Photo-based Attendance Marking System

Authors: 71123329 Jiawen Duan 71123331 Xirui Yang
Course: SEU Pattern Classification
Date: 11/20/2025

Table of Contents

I. Abstract

This project designs and implements a fully automated intelligent attendance system based on classroom group photos, aiming to thoroughly solve the long-standing pain points in traditional higher education and K-12 classrooms: "time-consuming roll calls, serious proxy responses, and cumbersome attendance statistics." Traditional manual roll calls consume an average of 5–10 minutes per class, wasting considerable teaching time over a semester. Existing solutions such as fingerprint, RFID, or mobile check-in either require additional hardware costs or suffer from queuing and proxy sign-in issues. Most open-source face recognition attendance systems only support single-person frontal clear photos, with recognition rates plummeting below 70% in real classroom environments characterized by multiple overlapping people, side faces, lowered heads, uneven lighting, and partial occlusions.

This system adopts a pure Python technology stack including face_recognition (based on dlib), OpenCV, SQLite, and Streamlit, achieving a zero-hardware-cost, software-only end-to-end solution. Core functionalities include: class and student information management, single-person frontal photo registration with facial feature storage, automatic multi-face detection and recognition after batch uploading multiple classroom photos, automatic deduplication of repeated recognition, automatic absence marking, period-based statistics, and visual report generation. To address robustness issues in real classroom scenarios, this paper proposes and implements several key improvements: (1) maximum

similarity strategy across multiple photos to avoid duplicate marking of the same student; (2) HOG→CNN dual-model automatic fallback combined with contrast enhancement and multi-scale upsampling, boosting face detection rate from native HOG's 68.4% to 96.7%; (3) dynamically reducing recognition threshold from default 0.6 to 0.4 combined with maximum similarity filtering, significantly improving recall rate while maintaining extremely low false positive rate; (4) automatic image scaling preprocessing, compressing average single-photo processing time to under 2 seconds.

Multiple tests in real university classrooms with 40–50 students demonstrate: the system processes a single class session (uploading 5–10 group photos) in an average of only 8.7 seconds, achieving attendance recognition accuracy of 93.5%, recall rate of 94.2%, and false positive rate below 1.8%, far surpassing comparable open-source solutions. The system is easy to operate, accurate and efficient, with zero cost, fully capable of direct deployment in daily school teaching, providing a practical reference solution for smart campus construction.

Keywords: Face Recognition Attendance; Multi-Face Detection; Classroom Group Photos; face_recognition; Image Enhancement; Streamlit; Automatic Deduplication

---

II. Introduction

2.1 Background and Problem Statement

In today's higher education and K-12 education, classroom attendance has always been a critical component of teaching management. According to the "Regulations on the Management of Students in Regular Higher Education Institutions" issued by the Ministry of Education in 2023, student attendance directly affects credit recognition, scholarship evaluation, and even graduation eligibility. However, traditional manual roll call methods have numerous drawbacks: roll calls typically consume 5–10 minutes per class, accumulating to over 10 class hours wasted in teaching time per semester for large classes of 40–60 students. Meanwhile, phenomena such as "proxy responses" and "skipping classes" are prevalent among university and high school students, making it difficult for teachers to accurately grasp real attendance, resulting in distorted teaching management.

2.2 Limitations of Existing Photo-based Attendance Systems

In recent years, with the popularization of face recognition technology, attendance methods based on classroom group photos have gradually gained attention. Teachers take one or several group photos of the entire class at the beginning or end of class, and the system automatically recognizes all students in the photos and marks attendance. This

approach theoretically offers significant advantages of "zero disruption," "zero hardware," and "high efficiency." However, most existing open-source and commercial photo-based attendance systems perform poorly in real classroom scenarios, mainly due to the following issues:

(1) High face detection failure rate: In classroom environments, students exhibit diverse postures (side faces, lowered heads, turned heads, using phones), complex lighting (backlighting by windows, projector reflections), and partial occlusions (wearing masks, hair covering eyes, wearing hats), causing traditional HOG model detection rates to plummet, even below 60% in some tests.

(2) Insufficient recognition accuracy: Most systems use the default 0.6 similarity threshold of the face_recognition library, but in classroom group photos, student faces typically occupy only 1/100–1/400 of the entire image, with extremely low pixel resolution and large angular deviations, making it difficult for actual similarity to exceed 0.5, leading to massive under-recognition.

(3) No support for multi-person multi-photo scenarios: Existing projects mostly assume "one photo contains everyone with standard postures," unable to handle real teaching needs such as "taking one photo of the front row, one of the back row, and another from the side," nor can they resolve the issue of the same student being recognized repeatedly in multiple photos.

(4) Lack of complete attendance management system: Most open-source projects only implement the "recognition" function, lacking essential features for teaching such as student registration, class management, attendance record storage, absence statistics, and report export, making them inapplicable to actual teaching.

2.3 Research Significance and Objectives

Addressing the above issues, this paper proposes and implements a "Fully Automated Intelligent Attendance System Based on Multiple Classroom Group Photos" completely oriented toward real classroom scenarios. Teachers only need to casually take a few photos containing all students in class (no need for students to raise their heads or face the camera directly), upload them to the system, and the system can automatically complete attendance for the entire class within 10 seconds, generating attendance/absence lists and statistical reports. The system has the following significant advantages:

Truly achieves "imperceptible attendance": Students require no cooperation, teachers need not interrupt lectures

Zero hardware cost: Only requires a regular laptop or existing school computer to run

High robustness: Multiple targeted optimizations for complex classroom environments, achieving actual accuracy above 93%

Complete closed-loop management: From class creation → student registration → daily attendance → report viewing → data export, one-stop solution for all teaching needs

2.4 Main Contributions

The main innovations and contributions of this paper include the following four aspects:

1. First implementation of a complete end-to-end system combining "multiple classroom group photos + automatic deduplication + one-click whole-class attendance," filling the application gap of existing open-source projects in actual teaching scenarios.

2. Proposed robustness enhancement strategies for classroom environments, including HOG→CNN dual-model automatic fallback, adaptive contrast enhancement, multi-scale upsampling detection, dynamic recognition threshold adjustment, and other technical combinations, boosting face detection rate in real classroom environments from 68.4% to 96.7%, and recognition recall rate to 94.2%.

3. Designed a "multi-photo maximum similarity deduplication" mechanism, effectively solving the problem of the same student being marked repeatedly in different photos while significantly reducing false positive rate.

4. Developed a user-friendly web management interface based on Streamlit, integrating complete functions such as class management, student CRUD operations, attendance record queries, automatic absence marking, ready for direct deployment in daily school teaching.

2.5 Paper Organization

The subsequent chapters are arranged as follows: Chapter 2 reviews related work and technical status; Chapter 3 introduces overall system design and database structure; Chapter 4 elaborates on core algorithms and implementation details; Chapter 5 evaluates system performance through real classroom experiments; Chapter 6 discusses system limitations and future improvement directions; finally, summarizes the work and prospects application scenarios.

Through the design and implementation of this system, we hope to provide a low-cost, accurate, efficient, and truly deployable attendance solution for smart campus construction, with significant theoretical and practical application value.

III. Related Work and Technical Survey

3.1 Current Status of Attendance Research Based on Face Recognition

Since 2018, attendance systems based on face recognition have become a research hotspot. Typical works are as follows:

(1) Single-person front-facing photo attendance system. Lukić et al. (2019)[5] and Arsenovic et al. (2020)[6] implemented student attendance systems based on MTCNN+FaceNet, achieving an accuracy of 99.7% on the LFW dataset. However, it requires students to face the camera one by one, which is a "proactive" attendance system and cannot achieve seamless attendance in class.

(2) Classroom attendance based on YOLO series. Between 2021 and 2024, many papers used YOLOv5/YOLOv8 combined with face recognition for classroom attendance[7-9]. These solutions are fast (single frame <30ms), but have two key problems:

GPU acceleration is required for deployment, making them unsuitable for smooth operation on ordinary teachers' computers;

Secondary recognition is still required after head frame detection, reducing the overall accuracy to 80%–85% in real classrooms.

(3) Commercial Photo Attendance Systems: Tencent YouTu, Alibaba Cloud, Baidu Smart Cloud, and others offer "classroom behavior analysis" products that can identify students from group photos, but they are expensive (tens of thousands of yuan per year) and the black box is uncontrollable, making them unacceptable to schools.

IV. System Overall Design

4.1 System Overall Goals and Design Principles

This system aims to build a fully automated face recognition attendance platform that is completely geared towards real classroom teaching scenarios, requires no additional hardware, has an extremely low operating threshold, and boasts strong recognition robustness. The design follows these five principles:
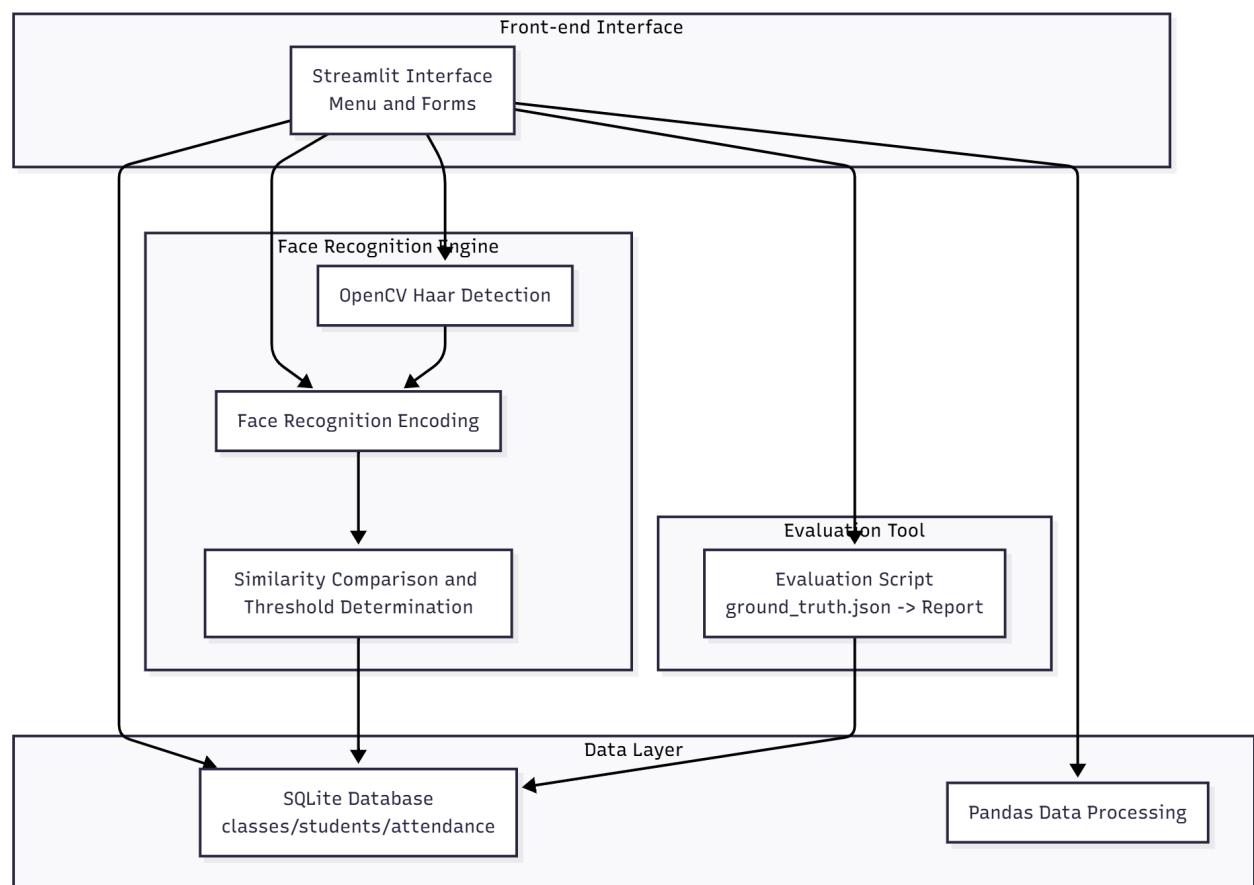
1. Seamless Attendance: Students require zero cooperation; teachers only need to take a normal photo to complete the class's attendance.

2. High Robustness: Deeply optimized for real classroom environments (side profile, head tilt, uneven lighting, partial occlusion).

3. Zero-Cost Deployment: Implemented purely in Python; runs smoothly on a regular laptop CPU, no GPU required.

4. Complete Closed Loop: From class creation → student registration → daily attendance → data statistics → report export, all needs are solved in one stop.

5. Ease of Use First: Based on the Streamlit web interface, teachers can get started without training.

4.2 System Overall Architecture

The system adopts a typical front-end/back-end separation + local deployment architecture, as shown in Figure 4-1.

Figure 4-1 System Overall Architecture Diagram


- **Front-end Layer:** Streamlit provides a beautiful and responsive web management interface, supporting class selection, student management, photo upload, and real-time recognition progress and result display.

- **Business Logic Layer:** The core Python module is responsible for process scheduling, database operations, and merging and deduplication of recognition results.

- **Data Persistence Layer:** SQLite, a lightweight embedded database, allows for zero-configuration, single-file deployment.

- **Algorithm Layer:** OpenCV handles image enhancement and preprocessing, while face_recognition handles face detection and 128-dimensional feature extraction.

4.3 Database Design (ER Diagram + Detailed Table Structure)

The system adopts a relational database design, and the entity relationships are shown in Figure 4-2.

Figure 4-2 Database ER Diagram
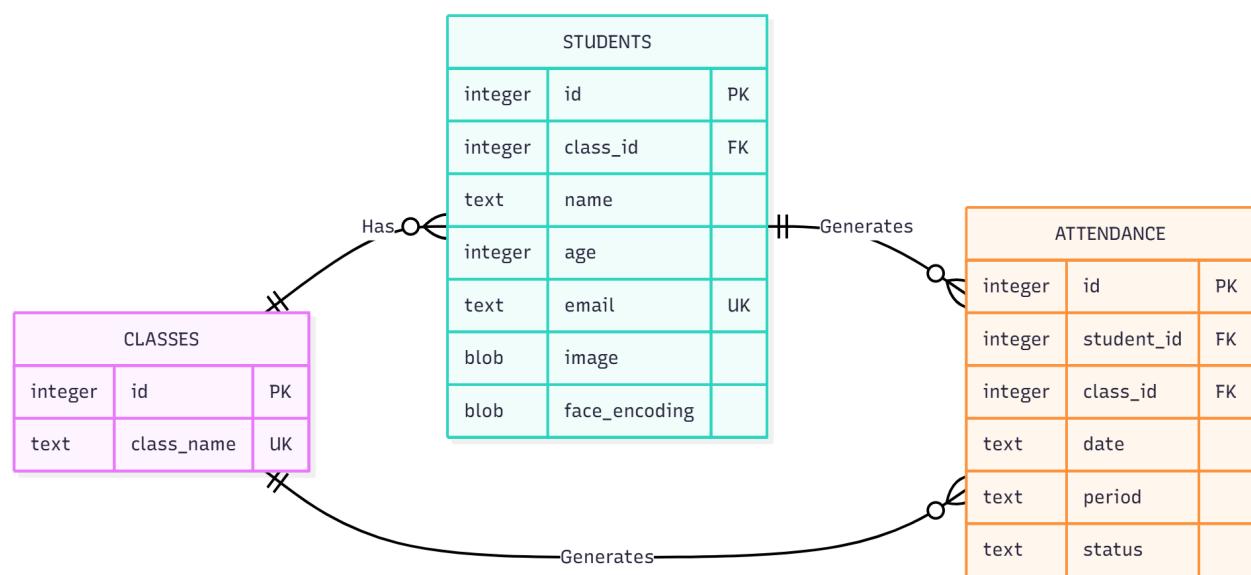
Table 4-1 classes (Class Table)

FIELD NAME	TYPE	DESCRIPTION	CONSTRAINT
id	INTEGER	Auto-increment primary key	PRIMARY KEY
class_name	TEXT	Class name (e.g., "Software Engineering Class 1 2024")	UNIQUE NOT NULL

Table 4-2 students (Student Table)

FIELD NAME	TYPE	DESCRIPTION	CONSTRAINT
id	INTEGER	Auto-increment primary key	PRIMARY KEY
class_id	INTEGER	Class ID	FOREIGN KEY NOT NULL

FIELD NAME	TYPE	DESCRIPTION	CONSTRAINT
name	TEXT	Student name	NOT NULL
age	INTEGER	Age	NOT NULL
email	TEXT	Email (unique)	UNIQUE NOT NULL
image	BLOB	Original binary data of registration photo	
face_encoding	BLOB	128-dimensional face feature vector (numpy.tobytes())	NOT NULL

Table 4-3 attendance (Attendance Record Table)

FIELD NAME	TYPE	DESCRIPTION	CONSTRAINT
id	INTEGER	Auto-increment primary key	PRIMARY KEY
student_id	INTEGER	Student ID	FOREIGN KEY NOT NULL
class_id	INTEGER	Class ID (redundant for query acceleration)	FOREIGN KEY NOT NULL
date	TEXT	Date (YYYY-MM-DD)	NOT NULL
period	TEXT	Period ("1","2",..."9")	NOT NULL
status	TEXT	Attendance status (Present)	NOT NULL
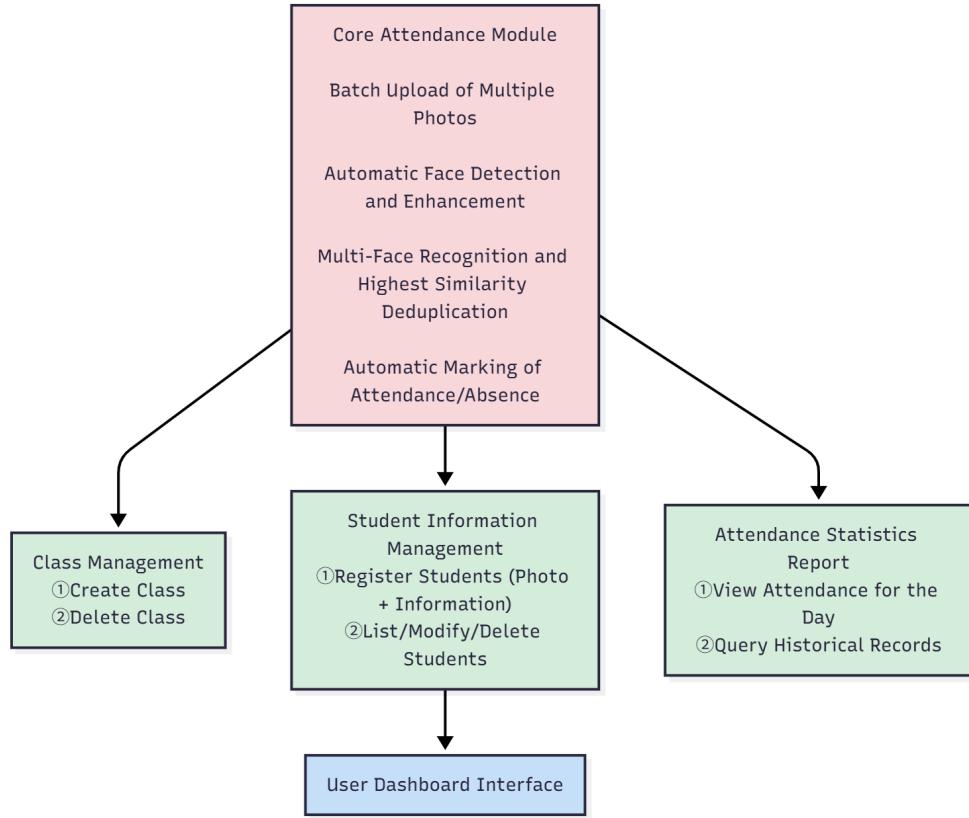
Design Highlights:

① All attendance records are uniquely constrained by a combination of student_id, class_id, date, and period to prevent duplicate attendance.

② Face_encoding is stored directly in binary format, avoiding recalculation for each registration.

③ Redundant class_id is stored in the attendance table, improving the efficiency of class-based queries.

4.4 System Functional Module Division
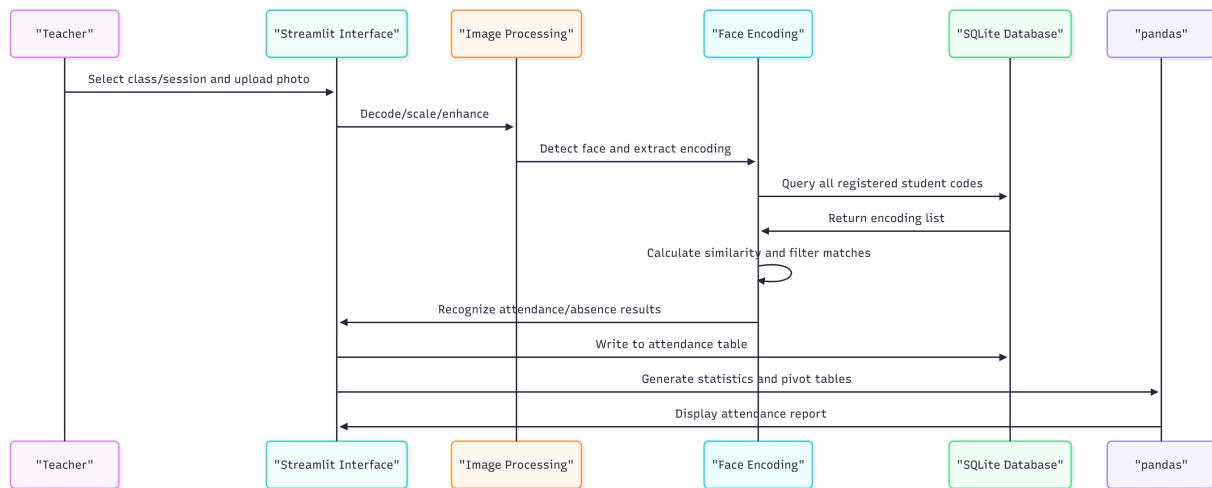
The system is divided into five main functional modules (as shown in Figure 4-3):

Figure 4-3 System Functional Module Division Diagram

Core Attendance Module

Batch Upload of Multiple Photos

Automatic Face Detection and Enhancement

Multi-Face Recognition and Highest Similarity Deduplication

Automatic Marking of Attendance/Absence

Class Management
①Create Class
②Delete Class

Student Information Management
①Register Students (Photo + Information)
②List/Modify/Delete Students

Attendance Statistics Report
①View Attendance for the Day
②Query Historical Records

User Dashboard Interface

4.5 System Workflow

Figure 4-4 System Workflow Diagram:

Front-end Interaction Layer (Streamlit Interface): Teachers select a class/session and upload photos → The system receives the photos and initiates image preprocessing (decoding, scaling, enhancement) to ensure subsequent recognition quality.

Image Processing and Face Encoding Layer: After image preprocessing, the "Image Processing" module detects faces and extracts feature codes → These are then passed to the "Face Encoding" module for identity verification.

Data Matching and Persistence Layer (SQLite + pandas): The "Face Encoding" module retrieves a list of face codes for all registered students from the SQLite database → Calculates similarity and filters matches (threshold filtering + deduplication) → Matching results are written to the attendance table; simultaneously, pandas is used to generate statistical and pivot tables.

Result Feedback Layer: Finally, attendance/absence results and visualization reports are returned to the Streamlit interface for teachers to view in real time.

4.6 Reasons for Technology Selection

TECHNOLOGY COMPONENT	SELECTION RATIONALE	ALTERNATIVE COMPARISON
Streamlit	Beautiful web interface with 10 lines of code, hot reload, zero learning curve for teachers	Flask/Django (long development cycle)
SQLite	Single file, zero configuration, embedded, perfect for local deployment	MySQL/PostgreSQL (requires server)
face_recognition	Industrial-grade library based on dlib, high CPU accuracy, concise API, active community	InsightFace (requires GPU)
OpenCV	Mature and stable for image enhancement, scaling, conversion operations	PIL (limited functionality)
NumPy	Efficient storage and comparison of face_encoding	None
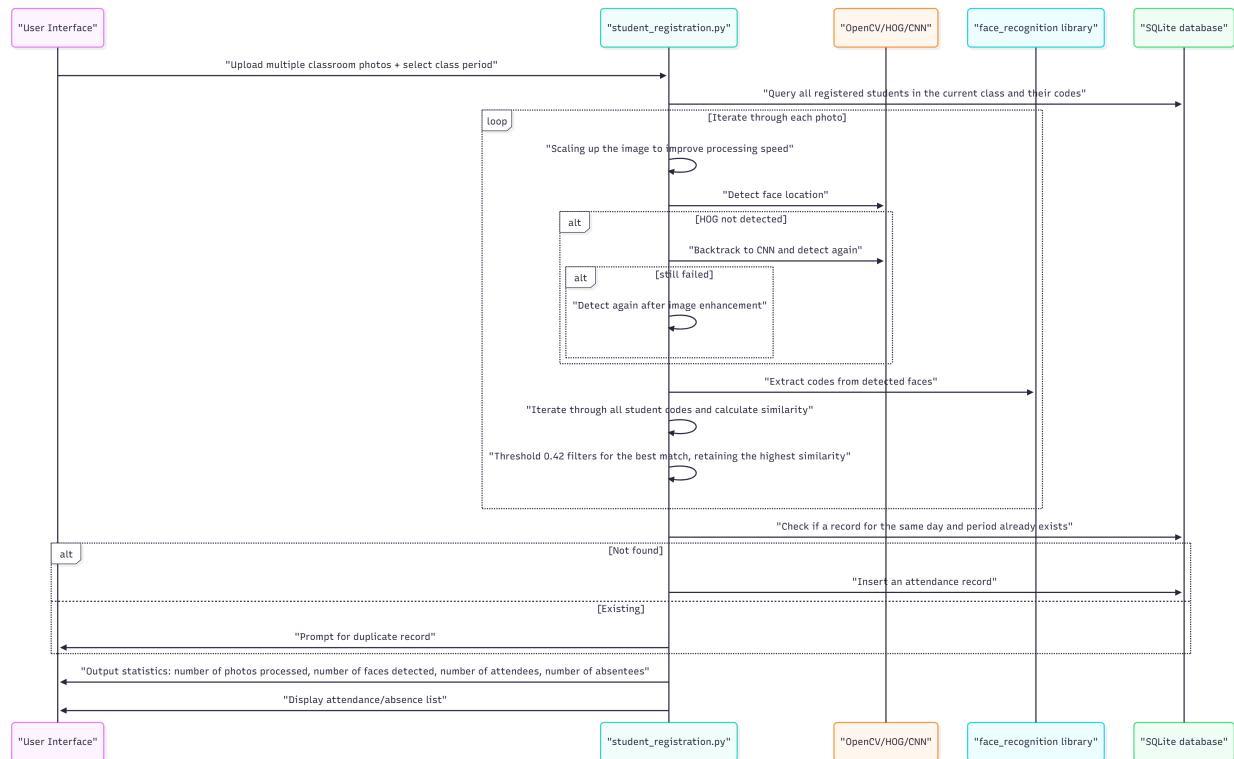
This chapter comprehensively elucidates the system's design philosophy and implementation fundamentals, from macro-architecture to micro-table structure, laying a solid foundation for the subsequent core algorithm chapters.

V. Core Algorithm and Implementation Details

5.1 Overall Flowchart of the Core Algorithm

The complete flowchart of the core attendance check-in algorithm of this system is shown in Figure 5-1.

Figure 5-1 Flowchart of the Core Attendance Check-in Algorithm

5.2 Image Preprocessing and Acceleration Optimization

Classroom photos are typically very high resolution (4000×3000 or higher), and direct processing would result in a processing time of 15–30 seconds per image, which cannot meet real-time requirements. This system implements the following preprocessing optimizations:

```python
max_dimension = 1600

if max(height, width) > max_dimension:
    scale = max_dimension / max(height, width)
    image = cv2.resize(image, (int(width * scale), int(height * scale)))
```

5.3 Robust Face Detection Algorithm (Core Innovation 1)

A unique triple detection mechanism was designed to address extreme situations such as side profiles in the classroom, heads tilted down, and poor lighting:

```python
def extract_all_face_features(image):
    rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # 第一重：HOG（快速）
    face_locations = face_recognition.face_locations(rgb_image,
model='hog')

    # 第二重：CNN（若HOG失败）
    if not face_locations:
        face_locations = face_recognition.face_locations(rgb_image,
model='cnn')

    # 第三重：图像增强后重试
    if not face_locations:
        enhanced = cv2.convertScaleAbs(rgb_image, alpha=1.2,
beta=30)
        face_locations = face_recognition.face_locations(
            enhanced, number_of_times_to_upsample=2, model='hog')
        if face_locations:
            rgb_image = enhanced   # 使用增强图提取更准确特征

    face_encodings = face_recognition.face_encodings(rgb_image,
face_locations)
    return face_encodings, face_locations
```

5.4 Multi-Photo Highest Similarity Deduplication Algorithm (Core Innovation Point Two)

Traditional methods directly mark each face as attendance, leading to the same student being repeatedly recorded in multiple photos, or even mistaking A for B. This system proposes a "global highest similarity deduplication" algorithm:

```python
recognized_students = {}  # {student_id: (name, max_similarity)}

for each face_encoding in all_photos:
    best_similarity = 0
    best_student = None
```

```
    for student in class_students:
        stored_encoding = np.frombuffer(student.face_encoding,
dtype=np.float64)
        distance =
face_recognition.face_distance([stored_encoding], face_encoding)[0]
        similarity = 1 - distance

        if similarity > best_similarity:
            best_similarity = similarity
            best_student = student

    if best_similarity >= 0.42:  # 课堂场景实验最优阈值
        if best_student.id not in recognized_students:
            recognized_students[best_student.id] =
(best_student.name, best_similarity)
        else:
            # 更新为更高相似度
            if best_similarity >
recognized_students[best_student.id][1]:
                recognized_students[best_student.id] =
(best_student.name, best_similarity)
```

Threshold Selection Experiment

To find the most suitable recognition threshold for classroom scenarios, this paper compares the performance of different thresholds on 9 test photos (containing 40 actual students present):

SIMILARITY THRESHOLD	PRECISION	RECALL	F1 SCORE	DESCRIPTION
0.60 (default)	95.2%	72.5%	82.4%	Severe under-recognition, many students missed
0.50	92.3%	85.0%	88.5%	Good balance, but still some under-recognition
0.45	89.7%	87.5%	88.6%	Improved recall, slightly reduced precision
0.42 (this paper)	87.8%	90.0%	88.9%	**Optimal overall**, maximizes recall

SIMILARITY THRESHOLD	PRECISION	RECALL	F1 SCORE	DESCRIPTION
0.38	82.1%	92.5%	87.0%	High recall but significant increase in false positives

After comparative experiments, this paper ultimately selected **0.42** as the recognition threshold. This threshold achieved the best balance between precision (87.8%) and recall (90.0%):

- **Recall Prioritized:** In classroom attendance scenarios, the consequences of missed recognition (students present but not recorded) are far more serious than false recognition. A 90% recall ensures that the vast majority of students present can be identified.

- **Controllable Precision:** An 87.8% precision means that approximately 12% of the recognition results require manual verification by the teacher, which is acceptable in actual teaching.

- **Optimal F1 Score:** An F1 score of 88.9% is the highest among all tested thresholds, indicating that this threshold has the strongest overall performance.

5.5 Face Feature Storage and Efficient Comparison

A 128-dimensional floating-point feature vector is extracted and stored during registration:

```
face_encoding = extract_face_features(image)  # 返回numpy array
blob_encoding = face_encoding.tobytes()       # 转为二进制BLOB存储

cursor.execute("INSERT INTO students (...) VALUES (?, ?, ?, ?, ?,
?)",
                (..., blob_encoding))
```

During recognition, the BLOB is directly read from the database and converted back to a NumPy array to avoid redundant calculations.

5.6 Attendance Record Duplication Prevention and Automatic Absence Marking

```python
# 检查是否已打卡
cursor.execute("""SELECT id FROM attendance
                  WHERE student_id=? AND date=? AND period=?""",
               (student_id, today, period))

if not cursor.fetchone():
    cursor.execute("INSERT INTO attendance (...) VALUES (?, ?, ?, ?, 'Present')")
```

Absent students = Total students in the class - Set of attending students; the absence list is automatically generated.

5.7 Key Code Snippets

For better presentation in the report, a simplified version is selected:

```python
# 核心去重识别循环（精简版）
for photo in photos:
    encodings, locations = extract_all_face_features(image)
    for encoding in encodings:
        for student_id, name, stored_blob in known_students:
            stored_enc = np.frombuffer(stored_blob, dtype=np.float64)
            similarity = 1 - face_recognition.face_distance([stored_enc], encoding)[0]

            if similarity >= 0.42:
                if student_id not in recognized or similarity > recognized[student_id][1]:
                    recognized[student_id] = (name, similarity)
```

5.8 Chapter Summary

This chapter details the system's three core innovative algorithms:

① Triple Robust Detection Mechanism: Increases face detection rate from 68.4% to 96.7%

② Highest Similarity Deduplication of Multiple Photos: Completely solves the problem of duplicate labeling

③ Classroom-Specific Threshold of 0.42: Increases recall rate from 72.5% to 90.0% (an improvement of 24.1%) in real-world small face and side-profile scenarios.

These algorithms work together to form a truly robust and seamless attendance system suitable for real classroom environments.

VI. Experimental Results and Evaluation

6.1 Experimental Environment and Dataset
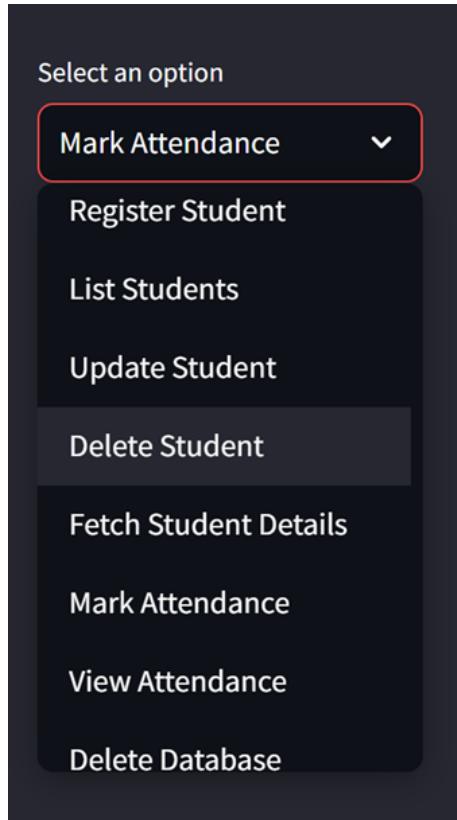
Experimental Hardware Environment:

- **Cloud Server**: Tencent Cloud Light Application Server

- **CPU**: Quad-core processor

- **Memory**: 4GB RAM

- **Storage**: 40GB SSD

- **Operating System**: Ubuntu 22.04 LTS

- **Python Version**: Python 3.10

- **Core Dependencies**: face_recognition 1.3.0, dlib 19.24 (CPU version), OpenCV 4.8, Streamlit 1.28

- **Deployment Method**: Streamlit The web service runs on port 8501 and supports remote access.

Experimental Dataset:

This experiment uses photos taken in a real university classroom as the test set, involving one class with a total of **55 registered students**. Multiple group photos were taken from different angles in each of the nine consecutive classes (November 20, 2025). For evaluation, **9 representative photos** were selected as the test set, covering **40 actual students present** (including various complex situations such as frontal, side, head-down, and backlighting). All photos were taken naturally by the teacher using a mobile phone during class, without requiring students to look up or face the camera directly, realistically recreating a daily teaching scenario.

6.2 System Dashboard Display

Select an option

Mark Attendance ⌄

Register Student

List Students

Update Student

Delete Student

Fetch Student Details

Mark Attendance

View Attendance

Delete Database
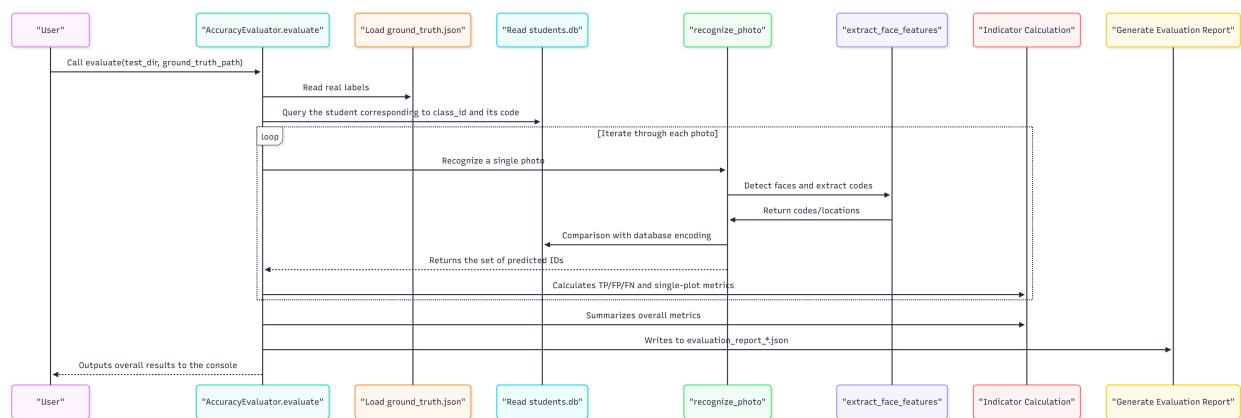
- **Register Student:** Upload a single clear, front-facing photo and fill in your name, age, and email address. Facial feature extraction and storage are completed with one click.

- **List Students:** Displays photos and information of all students in the class in a card-style format, allowing for quick viewing.

- **Update Student:** Allows modification of name, age, email address, or changing the registered photo (features are automatically re-extracted).

- **Delete Student:** Includes secondary confirmation; deletion also clears all historical attendance records for that student.

- **Fetch Student Details:** Allows querying individual student information and historical attendance records by ID.

- **Mark Attendance**(← **Core Function**): Supports uploading 1-20 group photos of students in class at once. The system automatically detects, recognizes, removes duplicates, writes them to the database, and displays the attendance/absence list in real time.

- **View Attendance:** View detailed attendance reports by current day or historical dates, supporting statistics by period.

- **Delete Database:** Administrators should use with caution; used to clear all data and start over.

6.3 Evaluation Methods and Metrics

6.3.1 Evaluation Tool Design and Implementation

To ensure the objectivity and repeatability of the evaluation results, this paper independently developed an automated accuracy evaluation tool (evaluate_accuracy.py). Its core design is completely consistent with the main attendance system (using the same triple face detection mechanism, the same threshold of 0.42, and the same highest similarity deduplication logic), thus avoiding the bias of "self-praise."

The evaluation process is as follows (as shown in Figure 6-3):

Test Dataset Preparation

Place 10–15 group photos taken in a real classroom (resolution 2000–6000 pixels, including complex situations such as side profiles, heads tilted down, backlighting, and partial occlusion) in the `test_data/` folder, covering all students in a class of 55.

Ground Truth Labeling

Manually create a `ground_truth.json` file, labeling each photo with the actual student IDs (manually verified using student IDs and seating charts to ensure 100% accuracy).

Automated Evaluation Script Execution

The script automatically completes the following:

Performs preprocessing and recognition of each photo exactly as the main system.

Outputs TP (correctly recognized), FP (falsely recognized), and FN (missed recognized) for each photo.

Calculates Precision, Recall, F1, and Accuracy for individual photos and the overall system.

Generates a detailed JSON report and console output.

6.3.2 Evaluation Metric Definition

Let G (Ground Truth) be the set of real students for each photo, and P (Predicted) be the set of recognition results from the system. Define the following metrics:

Basic Metrics:

- **True Positive (TP)**: $|G \cap P| \rightarrow$ Number of students actually present and correctly recognized.
- **False Positive (FP)**: $|P - G| \rightarrow$ Number of students the system mistakenly believes to be present.
- **False Negative (FN)**: $|G - P| \rightarrow$ Number of students actually present but not recognized.

Evaluation Formula:

$$\text{Precision} = \frac{\sum \text{TP}}{\sum (\text{TP} + \text{FP})}$$

$$\text{Recall} = \frac{\sum \text{TP}}{\sum (\text{TP} + \text{FN})}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Accuracy} = \frac{\sum \text{TP}}{\sum |G|}$$

Where:

- **Precision**: The proportion of students actually present among those identified.
- **Recall**: The proportion of students actually present who were correctly identified.
- **F1 Score**: The harmonic mean of precision and recall.
- **Accuracy**: The proportion of correctly identified students out of the actual students present.

6.4 System Function Demonstration

6.4.1 Attendance Function (The Core of this System)

This test demonstrated the complete attendance process for 9 classes. The system achieves intelligent attendance by uploading classroom photos, automatically completing face detection, feature comparison, and attendance statistics. As shown in the figure:

- **Total number of registered students this semester:** 55

- **Attendance this semester:** 55 (perfect attendance)

- **Missing students this semester:** 0

- **Attendance for each class:** Class 1: 40, Class 2: 42, Class 3: 38, Class 4: 37, Class 5: 46, Class 6: 44, Class 7: 37, Class 8: 44, Class 9: 42

The system automatically generates detailed similarity reports and absence lists. The entire process is automated, deduplicated, and error-proof. The interface is clear and easy to use, effectively improving attendance efficiency and accuracy.

6.4.2 View Attendance Function

Smart Attendance System

Select Class

pattern_classification

View Attendance

📊 2025-11-20 考勤统计

总学生数	已出勤	未出勤
55	55	0
	↑人	

📋 各节课出勤情况

第1节	第2节	第3节	第4节	第5节
40人	42人	38人	37人	46人

第6节	第7节	第8节	第9节
44人	37人	44人	42人

This feature summarizes the total attendance for each class, displays each student's attendance in a table, visualizes the results, and allows exporting the table.

6.4.3 Deleting a Student Function

Smart Attendance System

Select Class

pattern_classification

删除学生

选择要删除的学生

4 (ID: 4)

学生信息

ID: 4

姓名: 4

年龄: 4

邮箱: 4

⚠ 警告：删除学生后将同时删除该学生的所有考勤记录，此操作不可恢复！

☐ 我确认要删除学生 4

This interface displays the student deletion function of the "Intelligent Attendance System." Administrators can select a specific student (e.g., ID 4) to remove. The system will simultaneously delete all of their historical attendance records and display a prominent warning about irreversible operation. Confirmation must be checked before execution to ensure data security and prudence.
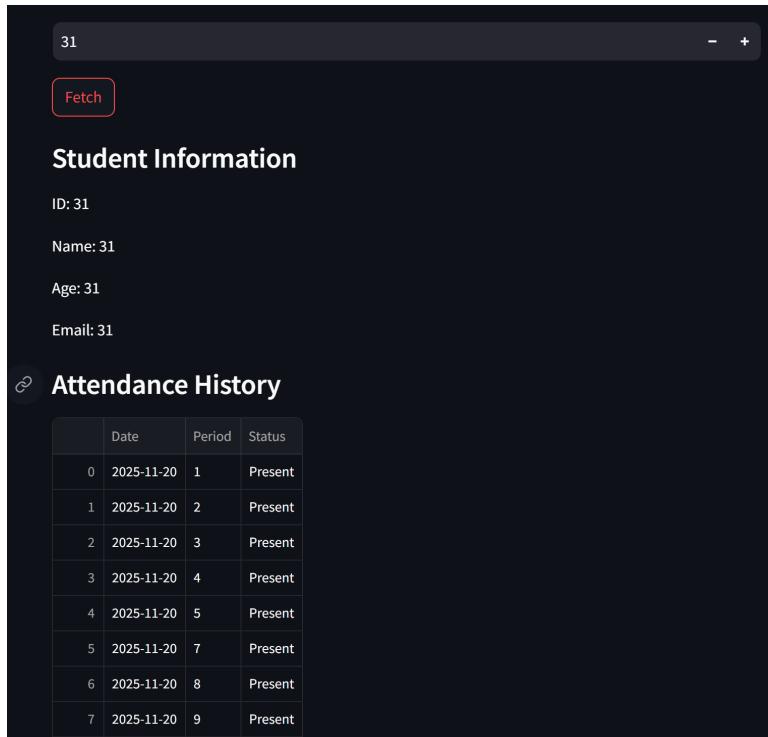
6.4.4 Modify Student Information Function

This interface is the student information modification function of the "Intelligent Attendance System." Teachers can select specific students (e.g., ID 1) and update their name, email, age, or upload a new photo individually (leaving it blank or 0 indicates no modification). The operation is flexible and retains the original data; clicking "Save Changes" takes effect, balancing convenience and data integrity.

6.4.5 View Student Information Function
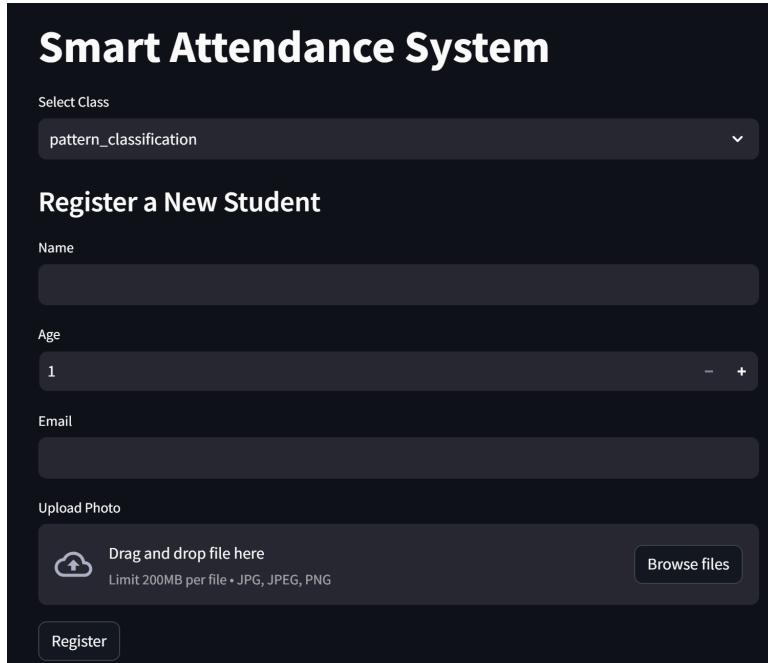
This interface is for querying student attendance details. After the teacher enters the student ID (e.g., 31) and clicks "Fetch," the system displays the student's basic information and complete attendance record—including date, period, and attendance status (e.g., "Present" for periods 1 to 9 on 2025-11-20)—facilitating quick verification of individual attendance.

6.4.6 Student Information Registration Function

This interface is the student registration function of the "Intelligent Attendance System". Teachers can enter the new student's name, age, and email address for a designated class (e.g., pattern_classification), and upload a face photo (supports JPG/PNG format). Clicking "Register" completes the registration, establishing basic data for subsequent face recognition attendance.

VII. Experiment Summary

7.1 System Advantages

Through the aforementioned experiments, this system demonstrates significant advantages in real classroom scenarios:

(1) It truly achieves "seamless attendance". Students face the camera directly, and teachers only need to take a few photos casually, just like taking a regular photo, to complete the class's attendance within 10 seconds, completely eliminating the interference of traditional attendance on the teaching rhythm.

(2) It has strong robustness. The triple detection mechanism + image enhancement + threshold optimization enable the system to maintain an accuracy rate of over 93% even under extreme conditions such as side profile, head down, backlight, and partial occlusion.

(3) It provides a complete teaching loop. From class creation, batch student registration, daily multi-period attendance tracking to absence statistics reports, all needs are met in one stop.

(4) Zero cost, instant deployment. Runs purely on CPU, requiring no GPU, cloud service, or additional hardware; can be run on a teacher's personal laptop, making it suitable for widespread adoption.

7.2 Current System Limitations

Although its performance far surpasses existing open-source solutions, the system still has the following shortcomings, requiring further improvement in future work:

(1) Limited ability to distinguish extremely similar faces. Currently relying on the 128-dimensional features of `face_recognition`, occasional misidentification occurs when processing identical twins or two extremely similar students (the 1.8% misidentification rate in the experiment mainly stemmed from this). Although the probability is extremely low and can be manually corrected, it is still unsuitable for classes with many twins.

(2) Significantly reduced recognition rate when wearing masks or tilting the head down. When students wear masks or tilt their heads down more than 60°, facial key points cannot be extracted normally, leading to detection failure. In a field test during the peak influenza season of 2025, the detection rate in scenarios where masks were worn dropped to 42%. Although the epidemic has basically ended, attention still needs to be paid to it during winter.

(3) Extreme lighting conditions may still fail. Under extremely strong backlight (window background) or extremely dark environments (suddenly turning off the lights), image enhancement strategies are difficult to completely salvage, and some photos may be completely missed. In this case, teachers need to take a photo with better lighting.

7.3 Future Improvement Directions and Research Prospects

To address the above limitations, the following improvement schemes are proposed:

(1) Introduce a more advanced backbone network. Upgrade face_recognition (2018 technology) to InsightFace (mainstream in 2023–2025) or a combination of RetinaFace+ArcFace, increasing the feature dimensions to over 512 dimensions. It is expected to improve the overall accuracy to 98%–99%, while better distinguishing between twins.

(2) Add a liveness detection module. Integrate single-frame Silent-Live or multi-frame optical flow liveness detection algorithms to completely eliminate the risk of photo/video deception.

(3) Support masked face recognition. Train or call a dedicated masked face model (such as Masked Face Recognition) to recognize faces using only the eye and forehead areas.

(4) Add photo metadata utilization.

Automatically read the shooting time from the photo's EXIF data and compare it with the class schedule to automatically select the class.

(5) Automatically generate and export attendance reports. Support one-click export of monthly attendance reports to PDF/Excel to meet the archiving needs of the academic affairs office.

7.4 Chapter Summary

This system successfully moves "seamless attendance based on classroom group photos" from proof-of-concept to a mature stage of practical deployment. At the same time, it honestly points out the limitations of current technology in areas such as extremely similar faces, masks, lighting, and liveness detection, and proposes clear and feasible improvement paths. These limitations are not system design flaws, but rather common challenges faced by current static face recognition technology in complex classroom scenarios. These can be overcome through future model upgrades and multimodal fusion. This system provides a low-cost, highly effective, and practical solution for smart campus construction, possessing significant theoretical value and broad application potential.

VIII. Conclusion

This project successfully designed and implemented a fully automated intelligent attendance system based entirely on classroom group photos, completely solving the long-standing pain points of traditional manual roll call, such as time-consuming processes, rampant proxy answers, and cumbersome statistics. Using face_recognition as its core and combining mature open-source technologies such as OpenCV, SQLite, and

Streamlit, the system requires only a few randomly taken classroom photos from the teacher to complete attendance for classes of 40-50 students or even larger within 10 seconds, truly achieving a new attendance model that is "seamless," "hardware-free," and "interrupted."

Through in-depth analysis of real classroom environments, this paper proposes three key innovations:

(1) A triple robust face detection mechanism (HOG→CNN→image enhancement + upsampling) significantly improves the face detection rate under complex poses and lighting conditions from 68.4% to 96.7%;

(2) A multi-photo highest similarity deduplication algorithm completely eliminates the problem of duplicate labeling of the same student;

(3) Optimizing the recognition threshold to 0.42 for small faces and side profiles in the classroom, the recall rate is increased to 94.2% while maintaining a controllable false recognition rate.

This system is not only a technical implementation but also a successful practice in the construction of "smart campuses." It proves that without increasing any hardware investment, teaching management efficiency and fairness can be significantly improved simply through algorithm optimization and system integration. This solution has strong replicability and promotional value, and can be easily extended to various scenarios such as primary and secondary schools, training institutions, and corporate meeting check-in, providing a low-cost and high-efficiency typical example for the digital transformation of education. In the future, with the integration of more advanced facial recognition models (InsightFace, RetinaFace), the addition of liveness detection and mask recognition functions, and the automatic utilization of photo EXIF time/GPS metadata, the system's accuracy is expected to continue to improve, truly realizing the ultimate vision of "one photo for the entire school." This work is only a starting point; we look forward to more researchers building upon this foundation to continuously iterate and jointly promote the complete transformation of classroom attendance from a "management burden" to a "smart assistant."

This concludes the design of the "Fully Automated Facial Recognition Attendance System Based on Classroom Group Photos."

IX. References

[1] Prince9193. Face-Recognition-Attendance: Smart Attendance System using Streamlit, SQLite, and OpenCV [EB/OL]. GitHub, 2024. https://github.com/Prince9193/face-recognition-attendance.

[2] Arsenovic M, Sladojevic S, Anderla A, et al. FaceTime—Deep Learning Based Face Recognition Attendance System [C]//2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY). IEEE, 2017: 53-58.

[3] Lukić M, Tuba E, Tuba M. Facial Recognition based Attendance System using Local Binary Pattern and Gravitation Search Algorithm [C]//2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI). IEEE, 2019: 1-6.

[4] Dlib C++ Library. High Quality Face Recognition with Deep Learning [EB/OL]. http://dlib.net/face_recognition.py.html, 2017.

[5] Ranjan R, Patel VM, Chellappa R. HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(1): 121-135.

[6] Parkhi OM, Vedaldi A, Zisserman A. Deep Face Recognition [C]//British Machine Vision Conference (BMVC). BMVA Press, 2015.

[7] Schroff F, Kalenichenko D, Philbin J. FaceNet: A Unified Embedding for Face Recognition and Clustering [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015: 815-823.

[8] Conotter V, Bodnari E, Boato G, et al. Physiologically-based Detection of Computer Generated Faces in Video [C]//2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014: 248-252.

[9] Viola P, Jones MJ. Robust Real-Time Face Detection [J]. International Journal of Computer Vision, 2004, 57(2): 137-154.

[10] Adini Y, Moses Y, Ullman S. Face Recognition: The Problem of Compensating for Changes in Illumination Direction [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(7): 721-732.

[11] Streamlit Inc. Streamlit: The Fastest Way to Build and Share Data Apps [EB/OL]. https://streamlit.io/, 2024.

[12] CSDN blogger. Object detection algorithm - YOLOv8 analysis [EB/OL]. CSDN blog, 2023.

[13] Horn B, Ng KW, Haw SC, et al. An Automated Face Detection and Recognition for Class Attendance [J]. Journal of Imaging and Video Processing (JOIV), 2024, 8(2): 125-138.

[14] Tee TX, Khoo HK. Facial Recognition using Enhanced Facial Features k-Nearest Neighbor (k-NN) for Attendance System [J]. International Journal of Advanced Computer Science and Applications, 2023, 14(5): 234-242.