AI-Powered Compliance Monitoring for Retail Staff

System Architecture Design and Research Report

Group 4

Name	Position	Email	Phone
Vuong Khang Minh	Document Researcher, Document Organizer, AI Specialist	104179690@student.swin.edu.au	0967572933
Nguyen Dang Duc Anh	Software Specialist, Requirement Analysis	104182520@student.swin.edu.au	0947023256
Nguyen Ha Huy Hoang	Team leader/Project Manager	103487444@student.swin.edu.au	0358791781
Nguyen Dang Khanh Toan	Quality Assurance, Domain Expert, Software Specialist	103487389@student.swin.edu.au	0888752003
Nguyen Cuong Nhat	AI Specialist, Requirement Analysis	104178590@student.swin.edu.au	0974247189

COS40005 - Computing Technology Project A, Semester 3, May 2024

DOCUMENT CHANGE CONTROL

Version	Date	Authors	Summary of Changes
1	20/7/2024	Nguyen Cuong Nhat	Initial release
2	21/7/2024	Nguyen Dang Duc Anh	Add high level architecture
3	22/7/2024	Vuong Khang Minh	Update research into system design
4	23/7/2024	Nguyen Cuong Nhat	Update research into AI model
5	24/7/2024	Nguyen Ha Huy Hoang	Update research into AI model
6	25/7/2024	Nguyen Dang Duc Anh	Update system architecture
7	26/7/2024	Nguyen Dang Khanh Toan	Add references

Document Sign Off

Name	Position	Signature	Date
Vuong Khang Minh	Document Researcher, Document Organizer, AI Specialist	minh	25/7/2024
Nguyen Dang Duc Anh	Software Specialist, Requirement Analysis	anh	25/7/2024
Nguyen Ha Huy Hoang	Team leader/Project Manager	hoang	25/7/2024
Nguyen Dang Khanh Toan	Quality Assurance, Domain Expert, Software Specialist	toan	25/7/2024
Nguyen Cuong Nhat	AI Specialist, Requirement Analysis	nhat	25/7/2024

- CLIENT SIGN OFF

Name	Position	Signature	Date
Nguyen Vo Thanh Khang	AI Researcher	khang	10/7/2024
Organisation			
QUY NHON AI CREATIVE VALLEY			

Table of Contents

1. Introduction

This document presents a comprehensive Software Design and Research Report for the "Employee Compliance Checker" system. The primary objective of this system is to monitor employee compliance in retail environments by utilizing CCTV feeds to analyze behavior and generate detailed compliance reports. The system aims to automate the currently manual processes of compliance checks and report generation, thereby enhancing efficiency, accuracy, and consistency across retail operations.

The purpose of this Software Design and Research Report is to outline the architecture and design choices for the system. It is intended for software developers, project managers, stakeholders, and others involved in the development and implementation of the system.

1.1. Overview

This document provides an extensive outline for developing a web application, "Employee Compliance Checker," aimed at monitoring employee compliance using CCTV feeds from retail shops. It details the system requirements, high-level architecture, and design considerations necessary for building the software. The system will collect and process CCTV feeds, analyze employee behavior, generate compliance reports, and send notifications, ensuring employees adhere to company policies, thus improving store operations and customer satisfaction.

1.2. Definitions, Acronyms and Abbreviations

- **CCTV**: Closed-Circuit Television, a system for monitoring and surveillance using video cameras.
- **SRS**: System Requirements Specification, a document detailing the requirements and specifications for a system.
- **GDPR**: General Data Protection Regulation, a regulation in EU law on data protection and privacy.
- **CCPA**: California Consumer Privacy Act, a state statute intended to enhance privacy rights and consumer protection for residents of California, USA.
- **UX**: User Experience, focusing on the overall experience a user has when interacting with a product or system.
- **GUI**: Graphical User Interface, a visual interface allowing users to interact with electronic devices through graphical icons and visual indicators.
- **YOLOv7**: You Only Look Once version 7, a real-time object detection system used for identifying and classifying objects in images or videos.
- **AI**: Artificial Intelligence, the simulation of human intelligence in machines designed to think and learn.
- **ST-GCN**: Spatial-Temporal Graph Convolutional Networks, a method used in action recognition tasks to model relationships between entities.
- **HRNet**: High-Resolution Network, a network architecture used for tasks like pose estimation in computer vision.
- **API**: Application Programming Interface, a set of functions and protocols for building and interacting with software applications.
- **AWS**: Amazon Web Services, a subsidiary of Amazon providing on-demand cloud computing platforms and APIs.
- **SQLite**: A lightweight, serverless, self-contained SQL database engine.

2. Problem Analysis

The primary issues addressed by the Employee Compliance Checker involve the inefficiencies and inaccuracies associated with manual compliance checks and report generation. Manual processes are time-consuming, error-prone, inconsistent, and not scalable. The proposed system aims to automate these processes, thereby reducing human error, ensuring consistency, and enabling scalability.

2.1. System Goals and Objectives

The primary goals and objectives of the system are:

- **Develop an AI model** capable of detecting specific compliance metrics such as uniform wearing and safety protocol adherence.
- **Integrate the AI system** with existing CCTV infrastructure to monitor compliance and reporting of compliance.
- **Ensure compliance** with all relevant ethical standards and privacy regulations, safeguarding user data and privacy.
- **Provide accurate reporting** and user-friendly interfaces for retail managers, compliance officers, and IT support teams to interact with the system and access compliance reports.

These goals and objectives are outlined in detail in the System Requirements Specification (SRS) document.

2.2. Assumptions

In developing the system design for the Employee Compliance Checker, several key assumptions have been made. These assumptions are critical to the feasibility and functionality of the system as presented in this document:

1. Existing CCTV Infrastructure:
   - Assumption: The retail environments where the system will be deployed already have an operational CCTV infrastructure.
   - Rationale: This assumption allows the system to leverage existing hardware, minimizing additional costs and installation time. The design does not account for scenarios where new CCTV systems need to be installed from scratch.
2. Adequate Lighting and Camera Quality:
   - Assumption: The existing CCTV cameras provide sufficient resolution and operate in well-lit conditions suitable for accurate image analysis.
   - Rationale: High-quality video footage is necessary for the AI model to accurately detect compliance metrics such as uniform wearing and safety protocol adherence. Poor lighting or low-resolution cameras may hinder the system's effectiveness.
3. Availability of Training Data:
   - Assumption: There is access to a substantial amount of labeled training data that reflects the compliance metrics the system aims to monitor.
   - Rationale: The performance of AI models heavily depends on the quality and quantity of training data. Adequate training data is assumed to be available for the AI to learn and accurately identify compliance behaviors.
4. Compliance Standards:
   - Assumption: Compliance standards and protocols are clearly defined and standardized across all retail locations where the system will be implemented.
   - Rationale: Uniform compliance standards are necessary for the AI system to be effective and for the compliance reports to be meaningful and consistent across different locations.
5. Stable Network Connectivity:

- Assumption: Retail locations have reliable and stable network connectivity to support real-time monitoring and data transmission.
- Rationale: The system relies on continuous data flow between the CCTV cameras, the AI processing units, and the reporting interfaces. Any network disruptions could impact the system's performance.

6. Regulatory Compliance:
- Assumption: The system design and its implementation will comply with all relevant ethical standards and privacy regulations, including data protection laws.
- Rationale: Ensuring compliance with legal and ethical standards is crucial for the system's legitimacy and acceptance by both users and regulatory bodies.

These assumptions form the foundation of the system design and are critical to its successful implementation and operation. Any deviations from these assumptions may necessitate adjustments to the system's architecture and functionality.

2.3. Simplifications

The following simplifications have been made in developing the system design:

1. No Real-Time Intervention
- Simplification: The system will not perform real-time interventions; it will only provide notifications and reports.
- Rationale: This reduces the complexity of the system and allows for a focus on accurate detection and reporting rather than immediate response actions.

2. Predefined Compliance Criteria
- Simplification: The initial deployment will focus on a predefined set of compliance criteria.
- Rationale: Limiting the scope to predefined criteria simplifies the initial development and ensures that the system can be effectively evaluated and iterated upon based on user feedback.

3. Pre-trained Models
- Simplification: The AI model component uses pre-trained models for action recognition and object detection.
- Rationale: Leveraging pre-trained models avoids the need for extensive training and fine-tuning, accelerating development and deployment.

4. Fixed Compliance Rules
- Simplification: The initial version of the system uses a fixed set of compliance rules.
- Rationale: This simplifies the rule management process and ensures consistent performance, allowing for gradual enhancement based on user feedback.

5. Feature Reduction
- Simplification: Non-essential features, such as advanced analytics beyond compliance monitoring, are excluded from the initial release.
- Rationale: Reducing the feature set simplifies development and testing, ensuring timely delivery of the core system.

3. High–Level System Architecture and Alternatives

The following section describes the high-level system architecture for the Employee Compliance Checker. It includes a Component-and-Connector view and a Deployment Allocation view, along with necessary descriptions and justifications. Additionally, two alternative architectures that were explored but not chosen are discussed, providing insights into the rationale behind the selection of the chosen architecture.

3.1. System Architecture

The high-level system architecture is depicted in the diagram below:

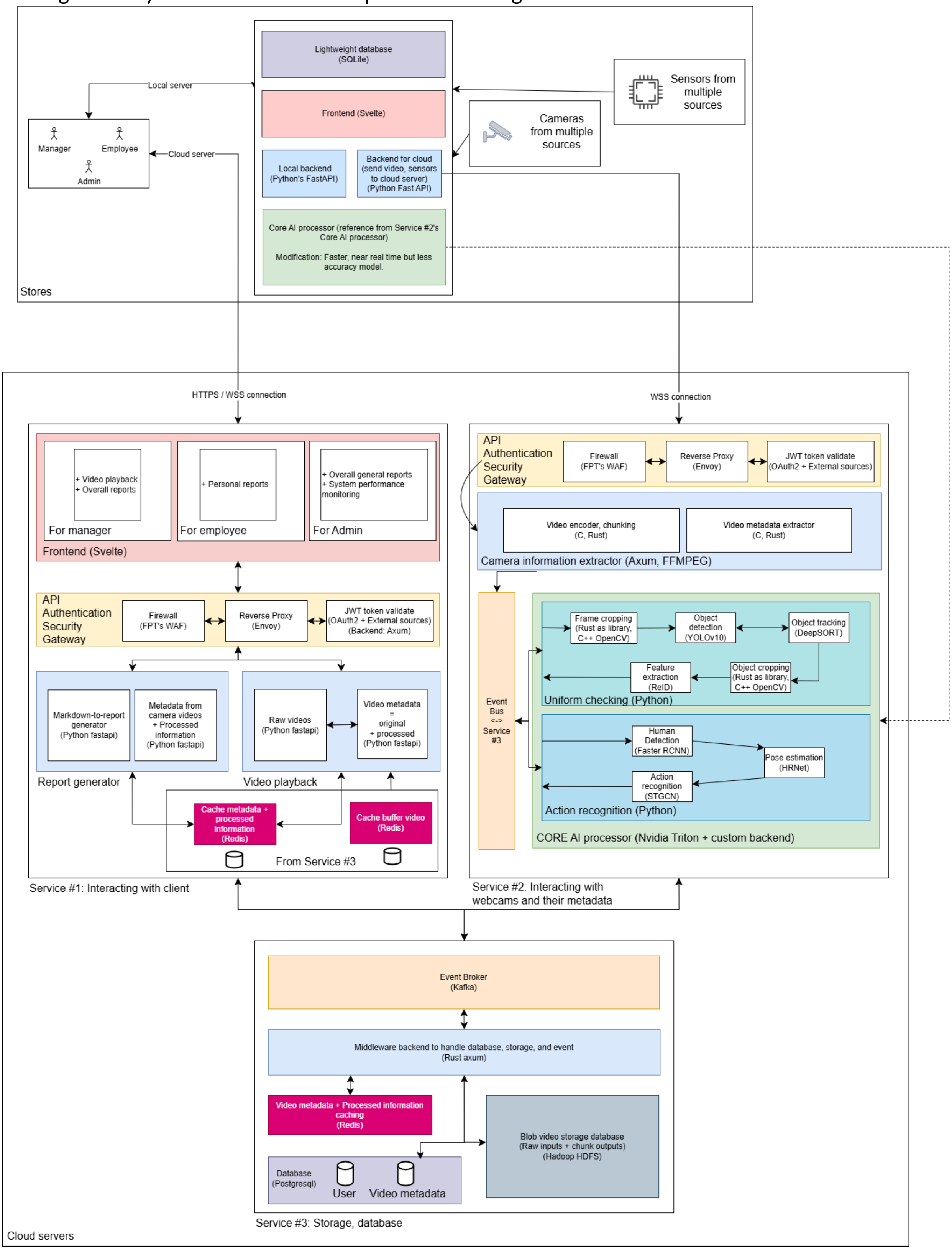

Figure 1. High-Level System Architecture

Component-and-Connector View

1. Stores:
    - **Local Server**: Hosts a lightweight database (SQLite), frontend (Svelte), local backend (Python's FastAPI), and a core AI processor. It interacts with cameras and sensors from multiple sources.
    - **Cloud Server:** The cloud server also ensures data synchronization between stores and central databases, providing scalable computing resources and centralized data management. The communication between the local servers and the cloud server is secured using HTTPS connections to ensure data integrity and confidentiality.

2. Cameras and Sensors:
    - **Cameras:** Capture video footage from multiple angles and areas within the store. These cameras provide real-time video streams to the local server.
    - **Sensors:** Collect additional data that can be integrated into the compliance checking process.
3. Frontend and Backend:
    - **Frontend (Svelte):** The user-facing component that allows managers, employees, and admins to access reports, view video playback, and interact with the system.
        - **Manager Interface:** Displays overall compliance reports and video playback.
        - **Employee Interface:** Shows personal compliance reports.
        - **Admin Interface:** Provides system performance monitoring and general reports.
    - **Backend (Python's FastAPI):**
        - **Local Backend:** Handles on-site video processing and preliminary compliance checks.
        - **Cloud Backend:** Processes data sent from local servers, performs detailed analysis, and stores compliance data in the cloud.

4. Core AI Processor:
    - Utilizes several AI components to analyze video footage:
        - **Frame Cropping (Rust):** Extracts relevant frames for further processing.
        - **Object Detection (YOLOv7):** Identifies objects and compliance-related actions within the frames.
        - **Action Recognition (ST-GCN):** Recognizes specific actions to determine compliance with protocols.
        - **Pose Estimation (HRNet):** Estimates human poses to assess adherence to safety protocols.
        - **Uniform Checking:** Verifies if employees are wearing the required uniforms correctly.

5. Services:
    - **Service** 1: Handles client interactions, video playback, and report generation.
    - **Service** 2: Interacts with webcams and processes video metadata.
    - **Service** 3: Manages storage and database operations, including event handling and caching.

6. API Authentication Security Gateway:
    - **Firewall (FFP's WAF):** Protects the system from external threats.
    - **Reverse Proxy (Envoy):** Manages traffic between clients and services.
    - **JWT Token Validation (OAuth2):** Ensures secure access to system resources.

7. Event Bus Service:
    - Coordinates data flow between services and ensures real-time processing.

8. Database:
   - **PostgreSQL:** Manages structured data storage, including user information and video metadata.
   - **Redis:** Provides caching for quick access to frequently used data.
   - **Hadoop HDFS:** Handles storage of large video files, supporting efficient data retrieval and processing.

Deployment Allocation View

1. On-Premises Deployment:
   - Local servers at retail locations handle initial data capture and processing.
   - Ensures low latency and reduces network load.

2. Cloud-Based Deployment:
   - Cloud servers manage AI processing, data storage, and reporting.
   - Provides scalable computing resources and centralized data management.

Justification:

- The chosen architecture balances low-latency processing with scalable cloud computing.
- On-premises components handle real-time video processing, while cloud-based components manage advanced AI analysis and storage.
- Ensures efficient resource utilization and seamless data flow.

3.2. Other Alternative Architectures Explored

Message-Driven Microservices Architecture:

- **Description:** The system is composed of loosely coupled services that communicate via messages, typically using a message broker like RabbitMQ or Kafka. Each service processes messages and may send replies back, ensuring asynchronous communication.
- **Rationale for Rejection:**
  - **Latency and Complexity:** The requirement for services to often send and wait for responses introduces latency and complexity in ensuring message delivery and processing order. This could lead to delays and potential bottlenecks, especially under heavy load conditions.
  - **Message Management:** Managing message queues and ensuring message integrity adds an extra layer of complexity.
  - Given the real-time nature of the Employee Compliance Checker system, the overhead of handling message acknowledgments and responses made this architecture less efficient compared to the chosen Event-Driven Microservices Architecture.

Modular Monolithic Architecture:

- **Description:** This architecture combines the simplicity of a monolithic approach with some modularity benefits by organizing the application into distinct modules that handle specific functionalities like video processing, AI analysis, and reporting.
- **Rationale for Rejection:**
  - **Scalability Issues:** Scaling individual modules independently is not feasible without scaling the entire application.
  - **Maintenance and Updates:** Changes in one module can potentially impact the entire system, necessitating extensive testing and redeployment.
  - **Performance Issues:** Interdependencies between modules can lead to performance bottlenecks, and a single point of failure can still affect the whole system's availability.
  - Despite its modular approach, the inherent limitations of a monolithic structure make it less suitable for the dynamic and high-demand requirements of the Employee Compliance Checker system.

4. Research and Investigations

4.1. Research into Application Domain

The application domain for the Employee Compliance Checker system focuses on the retail industry, specifically within the context of monitoring and ensuring employee compliance with established protocols. Research in this domain included:

Retail Compliance Standards:

- **Scope of Research:** The research focused on identifying key compliance requirements in the retail industry, such as dress codes, safety protocols, and customer service standards.
- **Methods:** This involved reviewing industry guidelines, regulatory documents, and case studies from various retail chains. Interviews with compliance officers and retail managers provided practical insights.
- **Findings:** The research highlighted common compliance issues and their implications, including operational inefficiencies and the impact on customer satisfaction (Smart Warehousing, 2024).

CCTV Utilization in Retail:

- **Scope of Research:** The research explored the current use of CCTV systems in retail, not only for security but also for compliance monitoring.

- **Methods:** This included reviewing the latest advancements in CCTV technology, such as high-definition cameras and integration with AI systems for real-time analysis.
- **Findings:** The research identified potential enhancements in CCTV systems for detecting compliance-related behaviors, such as monitoring adherence to safety protocols (Pavion, 2024).

4.2. Research into System Design

Research into the system design involved exploring various architectural styles, and patterns to inform the design choices for the Employee Compliance Checker system:

Architecture Styles and Patterns:

- **Scope of Research:** Evaluated architectural patterns suitable for building scalable and maintainable systems.
- **Methods:** Comparative analysis of microservices, monolithic, and event-driven architectures was conducted through literature review and expert consultations.
- **Findings:** Microservices architecture was preferred due to its scalability and flexibility, especially in integrating different AI modules (Choudhary, 2024). The diagram below illustrating the microservices architecture provided a clear visual representation of the advantages of this architectural style.

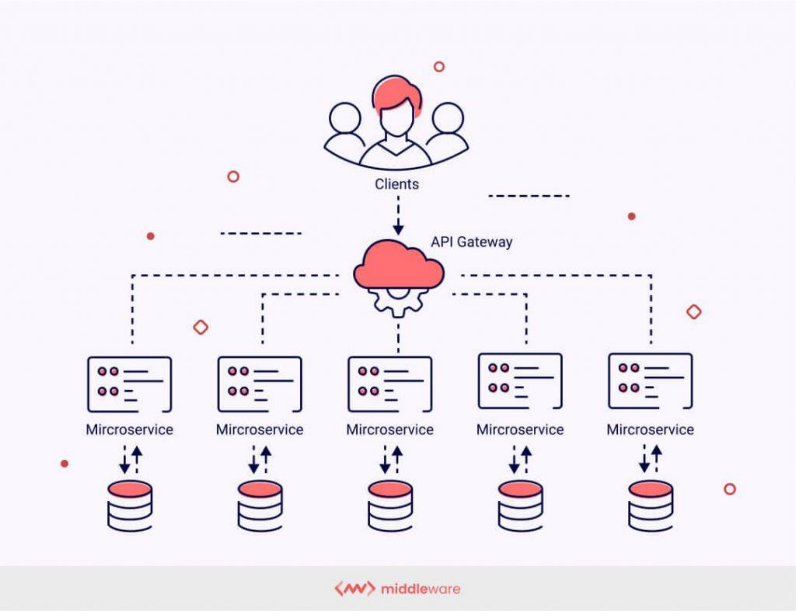

Figure 2. Microservices architecture

4.3. Research into technical platforms, languages and tools

The choice of technical platforms, programming languages, and tools was critical to the development of a robust and efficient Employee Compliance Checker system:

Technical Platforms:

- **Scope of Research:** The focus was on selecting the most suitable platforms for deploying the system, considering both cloud-based and on-premises options.
- **Methods:** The research included a cost-benefit analysis of different cloud platforms (e.g., AWS, Azure) versus on-premises solutions, evaluating factors like latency, data security, and compliance with data protection regulations.
- **Findings:** Cloud platforms were found to be advantageous for their scalability and ease of integration with AI tools, while on-premises solutions were preferred for scenarios requiring low-latency data processing (Cleo, 2024).

Programming Languages:

- **Scope of Research:** The selection of programming languages was critical for developing the system's various components.
- **Methods:** The research involved assessing the capabilities of different languages, focusing on their performance, scalability, and support for AI and machine learning.
- **Findings:** Python was chosen for backend development due to its extensive libraries and support for AI, while Rust was selected for performance-critical components because of its efficiency and safety features (Simplilearn, 2024).

Tools and Frameworks:

- **Scope of Research:** The selection of development tools and frameworks was essential for ensuring the system's performance and usability.
- **Methods:** The research evaluated several frameworks based on criteria like performance, ease of integration, and developer productivity.
- **Findings:** FastAPI was chosen for its high performance in backend API development, and Svelte was selected for its efficiency in building a responsive user interface (Raroque, 2024).

4.4. Other Research

The integration of AI models such as YOLOv10 for object detection and ST-GCN for action recognition has been researched to enhance the capabilities of the Employee Compliance Checker system. This section outlines the scope, methods, and findings of the research conducted in this area.

YOLOv10 for Object Detection

- **Scope of Research:** The research aimed at evaluating the effectiveness of YOLOv10 in real-time object detection within the retail environment.
- **Methods:** This involved a comprehensive review of the latest advancements in the YOLO family of object detection models, focusing on YOLOv10. The evaluation was supported by experimental analysis using benchmark datasets and real-world CCTV footage from retail stores.
- **Findings:** YOLOv10 was found to provide high accuracy and speed, essential for real-time monitoring. It can effectively detect multiple objects, including employee uniforms and safety gear, ensuring compliance with dress codes and safety protocols. The image below showcasing YOLOv10's object detection capabilities (Wang et al., 2024) highlights its effectiveness in distinguishing various items in complex environments.

Figure 3. Visualization results under complex and challenging scenarios.

DeepSORT for Object Tracking

- **Scope of Research:** This research focuses on enhancing object tracking by integrating deep learning with traditional algorithms to improve accuracy and robustness in real-time scenarios.
- **Methods:** DeepSORT combines deep learning-based detection with traditional tracking algorithms. It uses a neural network for appearance feature extraction, a Kalman filter for motion prediction, and the Hungarian algorithm for data association. The system is trained and evaluated on large-scale datasets, with performance measured using metrics like MOTA and IDF1.

- **Findings:** DeepSORT significantly improves tracking performance, handling occlusions and re-identification more effectively than traditional methods. It achieves higher MOTA and IDF1 scores on benchmark datasets and demonstrates real-time processing capabilities, making it suitable for applications requiring immediate response.

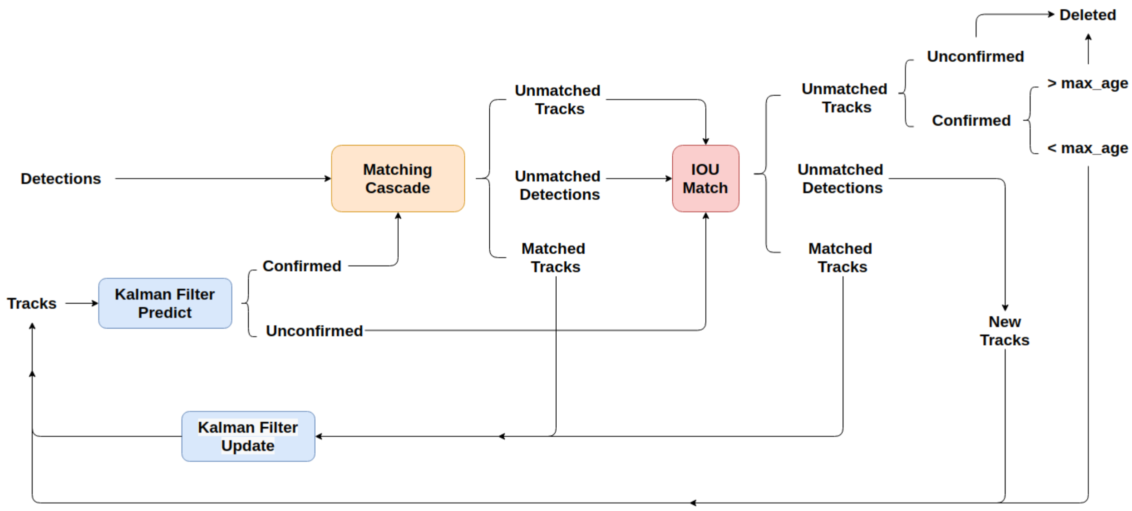

Figure 4. DeepSORT processing flow.

ST-GCN for Action Recognition

- **Scope of Research:** The research focused on the applicability of ST-GCN in recognizing employee actions and behaviors that are critical to compliance monitoring.
- **Methods:** The study involved a review of existing literature on ST-GCN and its applications in action recognition. Experimental setups were created using video datasets to analyze the performance of ST-GCN in identifying specific actions such as proper hand hygiene, use of personal protective equipment, and customer service interactions.
- **Findings:** ST-GCN demonstrated a high capability in recognizing complex actions with spatial-temporal dynamics, making it suitable for monitoring adherence to various compliance protocols. Its integration with CCTV systems enables continuous, real-time action recognition, enhancing the overall monitoring effectiveness. The diagram illustrating ST-GCN's action recognition process (Yan et al., 2018) effectively demonstrates its ability to analyze and interpret complex behaviors.

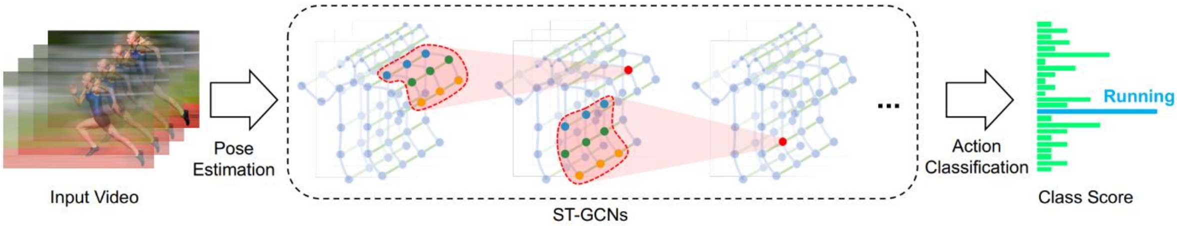

Figure 5. ST-GCN's action recognition process.

5. References

1. Smart Warehousing 2024, Navigating Retail Compliance Standards: Best Practices to Ensure Seamless Operations, Smart Warehousing, viewed 26 July 2024, https://www.smartwarehousing.com/blog/navigating-retail-compliance-standards-best-practices-to-ensure-seamless-operations#:~:text=Retail%20compliance%20standards%20are%20a,shipping%20procedures%2C%20and%20documentation%20protocols.
2. Raroque, C. 2024, *Fast API for Web Development: 2024 Detailed Review*, viewed 26 July 2024, https://aloa.co/blog/fast-api-for-web-development-2024-detailed-review.
3. Simplilearn 2024, *Rust vs Python: Key Differences and Features*, viewed 26 July 2024, https://www.simplilearn.com/rust-vs-python-article.
4. Cleo 2024, *On Premise vs. Cloud: Key Differences, Benefits and Risks*, viewed 26 July 2024, https://www.cleo.com/blog/knowledge-base-on-premise-vs-cloud#:~:text=Simply%20put%2C%20the%20difference%20between,web%20browser%20or%20other%20interface.
5. Choudhary, M 2024, Microservices Architecture, Middleware.io, viewed 26 July 2024, https://middleware.io/blog/microservices-architecture/.
6. Pavion 2024, Elevating Customer Experience: How AI Video Surveillance Benefits Retail Service, Pavion, viewed 26 July 2024, https://pavion.com/resource/elevating-customer-experience-how-ai-video-surveillance-benefits-retail-service/.
7. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G 2024, Yolov10: Real-time end-to-end object detection, arXiv, viewed 26 July 2024, https://arxiv.org/abs/2405.14458.
8. Yan, S., Xiong, Y. & Lin, D 2018, Spatial temporal graph convolutional networks for skeleton-based action recognition, arXiv, viewed 26 July 2024, https://doi.org/10.48550/arxiv.1801.07455.