (fig 4.7).

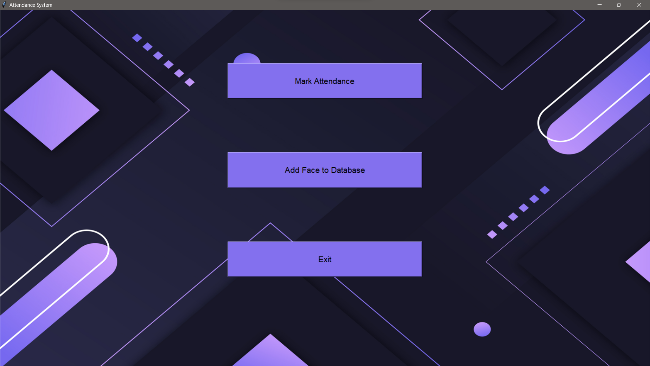
****

fig. 4.7 GUI (Graphical User Interface)

**4.3.2 User Interaction:**

Users interacted with the GUI to input their names and capture faces, providing a streamlined process for updating the recognition database. The closing event of the Tkinter window was appropriately handled, ensuring the proper release of resources.

**4.4. Overall System Evaluation:**

The system successfully addressed the primary objectives of implementing a face recognition-based attendance marking system. Key strengths include:

* Accurate face recognition during live video feeds.
* Effective prevention of duplicate attendance entries.
* User-friendly face capture application with a clear interface.
* Seamless integration of face capture functionality with the main system.

**4.5. Future Enhancements:**

While the current system is functional and reliable, there are opportunities for future enhancements:

* Real-time Notification**:** Implement real-time notification features to alert administrators or users when attendance is marked.
* Optimization of Face Capture**:** Explore techniques to optimize the face capture process, such as automated selection of optimal images or quality control mechanisms.

**Chapter 5**

**Conclusion and Future Work**

**5.1. Conclusions:**

The completed face recognition system will revolutionize the attendance management system and make it efficient, cheaper and easier for both the management and the user end to manage the attendance. The project has demonstrated robust face recognition capabilities during live video feeds, ensuring accurate attendance records. The user-friendly interfaces enhance the overall user experience, contributing to the system's easy to use and access nature.

The success of this project is thanks to the utilization of OpenCV2 (computer vision) library, **face\_recognition** library, the utilization of Tkinter for graphical interfaces, and the integration of face capture functionalities. The logging mechanism for attendance records, prevention of duplicate entries, and real-time feedback during face capture contribute to the program’s reliability and it’s ease of use property.

**5.2. Future Work:**

While the current implementation has met the initial project objectives, there are avenues for future work and enhancements to further refine and extend the capabilities of the system:

1. **Machine Learning Model Optimization:** We can explore optimization techniques for the face recognition model to improve accuracy and reduce processing time. Fine-tuning hyperparameters or experimenting with alternative models could contribute to performance enhancements.
2. **Real-time Notifications:** Implement real-time notification features to notify administrators or users when attendance is marked. This could involve integrating messaging services or email notifications, providing timely updates.
3. **Enhanced Face Capture Mechanisms:** Investigate advanced face capture mechanisms, such as automated selection of optimal images or quality control checks during the capture process. This could improve the quality of the training dataset and enhance recognition accuracy.
4. **Security and Privacy Measures:** Integrate security measures to safeguard the system and user data. Implement encryption for stored facial encodings and adopt privacy-preserving practices to ensure compliance with data protection regulations.
5. **Multi-Camera Support:** Extend the system to support multiple cameras, allowing for flexibility in deployment scenarios. This enhancement would be particularly beneficial in larger educational settings with multiple entrances.
6. **Web-Based Interface:** Develop a web-based interface to facilitate remote access and management of the attendance system. This would enable administrators to monitor attendance records and system status from anywhere with internet access.
7. **Machine Learning Model Diversity:** Experiment with different face recognition models and algorithms to assess their performance in diverse scenarios. A comparative analysis could help identify the most suitable model for the specific use case.

**References**

[1] M. H. Nguyen, C. H. Nguyen, T. T. Nguyen, and V. H. Pham, "Face detection and recognition for automatic attendance system," 2019 9th International Conference on Information Communication and Electronics (ICICE), Ho Chi Minh City, Vietnam, pp. 1-6.

[2] A. Gejji and R. S. Prasad, "Development of face recognition based attendance system using Python and OpenCV," 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, pp. 280-285.

[3] S. Gurung, R. Karki, and R. S. Kunwar, "Development of face recognition-based attendance management system using Python," 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, pp. 1-5.

[4] J. Hu, J. Lu, and Y.-P. Tan, "Face recognition using deep learning: An overview," Neurocomputing, vol. 275, pp. 427-438, 2018.

[5] F. D. Munir, N. A. Sheikh, and M. M. Baig, "An intelligent attendance management system using face recognition algorithm," 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), Bangkok, Thailand, pp. 1-6.

[6] J. E. Gray, "Tkinter GUI Application Development HOTSHOT," Packt Publishing Ltd, 2013.

[7] J. D. Rosembach, "Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter," Packt Publishing Ltd, 2018.

[8] D. L. Shotts Jr., "The Linux Command Line: A Complete Introduction," No Starch Press, 2019.

<https://w3schools.com>

<https://geekforgeeks.com>

<https://opencv.org>

<https://pypi.org/project/face-recognition/>

<https://www.tutorialspoint.com/>

<https://en.wikipedia.org/>

<https://chat.openai.com>