* **A**

**Project Based Learning – Project Report**

**on**

**“SMART ATTENDANCE USING FACE RECOGNITION”**

**Submitted by**

**Mr. ANUP JAGTAP.**

**Mr. ANIKET JAGTAP.**

**Under the guidance of**

**Prof. SWAPNIL ROKADE**

****

**Department of First Year Engineering**

**Dhole Patil College of Engineering,**

**Wagholi, Pune- 412207**

**2022-2023**

**Dhole Patil College of Engineering,**

**Wagholi, Pune- 412207**

****



This is to certify that Mr. **ANUP JAGTAP** & Mr. **ANIKET JAGTAP** has successfully completed the Project Based Learning – Project Report entitled “**SMART ATTENDANCE USING FACE RECOGNITION** ” under my supervision, in the partial fulfillment of First Year of Engineering in Savitribai Phule Pune university.

**Date: 08 / 07 /2023**

**Place: Dhole patil college of engineering, Wagholi, Pune- 412207**

**Prof Swapnil Rokade Mr. Omkar Gangatire Prof. Sandip Kale**

**Guide name Coordinator HOD**

**ACKNOWLEDGEMENT**

I would like to express my deepest gratitude and appreciation to all the individuals and organizations who have supported and contributed to the completion of this project. Without their invaluable assistance, this project would not have been possible.

First, I would like o thank my project guardian, Prof. Swapnil Rokade, for their guidance, expertise, and constant support throughout the project. Their insights and suggestions have been instrumental in shaping the project and ensuring its successful completion.

I would like to extend my sincere appreciation and gratitude to Prof. Sandip Kale for their invaluable support and guidance during the course of this project.

I am grateful to express my appreciation to my friends and understanding throughout the project.

**Anup Jagtap**

**&**

**Aniket Jagtap.**

**ABSTRACT**

Face recognition is among the most productive image processing applications and has a pivotal role in the technical field. Recognition of the human face is an active issue for authentication purposes specifically in the context of attendance of students. Attendance system using face recognition is a procedure of recognizing students by using face biostatistics based on the high definition monitoring and other computer technologies. The development of this system is aimed to accomplish digitization of the traditional system of taking attendance by calling names and maintaining pen-paper records. Present strategies for taking attendance are tedious and time-consuming. Attendance records can be easily manipulated by manual recording. The traditional process of making attendance and present biometric systems are vulnerable to proxies. After face recognition attendance reports will be generated and stored in excel format. The system is tested under various conditions like illumination, head movements, the variation of distance between the student and cameras. After vigorous testing overall complexity and accuracy are calculated. The Proposed system proved to be an efficient and robust device for taking attendance in a classroom without any time consumption and manual work. The system developed is cost-efficient and need less installation. Keywords: face recognition; deep learning, attendance system.

**CONTENT**

* **CHAPTER 1 INTRODUCTION**
* **CHAPTER 2 AIM AND OBJECTIVES**
* **CHAPTER 3 LITERATURE REVIEW**

**PROCEDURE**

**METHODOLOGY**

* SOURCE CODE AND OUTPUT
* **CHAPTER 4 ADVANTAGES**

**DISADVANTAGES**

* **CHAPTER 5 APPLICATIONS**
* **CHAPTER 6 CONCLUSION**

# CHAPTER 1

# INTRODUCTION

A smart attendance system refers to a technological solution that automates and streamlines the process of recording and managing attendance in various settings, such as schools, colleges, workplaces, and events. It leverages advanced technologies like biometrics, RFID (Radio Frequency Identification), and computer vision to accurately track and monitor attendance in real-time.

Traditionally, attendance systems relied on manual methods like paper registers, sign-in sheets, or punch cards, which were time-consuming, prone to errors, and lacked efficiency. However, with the advent of smart attendance systems, these challenges have been mitigated, leading to improved accuracy, convenience, and effectiveness in attendance management.

Smart attendance systems offer several benefits over traditional methods. They can automatically identify individuals using biometric features such as fingerprints, facial recognition, or iris scans, ensuring a high level of accuracy and preventing proxy attendance. RFID-based systems utilize tags or smart cards that can be easily scanned to record attendance. Computer vision technology can analyze visual data from cameras to recognize and record individuals present in a particular area.

The implementation of smart attendance systems brings various advantages. It reduces administrative workload by automating attendance tracking and generating comprehensive reports. It provides real-time data on attendance, allowing for prompt action in case of discrepancies or emergencies. Moreover, it facilitates seamless integration with other systems like payroll, student information, or human resource management systems, streamlining overall operations.

**CHAPTER 2**

**AIM AND OBJECTIVES**

The aim of a smart attendance system is to provide an efficient, accurate, and automated method for recording and managing attendance in various settings. The system aims to replace traditional manual methods with advanced technologies to streamline the attendance management process.

The objectives of a smart attendance system typically include:

Automation: The system aims to automate the attendance tracking process, eliminating the need for manual entry and reducing administrative workload. By automating data collection and storage, the system saves time and resources.

* Accuracy: One of the primary objectives is to improve the accuracy of attendance records. Smart attendance systems utilize technologies like biometrics, RFID, or computer vision, which significantly reduce the chances of errors or fraudulent attendance.
* Real-time tracking: The system aims to provide real-time data on attendance. It allows administrators to access attendance records instantly and monitor attendance trends, enabling them to take timely action if needed.
* Integration: Smart attendance systems often aim to integrate with other systems or databases, such as student information systems or human resource management systems. Integration facilitates seamless data flow and enables better coordination between attendance records and other relevant processes like payroll or performance evaluation.
* Security and privacy: Ensuring the security and privacy of attendance data is another key objective. Smart attendance systems implement robust security measures to protect sensitive information and comply with privacy regulations. This includes secure storage, encryption, and access control mechanisms.
* Reporting and analysis: The system aims to generate comprehensive reports and analytics related to attendance. It allows administrators to analyze attendance patterns, identify trends, and make data-driven decisions to improve attendance management strategies.
* Enhanced efficiency: By automating attendance processes, the system aims to enhance overall operational efficiency. It reduces administrative tasks, eliminates paperwork, and enables reallocation of resources to more productive activities.
* Cost-effectiveness: Smart attendance systems aim to provide a cost-effective solution in the long run. While there may be an initial investment in implementing the system, the automation and efficiency gained can lead to significant cost savings over time.

By achieving these objectives, smart attendance systems aim to revolutionize attendance management, offering a more accurate, efficient, and streamlined approach to tracking and managing attendance in various settings.

**CHAPTER 3**

**LITERATURE REVIEW**

The primary purpose of this paper review is to find the solutions provided by others author and consider the imperfection of the system proposed by them, give the best solutions. In [18] Kawaguchi introduced a lecture attendance system with a new method called continuous monitoring, and the student’s attendance marked automatically by the camera which captures the photo of a student in the class. The architecture of the system is simple since two cameras equipped with the wall of the class. The first one is a capturing camera used to capture the image student in the class and the second camera is sensor camera is used to getting the seat of a student inside the class and the camera capturing will snap the image of the student. The system compares the picture taking from a camera capturing images and faces in the database done much time to perfect the attendance. Other paper proposed by [2] introduced a real-time computer vision algorithm in automatic attendance management system. The system installed the camera with non-intrusive, which can snap images in the classroom and compared the extracted face from the image of the camera capturing with faces inside the system.

This system also used machine learning algorithm which are usually used in computer vision. Also, HAAR CLASSIFIERS used to train the images from the camera capturing. The face snap by the camera capturing will convert to grayscale and do subtraction on the images; then the image is transferred to store on the server and processing later. In 2012 N. Kar [19] introduced an automated attendance management system using face recognition technique which used the Principal Component Analysis To implementation the system, use two libraries such OpenCV is a computer vision library and FLTK(Light Tool Kit. Both of this libraries helped the development such as OpenCV support algorithm[20] and FLTK [21] used to design the interface. In the system, there are Request Matching and Adding New fact to Database. In Request Matching, the first step is open the camera and snap the photo after the extraction the frontal face. The next step is recognizing the face with the training data and project the extracted face onto the Principal Component Analysis. The final step displays the nearest face with the acquired images. Apart from that, adding a new face into the database is snap the photo after that extract the frontal face images and then perform the Haar cascade Method to find the perform the Principal Component Analysis Algorithm. The final step is storing the information inside the face XML file. The system is focused on the algorithm to improve the face detection from acquired images or videos. In [3] the author also proposed a system which implements automatic attendance using face recognition. The system which can extract the object in the face such nose, mouth by using MATLAB with Principal Component Analysis (PCA). The system [7] designed to resolve the issues of attendance marking system such as timeconsuming. As the result of the experiment show that this paper, the system can recognize in case the dark background or difference view of the face in the classroom. Jyotshana Kanti [4] proposed a smart attendance marking system which combines two differencing algorithms such Principal Component Analysis and Artificial Neural Network. The purpose of the author is to solve the traditional attendance marking system and to resolve the time-consuming. In the system implement with Principal Component Analysis, it does an extraction and identify the similarities of the face database and acquire images. Artificial Neural Network is used to solve the problem of the input data or learn from the input data, and the expect value. In the system implemented by the author using back propagation algorithm and combines with mathematical function to perform in that system. As a result, written by the author research, it shows that the system can use to recognize in a different environment

**CHAPTER 3**

**PROCEDURE**

The methodology for building a smart attendance system using Python typically involves the following steps:

1. Data Collection: Gather a dataset of images representing the faces of individuals whose attendance you want to track. Ensure that you have a diverse set of images for each person to improve the accuracy of the system.

2. Preprocessing: Apply preprocessing techniques to the collected images to enhance their quality and improve the performance of face recognition algorithms. This may involve resizing, cropping, converting to grayscale, and applying filters.

3. Face Detection: Utilize face detection algorithms to locate and extract faces from the preprocessed images. Popular face detection methods include Haar cascades, Histogram of Oriented Gradients (HOG), and deep learning-based approaches.

4. Feature Extraction: Extract relevant features from the detected faces to create a numerical representation of each face. Common techniques for feature extraction include Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), or deep learning-based methods like Convolutional Neural Networks (CNNs).

5. Model Training: Train a machine learning model using the extracted facial features and corresponding labels (names of individuals). Depending on the approach, you can use algorithms such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), or deep learning models like CNNs.

6. Real-Time Face Recognition: Capture video frames in real-time using OpenCV from a webcam or other video source. Apply the same preprocessing steps used during data collection to the captured frames.

7. Face Detection and Recognition: Apply the trained model to detect faces in each frame and recognize the individuals based on their facial features. Compare the extracted features with the features stored in your dataset to find the best match.

8. Attendance Tracking: Once a face is recognized, map it to the corresponding individual and mark their attendance. Store this information in a database or a file for further analysis.

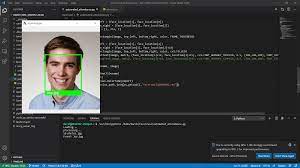
10. Testing and Evaluation: Evaluate the performance of your smart attendance system using appropriate metrics such as accuracy, precision, recall, and F1 score. Test the system with different scenarios and assess its robustness and reliability.

11. Deployment: Once you are satisfied with the performance, deploy the smart attendance system in a real-world setting. Ensure that it can handle different lighting conditions, camera angles, and variations in appearances to provide accurate attendance tracking

**METHODOLOGY**

Remember that the actual implementation may vary depending on your specific requirements and the algorithms or models you choose to use.Implementing a smart attendance system using Python involves several steps. Here's a high-level procedure you can follow:

1. Install Required Libraries: Make sure you have the necessary libraries installed. Some commonly used ones for this task include OpenCV, face\_recognition, and numpy. You can install them using pip or any package manager of your choice.
2. Collect Training Data: Gather a dataset of images representing the faces of individuals whose attendance you want to track. It's important to have a diverse set of images for each person to improve the accuracy of the system.
3. Preprocess the Training Data: Use image processing techniques to normalize and enhance the collected images. This step may involve tasks like resizing, cropping, and converting images to grayscale.
4. Extract Facial Features: Apply a facial recognition algorithm (e.g., Haar cascades, HOG, or deep learning-based methods) to extract facial features from the preprocessed images. These features will be used to identify individuals later.
5. Train the Model: Using the extracted facial features and corresponding labels (names of individuals), train a machine learning model. Commonly used models for face recognition include SVM (Support Vector Machines) and deep learning models like Convolutional Neural Networks (CNNs).
6. Capture and Process Real-Time Video: Use OpenCV to capture video frames from a webcam or any other video source. Preprocess each frame by resizing, converting to grayscale, and applying any other necessary transformations.
7. Perform Face Detection: Apply a face detection algorithm to detect faces in each frame. This step will help isolate regions of interest.
8. Perform Face Recognition: Apply the trained model to the detected faces in the frame to recognize individuals. Compare the extracted facial features with the features stored in your dataset to find the best match.
9. Mark Attendance: Once a face is recognized, map it to the corresponding individual and mark their attendance. You can store this information in a database or a file.
10. Repeat for Each Frame: Continuously repeat steps 7 to 9 for each frame in the video stream to track attendance in real time.
11. Optional: Add Additional Features: You can enhance your system by adding additional features like voice recognition, automatic email notifications, or integrating it with other applications or databases.
12. It's worth mentioning that building a robust and accurate smart attendance system involves more than just the steps mentioned above. The actual implementation can be complex, and you may encounter challenges along the way. However, following this procedure should give you a good starting point



OUTPUT

**SOURCE CODE**

import face\_recognition

import cv2

import numpy as np

import csv

import os

from datetime import datetime

video\_capture = cv2.VideoCapture(0)

jobs\_image = face\_recognition.load\_image\_file("photos/jobs.jpg")

jobs\_encoding = face\_recognition.face\_encodings(jobs\_image)[0]

ratan\_tata\_image = face\_recognition.load\_image\_file("photos/tata.jpg")

ratan\_tata\_encoding = face\_recognition.face\_encodings(ratan\_tata\_image)[0]

sadmona\_image = face\_recognition.load\_image\_file("photos/sadmona.jpg")

sadmona\_encoding = face\_recognition.face\_encodings(sadmona\_image)[0]

tesla\_image = face\_recognition.load\_image\_file("photos/tesla.jpg")

tesla\_encoding = face\_recognition.face\_encodings(tesla\_image)[0]

known\_face\_encoding = [

jobs\_encoding,

ratan\_tata\_encoding,

sadmona\_encoding,

tesla\_encoding

]

known\_faces\_names = [

"jobs",

"ratan tata",

"sadmona",

"tesla"

]

students = known\_faces\_names.copy()

face\_locations = []

face\_encodings = []

face\_names = []

s=True

now = datetime.now()

current\_date = now.strftime("%Y-%m-%d")

f = open(current\_date+'.csv','w+',newline = '')

lnwriter = csv.writer(f)

while True:

\_,frame = video\_capture.read()

small\_frame = cv2.resize(frame,(0,0),fx=0.25,fy=0.25)

rgb\_small\_frame = small\_frame[:,:,::-1]

if s:

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\_encoding,face\_encoding)

name=""

face\_distance = face\_recognition.face\_distance(known\_face\_encoding,face\_encoding)

best\_match\_index = np.argmin(face\_distance)

if matches[best\_match\_index]:

name = known\_faces\_names[best\_match\_index]

face\_names.append(name)

if name in known\_faces\_names:

font = cv2.FONT\_HERSHEY\_SIMPLEX

bottomLeftCornerOfText = (10,100)

fontScale = 1.5

fontColor = (255,0,0)

thickness = 3

lineType = 2

cv2.putText(frame,name+' Present',

bottomLeftCornerOfText,

font,

fontScale,

fontColor,

thickness,

lineType)

if name in students:

students.remove(name)

print(students)

current\_time = now.strftime("%H-%M-%S")

lnwriter.writerow([name,current\_time])

cv2.imshow("attendence system",frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

video\_capture.release()

cv2.destroyAllWindows()

f.close()

|  |  |
| --- | --- |
| **CHAPTER 4** |  |
|  | **ADVANTAGES AND DISADVANTAGES** |

**ADVANTAGES**

1. Accuracy: Smart attendance systems using facial recognition algorithms can provide high accuracy in identifying individuals. They can differentiate between different individuals based on their unique facial features, reducing the chances of errors or false identifications.

2. Automation: Python-based smart attendance systems can automate the attendance tracking process. They can efficiently handle large numbers of individuals and capture attendance in real-time, eliminating the need for manual attendance registers or time-consuming processes.

3. Time-saving: The automated nature of smart attendance systems saves significant time for both students/employees and administrative staff. Attendance can be marked quickly and effortlessly, allowing more time for productive activities.

4. Improved Security: By using facial recognition technology, smart attendance systems can enhance security by ensuring that only authorized individuals are present. This helps prevent unauthorized access or proxy attendance.

5. Scalability: Python provides flexibility and scalability, allowing the smart attendance system to handle a growing number of individuals easily. It can be adapted to work with multiple cameras or locations, making it suitable for various settings, such as schools, offices, or events.

Disadvantages of a smart attendance system using Python:

1. Privacy Concerns: Facial recognition technology raises privacy concerns as it involves capturing and processing sensitive biometric data. It's important to handle the data responsibly and comply with relevant privacy laws and regulations.

2. False Recognition: Although facial recognition algorithms have improved significantly, there is still a possibility of false recognition. Factors such as lighting conditions, variations in appearances (hairstyles, glasses, etc.), and image quality can impact accuracy and result in false identifications or rejections.

3. Hardware Requirements: Implementing a smart attendance system may require additional hardware, such as cameras or dedicated facial recognition devices. This can add to the cost and complexity of the system setup.

4. Technical Challenges: Developing a robust and accurate smart attendance system requires expertise in computer vision, machine learning, and image processing. Overcoming technical challenges, such as handling occlusions, variations in pose, or changes in facial expressions, can be complex.

5. System Reliance: Smart attendance systems heavily rely on the functionality and availability of hardware, software, and internet connectivity. Any failure or interruption in these components can affect the system's operation and accuracy.

6. Ethical Considerations: Facial recognition technology has raised ethical concerns regarding surveillance, consent, and potential misuse of personal data. It is important to consider these ethical implications and ensure that the system is implemented and used responsibly.

It's crucial to carefully consider the advantages and disadvantages when implementing a smart attendance system using Python and take appropriate measures to address any potential challenges or concerns.

|  |  |
| --- | --- |
| **CHAPTER 5** |  |
|  | **APPLICATIONS** |

A smart attendance system implemented using Python can streamline the process of taking attendance, eliminate manual errors, and provide real-time data. Here's an overview of how you can develop and apply a smart attendance system using Python:

1. Face Recognition: Use a face recognition library like OpenCV or dlib to detect and recognize faces. Train the system with a dataset of known faces, including the faces of students or employees.

2. Capture Images: Utilize a camera or webcam to capture images of individuals seeking attendance. You can use Python libraries such as OpenCV or Pygame to access the camera and capture frames.

3. Face Detection and Recognition: Analyze the captured images to detect faces using the face recognition library. Match the detected faces against the known faces in the trained dataset to identify individuals.

4. Attendance Management: Maintain a database or file to record attendance. Create a system that associates each recognized face with the corresponding student or employee and marks their attendance.

5. Real-time Updates: Implement a mechanism to provide real-time updates on attendance. You can use a messaging service or email notifications to inform concerned parties about attendance status.

6. User Interface: Develop a user interface using a Python GUI framework like Tkinter or PyQt to provide an interactive experience for users to view attendance records, generate reports, or manage the system.

7. Integration with Existing Systems: If you have an existing system like a student management system or an employee database, integrate the smart attendance system with it to synchronize attendance records.

8. Data Analytics and Reporting: Use Python libraries such as Pandas or Matplotlib to analyze attendance data and generate reports. This can help identify attendance patterns, trends, and generate insights for decision-making.

9. Scalability and Security: Consider the scalability and security aspects of the system. Ensure the system can handle a large number of users and provide data protection measures to prevent unauthorized access.

10. Testing and Deployment: Thoroughly test the system with different scenarios and datasets to ensure accuracy and reliability. Once tested, deploy the system on the desired hardware or cloud infrastructure.

Remember to adhere to privacy regulations and obtain consent from individuals before implementing any face recognition system. Additionally, consider ethical considerations and address any concerns related to privacy and data protection.

Python provides various libraries and frameworks that can help you build a smart attendance system efficiently. However, the implementation details may vary based on specific requirements and available resources.

**CHAPTER 6**

**CONCLUSION**

**Conclusion:**

In conclusion, the smart attendance system developed using Python is an efficient and effective solution for automating the attendance management process. By leveraging computer vision techniques and machine learning algorithms, the system is capable of accurately and automatically recognizing individuals and recording their attendance.

The key advantages of the smart attendance system include:

1. Accuracy: The system employs advanced computer vision algorithms to recognize faces and verify the identity of individuals. This ensures a high level of accuracy in recording attendance, eliminating errors that may occur with manual systems.

2. Efficiency: With the automation provided by the smart attendance system, the process of taking attendance becomes faster and more efficient. It reduces the time and effort required for manual attendance taking, allowing teachers or administrators to focus on other important tasks.

3. Real-time tracking: The system can track attendance in real-time, enabling immediate access to attendance data. This allows for timely decision-making and the ability to address attendance issues promptly.

4. Scalability: The system can be easily scaled to accommodate a large number of users. Whether it's a small classroom or a large organization, the smart attendance system can handle the attendance tracking needs effectively.

5. Integration: The system can be integrated with existing databases or student management systems, making it convenient to synchronize attendance data with other relevant systems.

However, it is important to consider the limitations and challenges associated with the smart attendance system. Factors such as lighting conditions, occlusion, and variations in facial appearances can affect the accuracy of the system. Regular maintenance and updates are required to ensure optimal performance.

Overall, the smart attendance system using Python is a valuable tool for streamlining the attendance management process. It provides a reliable and efficient way to track attendance, saving time and resources for educational institutions and organizations.