WIPRO NGA Program – Data Engineering & AI/ML Batch

Capstone Project Presentation – 17th July 2024

Project Title - Smart Attendance System

Presented by -

1. Shubham Sangale
2. Utkarsh Shahane
3. Shivani Katoju
4. Sree Teja Boorla
5. Jahnavi Venumbaka

Smart Attendance System

1. Abstract:

The Smart Attendance System leverages the K-Nearest Neighbors (KNN) algorithm to automate and enhance the attendance tracking process in educational institutions, corporate environments, healthcare facilities, and public sector organizations. Traditional attendance methods often rely on manual processes, which can be time-consuming and prone to errors. This system utilizes facial recognition technology to identify people and record their attendance in real time. By implementing the KNN algorithm, the system classifies people based on their facial features, ensuring accurate identification even in varying conditions.

The solution integrates a user-friendly interface for institutions to monitor attendance records, analyze the individual participation, and generate reports. This approach not only streamlines attendance management but also enhances security and accountability in academic environments, contributing to more efficient administration. Attendance data will be recorded with timestamps and stored in a structured format, such as CSV files. This comprehensive solution aims to enhance accuracy, reliability, and efficiency in attendance management, reducing manual efforts and minimizing errors in record-keeping.

2. Introduction:

In today's fast-paced world, the need for efficient and accurate attendance tracking systems is much needed. Traditional methods of attendance recording, such as manual roll calls and paper-based systems, are often time-consuming, error-prone, and susceptible to manipulation. To address these challenges, modern technology offers innovative solutions, one such solution is a smart attendance system utilizing the K-Nearest Neighbors (KNN) algorithm for facial recognition.

This system leverages the power of machine learning to automate the identification and tracking of individuals based on their facial features. The KNN algorithm, known for its simplicity and effectiveness, serves as the backbone of this smart attendance system, ensuring high accuracy and reliability. The implementation of a KNN-based attendance system offers numerous benefits across various domains. This system not only saves time and resources but also provides valuable insights into attendance patterns and behaviors.

3. Objective:

The objective of this project is to harness the power of the K-Nearest Neighbors (KNN) algorithm to develop a face recognition-based attendance system. This system is designed to automate and streamline the process of attendance tracking in various settings such as educational institutions, offices, or events, by leveraging facial biometric data. The KNN algorithm, renowned for its simplicity and effectiveness in classification tasks, is well-suited for identifying individuals based on their facial features.

4. Scope:

The scope of this project encompasses the development of an Smart Attendance System leveraging the K-Nearest Neighbors (KNN) algorithm for facial recognition. This system aims to streamline the attendance tracking process by capturing real-time video frames, detecting and pre-processing faces, and accurately identifying individuals using the KNN algorithm. The project includes the implementation of a user-friendly interface that displays the real-time video feed and recognized faces.

5. Literature Theory:

5.1. Importance of Smart Attendance System:

Attendance systems have become increasingly important in various sectors, including educational institutions, corporate environments, and security systems. Manual attendance tracking methods are time-consuming, prone to errors, and can be easily manipulated. The implementation of automated systems ensures accurate, reliable, and efficient attendance tracking, reducing administrative burdens and increasing productivity.

5.2. Role of Machine Learning in Attendance Systems:

Machine learning has revolutionized the development of automated systems by enabling computers to learn from data and make decisions with minimal human intervention. In the context of attendance systems, machine learning algorithms are used to recognize individuals based on biometric features such as facial recognition. In this algorithm it includes K-Nearest Neighbors (KNN) model. This algorithm analyze patterns in data to accurately identify individuals and record their attendance.

5.3. Application of K-Nearest Neighbors (KNN) in Attendance Systems:

K-Nearest Neighbors (KNN) is a simple, yet powerful machine learning algorithm used for classification tasks. It is particularly suitable for scenarios where the relationship between the input features and the output labels is not linear, making it an effective choice for facial recognition-based attendance systems.

Key Steps in Implementing KNN for Attendance Systems:

- **Image Preprocessing:**

  Image preprocessing is a crucial step in preparing data for machine learning models, especially in facial recognition systems. It involves various techniques to enhance the quality of the images, standardize them, and make them suitable for the recognition algorithm.

- **Face Detection and Alignment:**

  Detected faces are cropped and aligned to ensure consistency in the training and recognition phases.

- **Feature Extraction:**

  Relevant features are extracted from the facial images to represent the data effectively. It involves identifying and extracting meaningful information from the preprocessed images that can be used to distinguish different faces.

- **Model Training:**

  The KNN algorithm is trained on the extracted features and their corresponding labels (e.g., names of individuals). Model training in this attendance system involves loading the preprocessed and labeled data, splitting it into training and testing sets, training a KNN classifier, evaluating its performance, and saving the trained model for future use.

- **Recognition and Attendance Marking:**

  In the recognition phase, new images are captured, and faces are detected and processed similarly. The trained KNN model predicts the identity of the detected faces based on their features. Attendance is recorded by saving the identified individual's name and the timestamp in a csv file.

- **Evaluation:**

  The KNN algorithm was trained and evaluated on the collected dataset, showing robust performance in recognizing faces. Saving the trained model enabled quick and efficient face recognition during real-time attendance marking.

5.4. Advantages and Challenges of Using KNN for Attendance Systems:

Advantages:

- **Simplicity:** KNN is easy to implement and understand, making it accessible for various applications.
- **Effectiveness:** It provides good accuracy for classification tasks with a sufficient amount of training data.
- **Flexibility:** The algorithm can be adapted to different datasets and environments without extensive modifications.

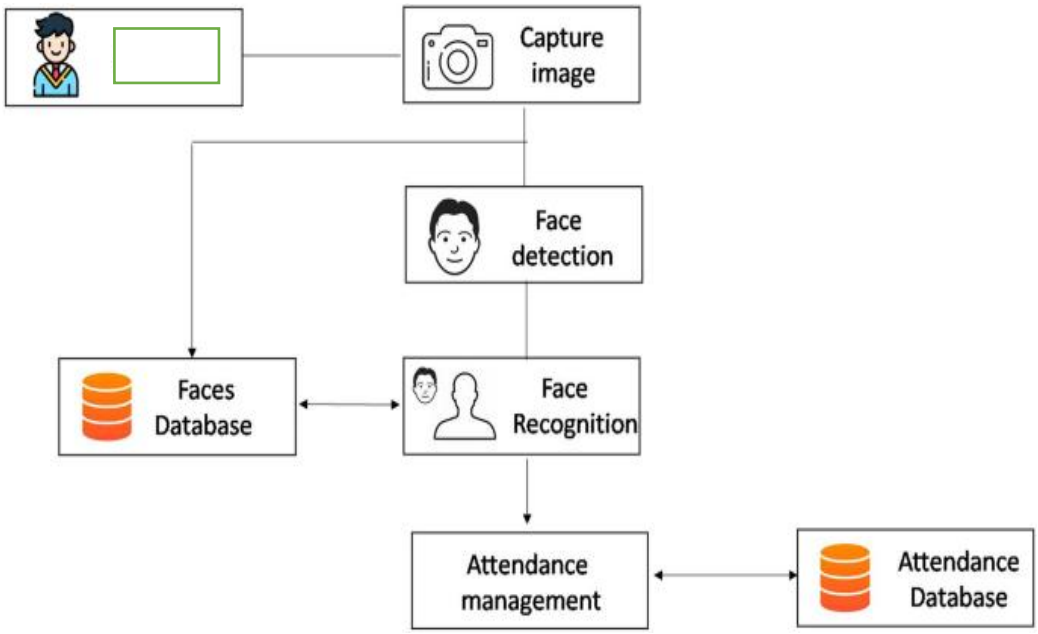
Challenges:

- **Scalability:** KNN can be computationally intensive for large datasets, as it requires calculating distances to all training samples.
- **Performance:** The accuracy of the KNN model can be affected by the choice of K and the quality of the training data. Proper preprocessing and feature extraction are crucial to achieving optimal results.

6. Methodology:

- **Data Collection:** Capture and label images using a webcam.
- **Data Organization:** Store images in labeled folders.
- **Pre-processing and Training:** Prepare data, train KNN model.
- **Model Evaluation:** Test and validate the model.
- **Real-Time Detection:** Implement real-time attendance recognition.
- **Marked Attendance:** Updated the Attendance Excel Sheet.

7. Architecture:



8. Dataset Collection and Preparation:

A dataset is collected and processed using a webcam, leveraging computer vision techniques and machine learning algorithms to create an automated attendance system. The system utilizes the OpenCV library for real-time face detection and image preprocessing, and the NumPy library for efficient numerical operations. The key steps in data preparation are as follows:

8.1. Data Collection:

- A webcam is used to capture real-time video frames.
- OpenCV's Cascade Classifier with the haarcascade_frontalface_default.xml file is employed to detect faces within the video frames.
- When a face is detected, it is cropped from the frame and resized to a fixed size of 50x50 pixels.
- The resized face images are saved in a directory structure that associates each image with the corresponding person's name.

8.2. Data Organization:

- The processed face images are stored in a structured directory, with each subdirectory named after the person whose face images are contained within it.
- The images are labeled with the person's name to create a labeled dataset that can be used for supervised learning.

8.3. Dataset Structuring:

- The labels (names) are stored in a separate the images folder.

8.4. Data Storage:

- The prepared dataset, including the face images and labels, is stored to database using the pickle library to enable easy loading and manipulation for model training and testing.
- The face images are saved as numpy arrays in a file named faces_data.pkl, and the corresponding labels are saved in a file named names.pkl.

8.5. Training and Testing Split:

- The dataset is split into training and testing sets using the train_test_split function from the sklearn.model selection module.
- Typically, 80% of the data is used for training the KNN classifier, and 20% is reserved for testing and evaluating the model's performance.

9. Environment Setup:

Operating System - Windows 11

IDE - PyCharm 2024.1.2

Libraries -

- **OpenCV:** (Open Source Computer Vision Library) Used for real-time computer vision, including capturing images from the webcam and detecting faces.
- **NumPy:** For numerical operations and handling arrays.
- **pickle:** For saving and loading data (face images and labels) and the trained model.
- **OS:** For interacting with the operating system, such as creating directories and checking for file existence.
- **csv:** For handling attendance records in CSV format.
- **time:** For timestamping attendance records.
- **datetime:** For handling date and time information.
- **win32com.client:** For text-to-speech functionality to notify the user when attendance is taken.
- **matplotlib:** For visualizing attendance data.
- **pandas:** For data manipulation and analysis.

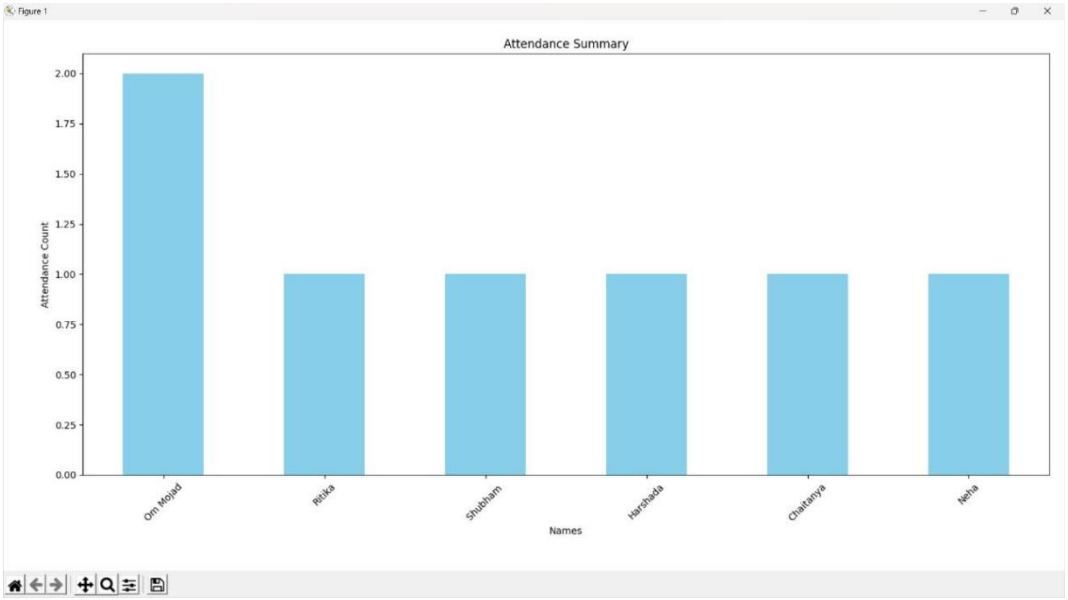
10. Results:

10.1. User Interface Window:

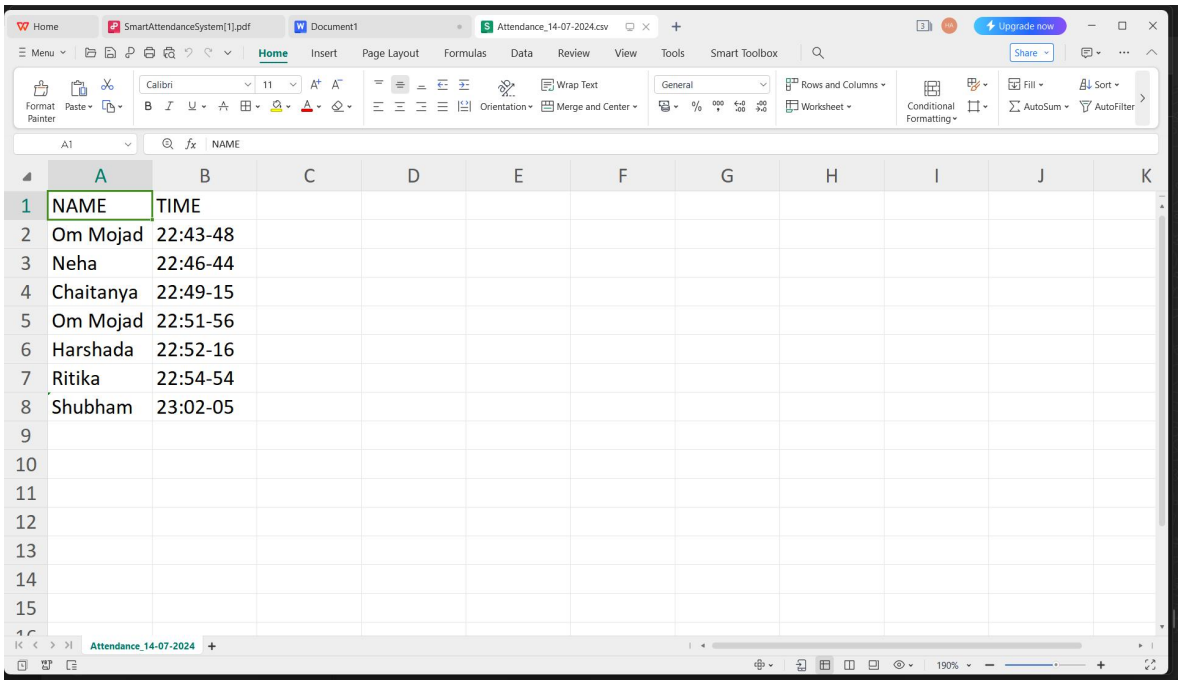
- Press '**o**'
- Press '**v**'
- Press '**q**'

10.2. Data Visualization:

A bar plot is created using matplotlib to display the attendance summary. The plot shows names on the x-axis and the count of the attendance on the y-axis. The plot is customized with labels, titles, and appropriate layout settings. This visualization helps in understanding the attendance patterns and provides a summary of how many times each person marked present.



10.3. Updated Attendance Sheet:



	A	B
1	NAME	TIME
2	Om Mojad	22:43-48
3	Neha	22:46-44
4	Chaitanya	22:49-15
5	Om Mojad	22:51-56
6	Harshada	22:52-16
7	Ritika	22:54-54
8	Shubham	23:02-05

11. Conclusion:

The face recognition-based attendance system using the KNN algorithm proved to be an effective solution for automating attendance tracking. The project involved several key steps, including data collection, pre-processing, model training, and attendance marking, each contributing to the system's overall success.

12. References:

- Smitha, Pavithra S Hegde and Afshin, Face Recognition based Attendance Management System, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 (2020).
- S. Sai Kumar, S. Adithya Varun, Dr. P. K. Sahoo and K. Eswaran, Class Room Attendance System using KNN, International Journal of Computer Science and Technology (2019).
- Mitesh Chauhan, Mandar Dhakate, Jaiveek Baria and Prof. Nileema Pathak, Attendance System using Face Recognition, International Research Journal of Engineering and Technology (IRJET) (2022).