

Event Attendance System using YOLOv12

Name-

1. Sneha Konade B-43
2. Kushagra Chaubey B-27
3. Shailaja Shiraphanali B-39
4. Trisha Shetty B-40

Department- B-tech (CSE)

Guide Name- Dr. Sagar D. Pande

**CEP Project Report**

**School of Engineering and Technology**

**Programme:** B-Tech CSE

CEP Project Title: Event Attendance System using YOLOv12

**Student Detail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Name** | **Roll No** | **Mobile No** |
| 1 | Sneha Konade | 43 | 8483978029 |
| 2 | Kushagra Chaubey | 27 | 7666671795 |
| 3 | Shailaja Shiraphanali | 39 | 9834496072 |
| 4 | Trisha Shetty | 40 | 7972845152 |

**Supervisor Detail**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Sign** |
| Dr. Sagar D. Pande | sagar.pande@pcu.edu.in |  |

**Acknowledgement**

We would like to express our sincere gratitude to Mr Sagar Dhanraj Pande for his invaluable guidance and support throughout the development of this project. His expertise and encouragement were instrumental in helping us navigate the challenges and successfully complete our work. We would also like to thank the faculty and staff of the **Department of Computer Science and Engineering, Pimpri Chinchwad University**, for providing the necessary resources and a conducive environment for conducting this project.

**Abstract**

This project introduces a face recognition- based attendance system using YOLOV12 to automate and streamline attendance tracking and events. The System captures attendees’ facial images compares them with stored data, and instantly record attendance. This approach enhances security, prevents proxy check- ins, and improves event efficiency by reducing manual errors and waiting times.

**Introduction:**

Traditional attendance methods like manual registers and QR codes are time-consuming and prone to errors. This project introduces a face recognition-based system using YOLOV12 to automate attendance tracking securely and efficiently. By capturing and verifying facial images in real time, it eliminates proxy check-ins and reduces manual effort. This project explores the development of such a system, utilizing YOLOV12 for real time face detection and recognition. Unlike systems that rely on cloud-based backend.

Services like Firebase, this system emphasises local processing and data management to enhance speed and potentially reduce dependency on constant internet- connectivity. By automating the attendance process, this system aims to minimize manual errors, prevent proxy attendance, and improve overall event management efficiency.

This approach not only streamlines event operations but also enhances security and improves user experience. By reducing long queues and automating attendance tracking, the system provides a more efficient and modern solution for event organizers.

**Proposed System Architecture**

**Overview:**

The architecture is designed to perform **real-time face detection and attendance marking** using YOLOv12, running entirely on **Google Colab**. It utilizes **local processing**, avoiding cloud APIs and ensuring fast, offline-capable execution. The system captures input from a webcam or video file, detects faces, compares them with pre-registered data, and logs attendance in a structured CSV file.

**System Components:**

**1. Input Module**

* **Source**: Webcam feed (via OpenCV) or uploaded video/image
* **Function**: Captures real-time frames to feed into the YOLOv12 model
* **Tools**: cv2.VideoCapture() for webcam or cv2.imread() for image

**2. Face Detection Module**

* **Model**: YOLOv12 pretrained weights for face detection
* **Function**: Detects faces in real-time from the captured frames
* **Output**: Bounding boxes of detected faces with confidence scores
* **Tools**: Python, OpenCV, YOLOv12 weights and configuration files

**3. Face Recognition & Matching**

* **Method**: Facial feature extraction (embedding) and comparison with known encodings
* **Process**:
  + Extract face region from YOLO detection
  + Convert it to embedding (e.g., using face\_recognition library or custom descriptor)
  + Compare embedding with pre-stored registered encodings using a similarity threshold
* **Result**: If matched → mark as "Present"

**4. Attendance Logging Module**

* **Output Format**: attendance.csv
* **Data Logged**:
  + Name of the matched person
  + Date
  + Time
  + Status ("Present")
* **Tools**: Python csv and datetime libraries

**5. Storage module**

* **Type**: Local (Google Colab temporary storage or downloadable CSV)
* **Stored Data**:
  + Known face embeddings
  + CSV file with attendance logs

## **Workflow / Data Flow Diagram**

## 

## **Methodology**

The Methodology will encompass the following key stages:

**1. Requirement Analysis**

This initial phase will involve a thorough analysis of the system requirements, including the desired accuracy of face recognition, the speed of attendance processing, the number of attendees the system needs to handle, security considerations and data storage needs.

**2. System Design**

This stage will focus on designing the system architecture, outlining the flow of data, and defining the interaction between different modules. It will include:

* decisions about how YoloV12 will be integrated
* how facial data will be stored and managed locally
* the design of the user-interface for both attendees and administration

**3. Development**

This is where the system is built. It involves:

* Implementing the frontend for user interaction.
* Integrating YoloV12 for face detection and recognition
* Developing the logic for attendance recording and data management

**4. Testing and Validation**

Rigorous testing will be conducted to evaluate the system’s performance. This will include:

* accessing the accuracy of face-recognition under various conditions
* measuring the speed of attendance processing
* testing the system’s robustness to handle variations in lighting and pose
* validating the data storage and retrieval mechanisms.

1. **Deployment and Monitoring**

This involves setting of the system for real world system and continuously monitoring its performance to ensure reliability and identify areas for improvement.

1. **Future Enhancement**

This stage will explore potential future improvements, such as incorporating liveness detections to prevent spoofing, optimizing the system for larger events, and adding more sophisticated data analysis and reporting features.

**Design and Implementation**

The system architecture is designed to perform efficient face recognition- based attendance tracking. It comprises of the following key components:

1. **Frontend(user-interface):** This component provides an intuitive interface for attendees to register and check-in, and for administrators to manage events and view attendance records. It will be developed as a web application to ensure accessibility across devices
2. **YOLOv12 integration**: YOLOv12 will be the core of the face detection and recognition module. It will be integrated to process the real-time video or image input, detect faces, and identify individuals based on pre-registered data.
3. **Local data storage and management**: Instead of relying on cloud-based database, this system will implement local data storage. This will include storing facial embeddings and associated attendee information, developing efficient algorithms for searching and matching faces within the local database, implementing data backup and security measures to protect attendee information.
4. **Attendance recording module**: This module will record the attendance of identified individuals, logging the time of check-in and associating it with the relevant event.

**Results and Discussion**

This section will present a comprehensive analysis of the system’s performance and capabilities. Key results will include:

1. **Face-recognition accuracy**:

Quantifying the system’s ability to correctly identify attendees under various conditions.

1. **Processing speed**:

Measuring the time taken to detect and recognize faces and record attendance, crucial for efficient event management

1. **System’s robustness**:

Evaluating the system’s ability to handle variations in input and potential errors.

1. **Comprising with the existing methods**:

Comparing the system’s performance to traditional attendance tracking methods in terms of speed, accuracy and efficiency.

1. **Discussions of trade-offs:** Analysing the advantages and disadvantage of using YoloV12 and local data storage compared to cloud-based solutions, considering factors such as speed, scalability, complexity and security implications.
2. **Impact on Event Management**: Discussing how the system can improve event management practices and the potential benefits for organisers and attendees

**Conclusion:**

This project demonstrates the development of an automated attendance tracking system leveraging YoloV12 for face recognition. By automating the attendance process, the system reduces the reliance on manual methods, improve accuracy and minimize the potential for fraud. This contributes to a more efficient and streamlined event management experience. However, it is crucial to acknowledge the ethical considerations surrounding the use of facial recognition technology, particularly concerning privacy and data security. Future implementations must prioritize responsible data handling practices, transparency and adherence to relevant regulations to ensure the ethical deployment of such systems.

**Reference:**

 **“Real-Time Face Detection and Recognition Using YOLO”**  
**Authors:** N. Maheshwari, S. Jain, A. Agarwal  
**Summary:** Demonstrates integration of YOLO with face recognition for real-time surveillance and attendance marking. Highlights the speed and efficiency of YOLO in detecting multiple faces.

 **“YOLOv4: Optimal Speed and Accuracy of Object Detection”**  
**Authors:** Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao  
**Summary:** Explores improvements in the YOLO family of models and benchmarks YOLOv4 for real-time detection. Acts as a foundation for understanding the evolution towards YOLOv12.

 **“A Review on Attendance Marking System Using Face Recognition”**  
**Authors:** S. Dey, A. Chakraborty, A. Pal  
**Summary:** Reviews biometric attendance systems and identifies face recognition as the most efficient contactless option.

 **“FaceNet: A Unified Embedding for Face Recognition and Clustering”**  
**Authors:** Florian Schroff, Dmitry Kalenichenko, James Philbin  
**Summary:** Introduces FaceNet, a deep learning model that generates embeddings for face recognition. Useful for face matching logic in your project.

 **“A Secure Attendance Management System Based on Face Recognition using Raspberry Pi”**  
**Authors:** G. R. Reddy, M. Prasad, D. Rajasekhar  
**Summary:** Proposes an offline face recognition system for attendance, using local hardware — similar in motivation to your Google Colab implementation.

 **“Performance Evaluation of Face Detection using YOLO for Surveillance Applications”**  
**Authors:** M. Shah, A. Verma  
**Summary:** Benchmarks YOLO’s face detection under various conditions and concludes it's ideal for real-time security and monitoring use cases.

 **“Comparison of Cloud-Based and Local Face Recognition Systems”**  
**Authors:** H. Singh, P. Sharma  
**Summary:** Analyzes pros and cons of using cloud vs local computation for face recognition systems. Your system aligns with the local-processing preference.