SOFTWARE ENGINEERING PROJECT

SpotOn (Project Proposal)

BY

Krittin Setdhavanich
Yanatchara Jeraja

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING
KASETSART UNIVERSITY

Academic Year 2025

SpotOn (Project Proposal)

BY

Krittin Setdhavanich
Yanatchara Jeraja

This Project Submitted in Partial Fulfillment of the Requirement for Bachelor Degree of Engineering (Software Engineering)
Department of Computer Engineering, Faculty of Engineering KASERTSART UNIVERSITY
Academic Year 2025

Approved By:

Advisor . . . . . **Date** . . . . . .
(Asst. Prof. Dr. Supaporn Erjongmanee)

Co-Advisor . . . . . **Date** . . . . . .
(Dr. Chaiwat Klampol)

Head of Department . . . . . **Date** . . . . . .
(Assoc. Prof. Dr. Punpiti Piamsa-nga)

Abstract

In modern surveillance environments, tracking individuals across multiple camera views presents significant challenges for security personnel, leading to inefficiencies and potential security gaps. The motivation behind this project is to reduce the cognitive burden on operators who must manually monitor numerous video feeds simultaneously while maintaining situational awareness. The SpotOn addresses the fragmentation of surveillance data across distributed camera networks, as this information typically exists in isolated feeds without meaningful integration. While camera networks may capture comprehensive coverage of physical spaces, without intelligent automation to connect these perspectives, security teams struggle with rapid identification and tracking of persons of interest. Our system proposes a solution through an AI-powered approach that seamlessly identifies and tracks individuals across camera transitions and visualizes movement patterns, with optional advanced query capabilities for enhanced usability. The effectiveness of our solution is evaluated through real-world deployment scenarios under varying environmental conditions. We expect our project to significantly enhance operational efficiency for security personnel while providing more timely and accurate situational awareness. We believe that intelligent automation of cross-camera tracking will transform how organizations approach surveillance and security operations.

Keywords*—Multi-Camera Tracking, Person Re-Identification, Computer Vision, Situational Awareness, Security Automation*

Acknowledgement

We would like to acknowledge the support of Department of Computer Engineering, Kasetsart University for providing us all software development and technical knowledge and also the opportunity to prepare the project.

We would like to express our deepest gratitude to our advisor, Asst. Prof. Dr. Supaporn Erjongmanee for her mentorship throughout this project.

We would like to thank all of our faculty professors and instructors who always gave us valuable knowledge, techniques and every important details of Software and Knowledge Engineering.

Krittin Setdhavanich
Yanatchara Jeraja

Table of Contents

List of Tables

Page

List of Figures

Chapter 1

Introduction

1.1 Background

In large campus and factory environments, tracking people across multiple cameras is crucial for safety, security, and operational efficiency. Security teams need to quickly locate individuals involved in incidents like theft or unauthorized access. Factory managers require clear understanding of worker movements to ensure safety protocol compliance. Emergency responders need to pinpoint locations during evacuations. Without effective multi-camera tracking, blind spots and identification failures lead to delayed responses and compromised safety [1]. When individuals transition between camera views, maintaining their identity becomes challenging. Existing systems often fail during these handoffs, creating fragmented surveillance coverage. For example, a student moving from an outdoor courtyard to a building lobby might be lost during transition, making it difficult to determine if they later entered a restricted area. This problem is particularly significant in sprawling campuses with numerous buildings or complex factory layouts where cameras operate independently. Environmental factors further complicate tracking. Lighting differences between bright outdoor areas and dim hallways alter appearances significantly. Crowded periods during class changes or shift rotations create frequent occlusions. Cameras at varying heights and angles provide drastically different perspectives of the same person, hindering accurate matching across views - a bag visible from the side might be completely hidden from the front.

This is a **cross-disciplinary project** that actively **aligns users, engineering, and security policy**. Our project aims to develop a robust

tracking system called *SpotOn* that consistently identifies and tracks individuals across camera networks, providing uninterrupted coverage in campus and factory settings. The project methodology relies on designing solutions systematically through a **hypothesis** $\rightarrow$ **test** $\rightarrow$ **iterate** cycle. By solving the identity maintenance challenge across camera transitions, our solution will enhance security monitoring, expedite emergency responses, and provide valuable data for facility management and resource optimization.

1.2 Problem Statement

The problem statement for the SpotOn addresses the fundamental challenge of maintaining consistent identification of individuals as they move through spaces monitored by multiple cameras. In various environments from campuses and factories to public spaces and transportation hubs, the ability to track individuals across camera transitions is critical yet remains largely unsolved. When a person moves from one camera's view to another, current systems frequently lose their identity trail, creating fragmented tracking data that fails during situations requiring continuous monitoring. This tracking discontinuity affects numerous applications—from security monitoring and emergency response to space utilization analysis and research on movement patterns. The consequences include critical delays during time-sensitive situations, incomplete data for operational decision-making, and increased workload for monitoring personnel who must manually reconnect identity fragments across camera boundaries. Technically, this challenge manifests in three interconnected problems. First, the multi-view detection problem requires accurately identifying and locating individuals within each camera feed despite visual challenges like occlusions, distance variations, and diverse human appearances across different environments. Second, the cross-camera identity preservation problem involves maintaining consistent person identification during transitions between non-overlapping camera views, where individuals may appear drastically different due to perspective shifts, lighting changes, and temporal gaps. Third, the unified spatial comprehension

problem necessitates integrating observations from distributed cameras into a coherent coordinate framework that enables operators to visualize continuous trajectories and understand movement patterns throughout the entire monitored environment. Current systems address these problems in isolation rather than as an integrated challenge, resulting in fragmented surveillance coverage that fails during critical monitoring scenarios.

The design of SpotOn is explicitly driven by this identity continuity problem, with all core components, workflows, and visualizations structured to minimize tracking fragmentation and reduce manual cognitive effort for operators.

1.3 Solution Overview

The SpotOn system is conceptualized to directly address the limitations identified in current multi-camera surveillance workflows. Rather than treating camera feeds as independent data sources, the system focuses on maintaining consistent identity information as individuals move through monitored environments. To achieve this, SpotOn employs an integrated computer vision platform composed of three core technical components:

1.3.1 Prominent Features

1. **Multi-View Person Detection:** A system that identifies and locates all individuals within each camera feed, creating bounding boxes around people in the video frames. This detection works independently on each camera, handling challenges like partial occlusions, varying lighting conditions, and environmental changes without attempting to match identities across cameras.

2. **Cross-Camera Re-Identification and Tracking:** A specialized system that matches and tracks individuals across different camera views. When someone exits one camera's view and appears in another camera, this system maintains

their identity by analyzing visual features like clothing, size, and walking patterns. The tracking component continuously updates each person's position and trajectory, preserving their identity across camera transitions despite changes in perspective, lighting, and viewing angle.

3. **Unified Spatial Mapping:** A mapping system that transforms detections from distributed cameras into a coherent coordinate system. This integration enables continuous trajectory visualization as individuals move through the monitored environment across multiple camera views, providing operators with a comprehensive spatial understanding of movement patterns throughout the facility.

1.3.2 Optional Features

1. **Agentic LLM Interface with Tool-Calling:**
   - The LLM utilizes RAG (Retrieval-Augmented Generation) to fetch real-time tracked person metadata and uses tool-calling to autonomously trigger camera switches or Re-ID queries based on user prompts.
   - Implemented LLMOps practices, including prompt guardrails to prevent system hallucinations, and integrated observability/monitoring for the LLM agent.
   - The backend uses dynamic routing and model selection to choose between lightweight and heavy LLM models based on the complexity of the user's query.
2. **Multi-Person Tracking:** Capability to simultaneously maintain identity and location data for multiple individuals across the camera network, preserving each person's unique identity despite occlusions, varying viewing angles, or extended periods where individuals are not visible in any camera.
3. **Movement Pattern Analysis:** AI-powered analysis to identify typical movement patterns and provide insights for

space utilization, traffic management, and anomaly detection in pedestrian flows across the monitored environment.

4. **Configurable Model Parameters:** A settings interface enabling users to fine-tune system performance by adjusting core parameters, including detection confidence, processing framerate, bounding box appearance, and Intersection over Union (IOU) threshold for object matching. This allows adaptation to specific environmental conditions and computational resources.

5. **Environmental Analytics Dashboard:** Real-time metrics including occupancy density maps, path trajectory analysis, and movement pattern heatmaps tailored for facility management, safety monitoring, and resource optimization.

1.3.3 Stretch Goals

1. **Voice Command Integration:** Allow personnel to interact with the system using natural voice commands (e.g., "Track the person in the blue jacket heading toward Building B") for hands-free operation during critical situations.

2. **Anomaly Detection:** Implement AI-powered anomaly detection to automatically identify unusual behavior patterns (such as suspicious movements, crowd formations, or potential security incidents) across the monitored environment.

3. **Privacy-Preserving Tracking:** Implement anonymization techniques that maintain tracking capabilities while protecting individual privacy through face blurring or feature abstraction in accordance with surveillance regulations and privacy laws.

4. **Automatic Environment Mapping:** System that creates spatial representations based on observed movement patterns, automatically identifying walkways, intersections,

and common routes without requiring manual mapping of the environment.

While the core techniques employed in SpotOn, such as person detection, tracking, and re-identification, are established within the computer vision domain, the contribution of this project lies in how these components are re-conceptualized as a unified, investigation-oriented surveillance system. Rather than addressing detection, tracking, or re-identification in isolation, SpotOn treats identity continuity across distributed cameras as a system-level problem involving spatial context, temporal gaps, and user interpretability. This problem-driven design directly addresses operational challenges faced by security personnel and facility managers, such as fragmented camera views, manual identity reconciliation, and high cognitive load during investigations. By prioritizing practical identity persistence and spatial coherence, the system is well aligned with real-world campus and factory surveillance workflows.

Beyond system integration, the design of SpotOn reflects deliberate AI model selection decisions driven by real-world deployment considerations. Rather than assuming a fixed re-identification solution, multiple Re-ID approaches were evaluated, and the final model was selected based on its robustness under cross-camera variations such as viewpoint changes, lighting differences, and appearance ambiguity. This careful selection allows re-identification to function as a reliable, conditional identity confirmation mechanism within the broader tracking pipeline, ensuring that AI components are not only technically sound but also operationally appropriate for real surveillance environments.

Importantly, this represents an architectural shift from conventional appearance-first pipelines toward a spatially grounded, identity-centric design, which meaningfully alters how multi-camera tracking systems are structured and operated in practice.

1.4 Target User

The SpotOn enables professionals in security, facility management, and emergency response to maintain consistent identification of individ-

uals across multiple camera views in complex environments, enhancing safety, optimizing operations, and improving response times. The primary users are personnel responsible for monitoring distributed camera networks across large spaces.

Each identified user role is directly mapped to specific system capabilities and use cases, ensuring that functional requirements reflect real operational needs rather than generic surveillance features.

Security Officer

Preferences	Pursue persons of interest (POIs) seamlessly, detect breaches.
Pain Points	Broken suspect trails during handoffs, slow incident reaction.
Goals	Continuous POI monitoring, faster threat neutralization, secure evidence.

Facility Manager

Preferences	Analyze movement flow, measure area usage accurately.
Pain Points	Incomplete traffic data, bottlenecks unidentified.
Goals	Optimize spatial layout, understand usage trends, enhance workflow.

Emergency Coordinator

Preferences	Oversee evacuations clearly, assess crowd levels instantly.
Pain Points	Gaps in crisis zone visibility, delayed finding of individuals.
Goals	Guide safe evacuations, speed up assistance delivery, maintain situational control.

Table 1.1: Personas from User Analysis for SpotOn

1.5 Benefit

The SpotOn delivers benefits directly addressing the three key challenges in multi-camera tracking. The multi-view person detection with environmental adaptation ensures reliable tracking despite lighting changes, occlusions, and perspective differences. The cross-camera re-identification solves the identity preservation problem by maintaining continuous tracking between cameras, eliminating fragmentation when individuals move through non-overlapping camera networks. The unified spatial mapping creates a comprehensive environmental understanding by integrating all camera feeds into a single coordinate system, allowing operators to visualize movement patterns spanning multiple buildings and zones. These capabilities translate to 70% reduced manual tracking workload, faster incident response times, more effective resource allocation, and comprehensive situational awareness across the entire monitored environment.

1.6 Timeline and Competency-Based Learning Pathway

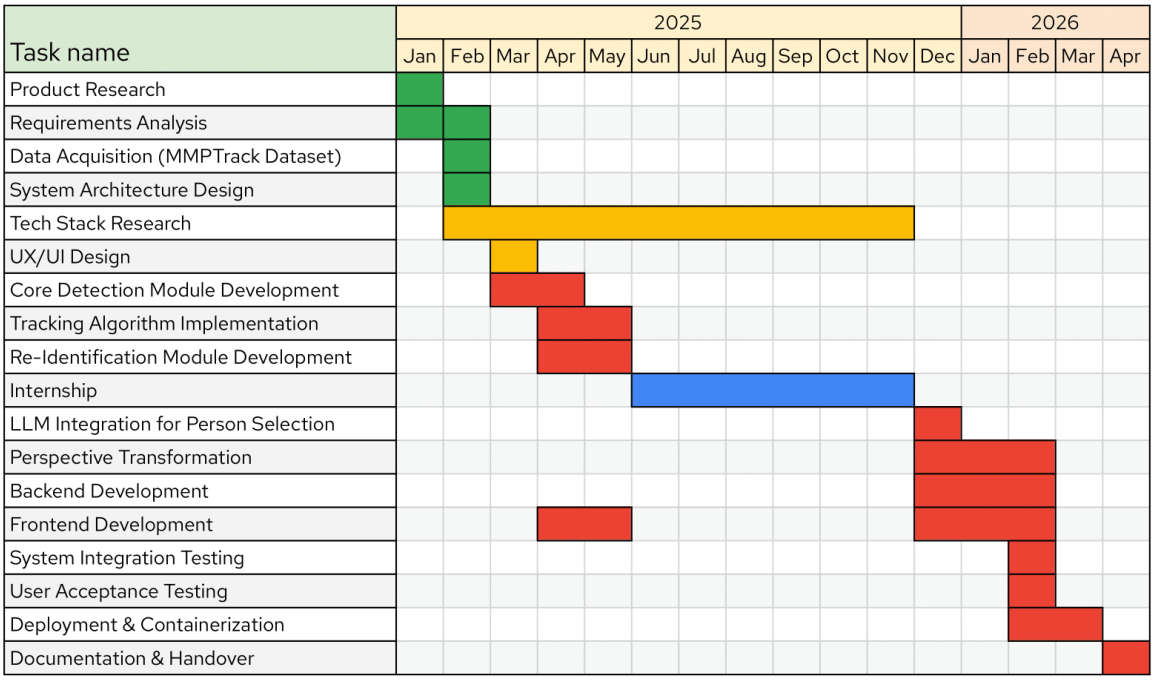

Figure 1.1: Timeline for SpotOn

As shown in figure 1.1, it represents the timeline of the project. The project began with product research in January 2025, followed by requirements analysis spanning January to February 2025. Data acquisition using the MMPTrack Dataset and system architecture design were completed in February 2025.

Technical stack research extended from February to November 2025, while UX/UI design was completed in March 2025. Core detection module development occurred in March-April 2025, with tracking algorithm implementation and re-identification module development taking place from April to May 2025.

An internship period ran from June to November 2025. LLM integration for person selection was completed in December 2025, while perspective transformation, backend development, and frontend development all occurred from December 2025 to February 2026.

System integration testing and user acceptance testing were conducted in February 2026. Deployment and containerization followed from February to March 2026, with final documentation and handover scheduled for completion in April 2026.

1.6.1 Personal Development Plan

The team has designed a **competency-based learning pathway** linked to this project, which serves as a core portfolio output. To validate our skills in Computer Vision, we will run **small experiments and mini-projects** comparing different Re-ID toolboxes (like FastReID vs. YoloReIDNet). We will use **feedback and metrics (such as rubric scores and completion rates)** to continuously adapt our learning, demonstrating **fast learning and skill transfer** from academic theory to applied surveillance contexts.

1.7 Terminology

- ***Object Detection***—a computer vision technique that identifies and locates objects within digital images or video frames, serving as the foundation for tracking individuals in each camera view.

- ***Re-Identification (Re-ID)***—the process of matching individuals across different camera views by comparing appearance features, enabling consistent identity preservation as people move between cameras.
- ***Multi-Target Multi-Camera Tracking (MTMCT)***—the task of tracking multiple individuals across a network of cameras with non-overlapping fields of view, maintaining identity consistency despite spatial and temporal gaps.
- ***Feature Extraction***—the process of identifying distinctive visual characteristics from a person's image that remain consistent despite environmental variations, critical for successful re-identification.
- ***Spatial Mapping***—a technique that converts observations from multiple camera views into a unified coordinate system, enabling continuous trajectory visualization across the entire monitored environment.
- ***Occlusion Handling***—methods to maintain tracking when individuals are partially or temporarily hidden by objects, structures, or other people in the camera view.

Chapter 2

Literature Review and Related Work

2.1 Competitor Analysis

	SpotOn	Hikvision	YOLORe-IDNet	JARVIS-MoCap
Primary Focus	Campus tracking with persistent identity	Single-camera surveillance with minimal cross-camera capability	Real-time detection without integrated systems	Research motion capture in controlled settings
Core Technology	Multi-model fusion with spatial mapping	Proprietary algorithms optimized for single views	YOLO detection with separate Re-ID network	Markerless pose estimation with synced cameras
Multi-Camera Detection	Independent detection robust to occlusions	Strong within-view detection, poor across views	No multi-camera management system	Requires calibrated camera arrays
Tracking and Re-ID	Identity across non-overlapping views	Excellent within single views, fails across views	Matches appearance without trajectories	Tracks pose, not identity
Spatial Representation	Unified coordinate system visualization	Limited to individual camera views	No spatial mapping functionality	Limited to pre-calibrated environments
System Accessibility	Web interface with easy deployment	Requires dedicated hardware	Command-line interface for experts	Research software with complex setup
Cross-Platform Support	Any device with browser access	Specific hardware and OS requirements	Platform-dependent implementation	Desktop with specific dependencies

Table 2.1: Competitor Analysis of SpotOn

In order to acknowledge the existing technology with similar functionalities, a competitor analysis is conducted. There are various systems

in the market that can track individuals, we have selected 3 of which most closely resemble our project:

- ***Hikvision AI Camera Systems***—a commercial leader in the Thailand AI camera market offering excellent single-camera tracking but limited cross-camera capabilities.
- ***YOLORe-IDNet***—a research implementation combining YOLO object detection with re-identification networks for real-time tracking without prior knowledge.
- ***JARVIS-MoCap***—an open-source markerless 3D motion capture system designed for research environments requiring precise tracking.

As shown in Table 2.1, each system possesses distinct strengths and limitations across key performance areas. Hikvision excels in proprietary single-camera tracking but struggles with cross-camera identity persistence. YOLORe-IDNet offers real-time detection capabilities but lacks integrated systems and spatial mapping functionality. JARVIS-MoCap provides precise markerless pose estimation but requires controlled environments with calibrated camera arrays and complex setup procedures. All competitors demonstrate significant limitations in maintaining unified tracking across non-overlapping views, implementing accessible user interfaces, or providing flexible deployment options across various platforms—capabilities that are essential for comprehensive campus security applications.

The SpotOn stands apart from competitors through its unique combination of persistent identity tracking across non-overlapping camera views and blind spots, a capability not found in Hikvision's proprietary hardware-dependent solution which excels at single-camera tracking but fails across non-overlapping views with time gaps. Unlike YOLORe-IDNet's research implementation which lacks continuous identity maintenance, our system preserves identity even when subjects disappear temporarily. JARVIS-MoCap requires controlled environments and specialized equipment, while our system works with existing camera infrastructure and adapts to varying lighting conditions through self-calibrating algorithms. Our multi-model fusion approach ensures reliable tracking

even in campus blind spots where other systems lose track. The system's interactive 3D mapping provides security personnel with comprehensive spatial awareness that surpasses the limited single-view representations of competitors, making it uniquely suited for complex campus environments where maintaining continuous identity is critical for security operations.

2.2 Literature Review

The development of our multi-camera person tracking system builds upon established research while making specific architectural decisions to address the limitations of alternative approaches found in recent literature. This review is organized into two categories: foundational works that we directly leverage, and alternative methodologies that we analyzed but ultimately rejected in favor of our proposed pipeline.

2.2.1 Foundational Works Leveraged

Our system integrates several state-of-the-art components and datasets that provide the necessary building blocks for robust tracking.

ByteTrack (Zhang et al. [2]) We leverage ByteTrack's "track-everything" philosophy, which associates low-confidence detection boxes rather than discarding them. This foundational work forms the core of our intra-camera tracking module, allowing our system to maintain trajectory continuity through brief occlusions or lighting changes without the need for complex post-hoc repairs.

FastReID (He et al. [3]) For appearance modeling, we utilize the FastReID toolbox. Its modular architecture and powerful backbone networks provide the discriminative feature extraction capabilities essential for our Re-ID stage. We integrate this strictly as a feature extractor within our larger pipeline, separating it from the tracking logic to ensure modularity.

MTMMC Dataset (Woo et al. [4]) The MTMMC dataset serves as our primary training and benchmarking ground. Its diverse real-world scenarios (campus and factory) with multi-modal data provide the realistic conditions necessary to fine-tune our detection and Re-ID models for the specific challenges of our deployment environment.

Modular 3D Tracking (Bredereke et al. [5]) We adopt the conceptual framework of modular 3D tracking using RGB cameras. While Bredereke et al. focus on synchronization, we leverage the core idea of mapping 2D tracks to 3D space as a distinct pipeline stage, which informs our Space-based Matching module.

2.2.2 Key Insights and Adaptations from Recent Methodologies

To ensure our system addresses the complexities of real-world deployment, we analyzed several recent alternative methodologies. Rather than viewing these as competing solutions, we treated them as sources of critical insight, adapting and evolving their core concepts to better suit our modular, privacy-focused architecture.

From Unified Architectures to Modular Decoupling

Insight from Literature: Gautam et al. [6] demonstrated with YOLORe-IDNet that detection and re-identification tasks are deeply interconnected and can be effectively solved within a single computational pass.

Our Adaptation: We learned from the efficiency of this coupling but refined the implementation into a **Decoupled Architecture** (using RT-DETR and FastReID as separate modules). This evolution allows us to retain the benefit of specialized feature extraction while gaining system modularity, enabling us to upgrade the detection engine (e.g., to newer Transformer models) without disrupting the re-identification gallery—a critical requirement for long-term system maintainability.

From Reactive Repair to Proactive Prevention

Insight from Literature: Wang et al. [7] rely on a post-hoc cleanup strategy. They allow tracklets to contain mixed identities and subsequently analyze the feature history using Gaussian Mixture Models (GMM). If the GMM detects multi-modal feature distributions (indicating two distinct persons under one ID), the system splits the track.

Our Adaptation: We analyzed that these ID switches typically originate from track fragmentation during low-confidence periods (e.g., occlusion). We improved robustness by prioritizing **Proactive Prevention**. Integrating ByteTrack allows us to associate low-confidence detections, effectively bridging these gaps. Since the track continuity is maintained through the "dip" in confidence, the need for re-identification—and the risk of swapping IDs—is eliminated at the source.

From Conditional Geometry to Architectural Overlap

Insight from Literature: Wu et al. [8] proved that 3D geometric constraints are a powerful fallback when visual appearance is unreliable due to occlusion. they implemented this as a conditional state that toggles on during occlusion.

Our Adaptation: We validated the importance of geometry but advanced the implementation by making it the **Primary Architectural Bridge** (Space-Based Matching) rather than a conditional fallback. By continuously projecting all tracks to a unified 3D map, our system naturally bridges occlusions through multi-view spatial overlap, ensuring that geometric consistency is an inherent property of the system rather than a triggered state.

From Trajectory Clustering to Sequential Filtering

Insight from Literature: Hachiuma et al. [9] showed that in environments with uniform clothing, physical trajectory provided a more reliable signal than visual features, suggesting that appearance can sometimes be a distraction.

Our Adaptation: We synthesized this insight with visual matching using a **Sequential Filtering** approach. Our pipeline applies **Space-Based Matching** continuously to all tracks, projecting them to a unified map to handle overlapping fields and maintain global consistency. **Re-Identification** is triggered *conditionally*—only when a valid spatial link suggests a cross-camera transition. This allows us to handle indistinguishable uniforms using spatial logic as the primary filter, while reserving expensive visual matching for necessary confirmations.

Resource-Aware Scalability in Multi-Camera Pipelines

Insight from Literature: Standard multi-camera tracking implementations often deploy independent detection and tracking instances for each camera feed. This 1-to-1 topology results in linear resource scaling, where computational overhead increases directly with the number of cameras, quickly saturating GPU memory.

Our Adaptation: We devised a **One-to-Many** architecture to address this bottleneck. Instead of redundant model instances, we utilize a **Single Shared Detection Model** that multiplexes frames from all active cameras. This feeds into a **Custom ByteTrack Implementation** designed to manage independent tracking states for multiple camera contexts within a single execution thread. This design allows our system to scale to additional camera feeds with constant model memory overhead, significantly improving efficiency for campus-wide deployment.

In conclusion, while existing systems have made significant progress in multi-camera tracking, there remains a gap in solutions that effectively integrate persistent identity tracking, environmental adaptation, and interactive spatial mapping for campus applications. Our SpotOn addresses this gap by combining state-of-the-art computer vision techniques designed for practical deployment in real-world environments.

Unlike existing systems that prioritize either single-camera performance or algorithmic benchmarks, SpotOn differentiates itself through a system-level integration of cross-camera identity preservation and unified spatial visualization. Commercial solutions often remain constrained to camera-centric views, while research implementations typically lack

deployment-ready integration and usability. By combining identity persistence across non-overlapping cameras with spatial trajectory mapping and a user-centered interface, SpotOn introduces a practical reinterpretation of multi-camera tracking focused on operational effectiveness rather than isolated model performance.

Chapter 3

Requirement Analysis

3.1 Stakeholder Analysis

3.1.1 Primary Stakeholders

1. **Users:** The primary stakeholders are the users of our multi-camera tracking system, including security personnel, facility managers, and emergency response coordinators. Their interaction with the system is crucial for effective monitoring, incident response, and data-driven decision-making. For a detailed specification of our users, see 1.4.
2. **System Administrators:** These individuals are responsible for the installation, configuration, and maintenance of the system. Their needs include ease of deployment, system stability, and access to diagnostic tools.
3. **University/Factory Management:** These stakeholders are interested in the overall benefits of the system, such as improved safety, optimized resource allocation, and enhanced operational efficiency. They require reports and data summaries to justify the investment and track performance.
4. **MTMMC Dataset Providers/Owners:** The creators and distributors of the Multi-Target Multi-Camera (MTMMC) dataset used for training and evaluating the AI components of this project (4.5.3).
    - **Stake:** They have an interest in ensuring the dataset is used responsibly and ethically, strictly according to the

terms and conditions outlined in the usage agreement signed by the project team. This includes preventing misuse or unauthorized redistribution of the data.

- **Project Consideration:** The project must adhere to all stipulations in the dataset usage license. Compliance will be documented, and proper attribution/citation will be provided in all relevant project outputs.

3.2 User Stories

3.2.1 Continuously Track Individuals Across the Area

As a security officer,
I want to continuously track individuals as they move throughout the monitored area, even when they move between different camera views,
so that I can maintain uninterrupted surveillance of persons of interest and respond quickly to developing situations..

3.2.2 Maintain Tracking During Brief Disappearances

As a security officer,
I want to maintain the track of individuals even if they are temporarily out of sight of all cameras,
so that I can have a complete picture of a person's movements, even if they briefly go around corners or into areas without direct camera coverage..

3.2.3 Visualize Movement Paths

As a facility manager,
I want to see a visual representation of how people move through the facility, including their paths and locations over time,
so that I can understand space usage, identify congestion points,

and make informed decisions about facility layout and resource allocation..

3.2.4 Access Historical Movement Data

As a facility manager or security officer,
I want to access and analyze historical data on individual and group movements, including options to export this data,
so that I can perform in-depth analysis of movement patterns, identify trends, and generate reports to improve operational efficiency, space utilization, or support incident investigations. I can also share this data with other systems..

3.2.5 Prioritize Tracking of Specific Individuals

As a security officer/emergency coordinator,
I want to designate specific individuals for increased monitoring and receive immediate updates on their location,
so that I can focus on persons of interest during security incidents or emergencies, ensuring a rapid and effective response..

3.2.6 Ensure System Reliability and Uptime

As a system administrator,
I want to easily monitor system performance, troubleshoot issues, and perform necessary maintenance tasks,
so that I can keep the tracking system running smoothly and minimize disruptions to users..

3.3 Use Case Diagram

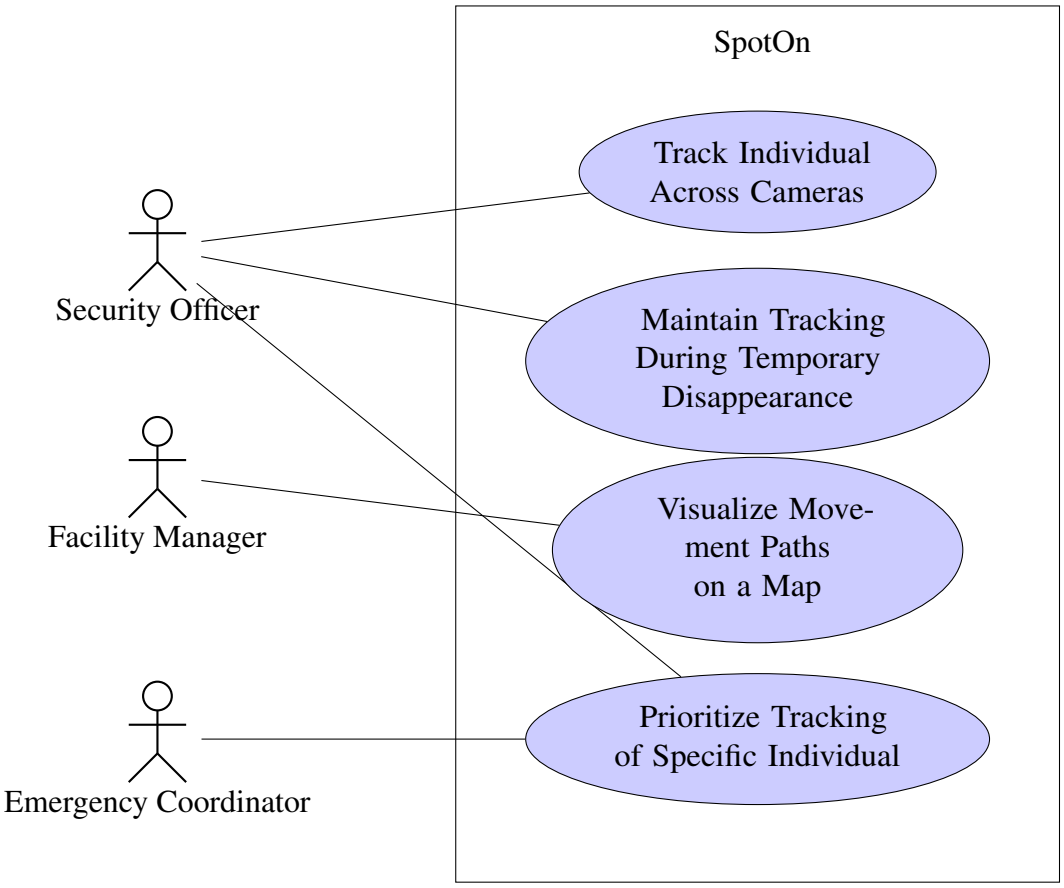

Figure 3.1: Use Case Diagram of SpotOn Multi-Camera Tracking System

The use case diagram (Figure 3.1) illustrates the interactions between different user roles (actors) and the core functionalities (use cases) of the SpotOn Multi-Camera Tracking System. The actors, shown on the left side of the diagram, represent the primary user groups interacting with the system:

- **Security Officer:** Interacts with tracking, maintaining tracking through gaps, and prioritizing specific individuals.
- **Facility Manager:** Primarily interacts with visualizing movement paths.
- **Emergency Coordinator:** Interacts with prioritizing the tracking of specific individuals during critical events.

This diagram provides a high-level overview of who uses the system and for what main purposes.

3.4 Use Case Model

3.4.1 Use Case 1: Track Individual Across Cameras

Actors: Security Officer

Description: Track an individual's movements across multiple camera views.

Scenario:

1. Log into SpotOn.
2. Select a location and date.
3. View available camera feeds.
4. See people with assigned IDs in each camera view.
5. View a map showing detected individuals.
6. Select an individual in any camera view.
7. See that individual highlighted in all cameras where they appear.
8. View their movement history on the map.

3.4.2 Use Case 2: Maintain Tracking During Temporary Disappearance

Actors: Security Officer
Description: Continue tracking an individual even when temporarily out of camera view.
Scenario:

1. Track an individual in the system.
2. When the person moves out of all camera views, the system marks this state.
3. When the person reappears, the system reconnects their identity.
4. See the individual's complete path, with estimated routes during gaps shown as dashed lines.

3.4.3 Use Case 3: Visualize Movement Paths on a Map

Actors: Facility Manager
Description: View movement patterns within a facility for space utilization analysis.
Scenario:

1. Log into SpotOn.
2. Select a location and date.
3. View a bird's-eye map of the facility.
4. See movement paths of all individuals overlaid on the map.
5. Filter paths by camera view.
6. Select an individual to highlight only their path.

3.4.4 Use Case 4: Prioritize Tracking of Specific Individual

Actors: Security Officer/Emergency Coordinator
Description: Focus system resources on tracking a specific individual.
Scenario:

1. Log into SpotOn.
2. Select a location and date.
3. View camera feeds and detected individuals.
4. Designate an individual for prioritized tracking.
5. See visual highlighting of the prioritized individual in all cameras.
6. View their position, detection time, tracking duration, and movement details.

3.5 User Interface Design

The user interface for the multi-camera tracking system is designed to offer a clear, intuitive, and efficient experience for security personnel, facility managers, and other stakeholders, focusing on retrospective analysis of recorded video feeds. The interface prioritizes the visualization of movement patterns, easy access to detailed tracking information.

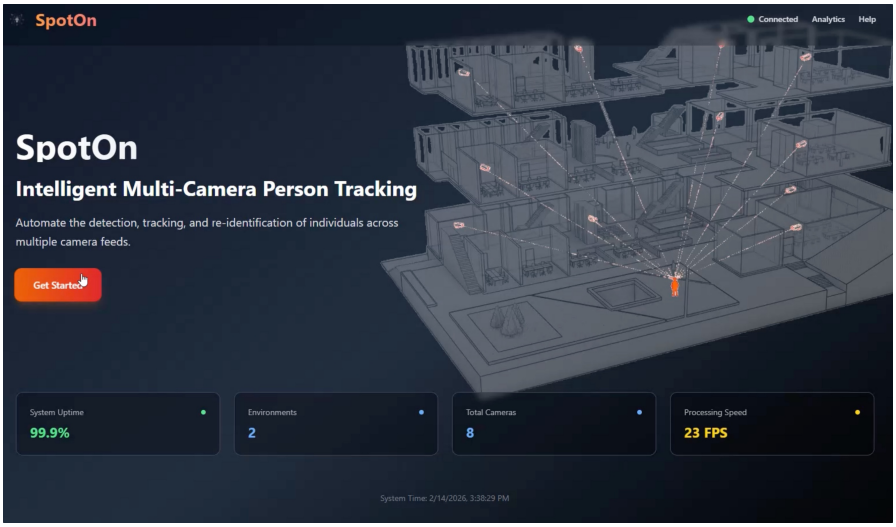

Figure 3.2: Landing Page

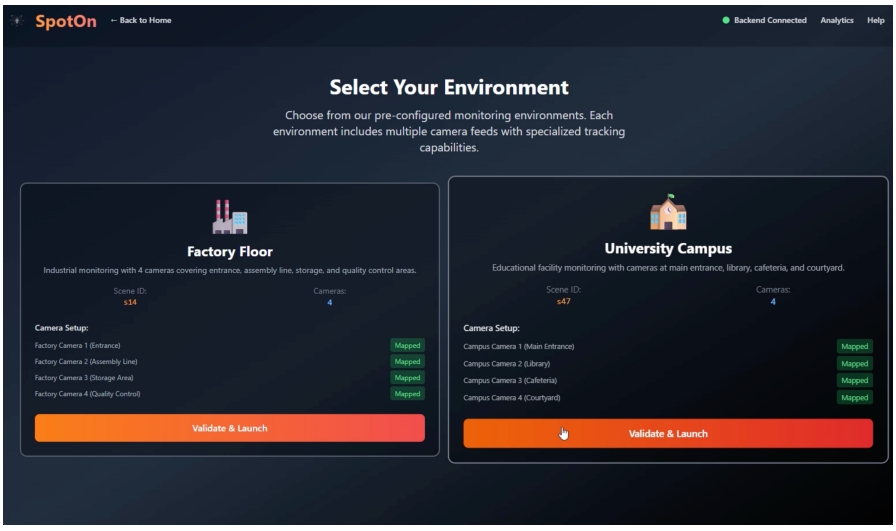

Figure 3.3: Environment Selection

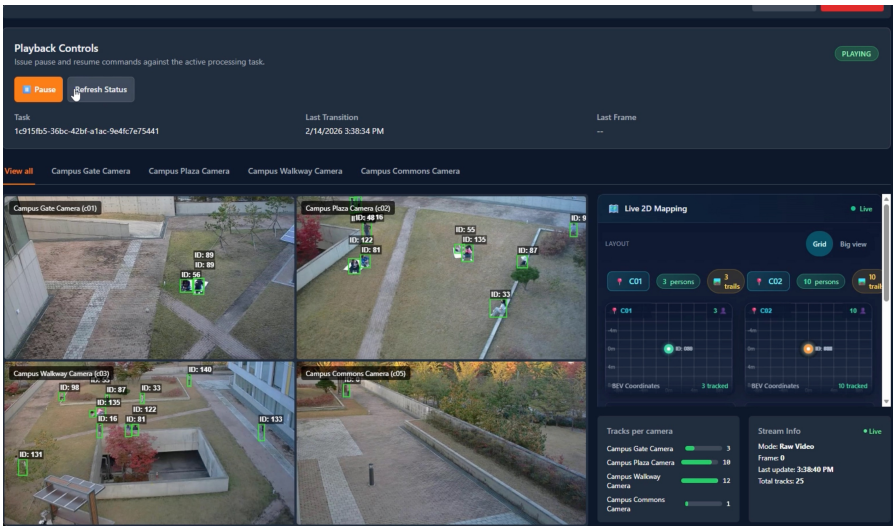

Figure 3.4: Video Feed Overview

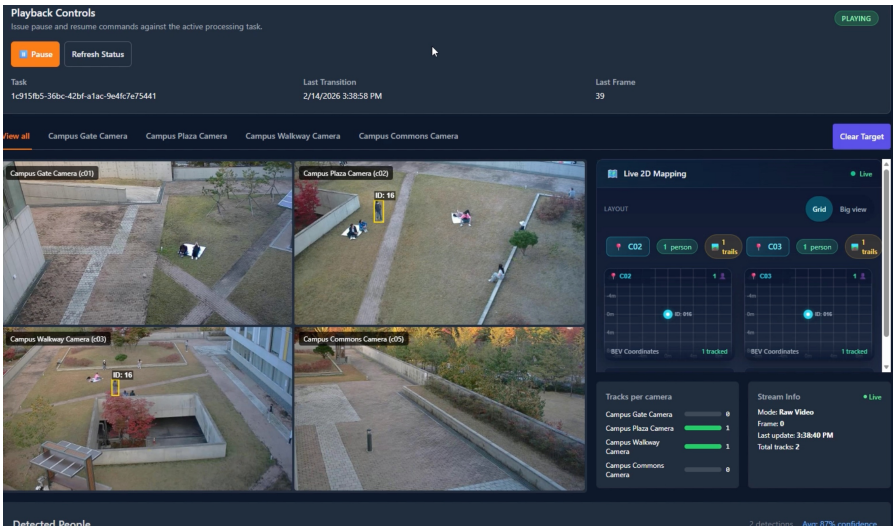

Figure 3.5: Tracking View

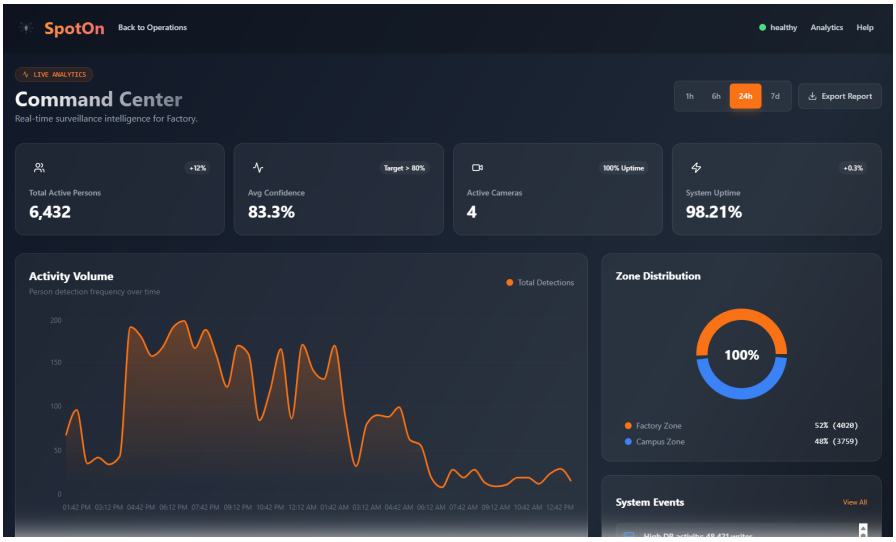

Figure 3.6: Analytics Dashboard - Overview

Figure 3.7: Analytics Dashboard - Detailed Metrics

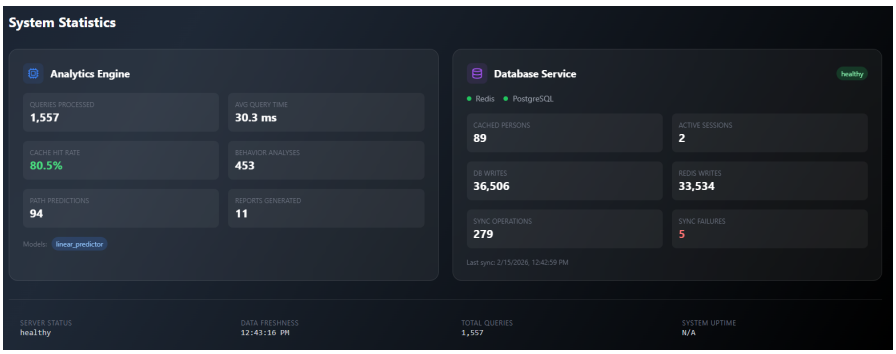

Figure 3.8: Analytics Dashboard - Trends

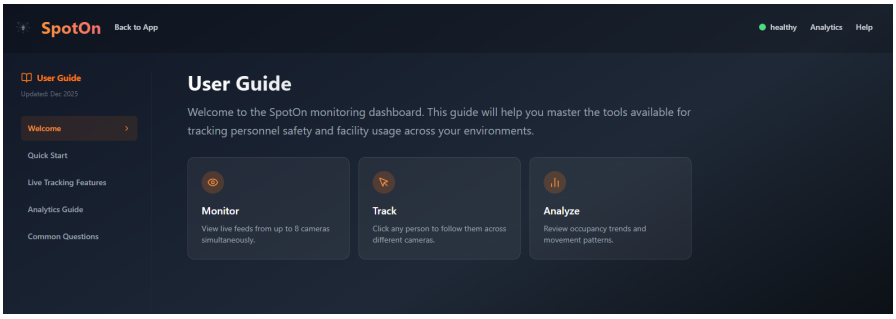

Figure 3.9: Help and Documentation

The user interface presents a streamlined workflow designed for efficiency. The process begins at the **Landing Page** (Figure 3.2), which serves as the entry point. Users then proceed to **Environment Selection** (Figure 3.3), where they choose the specific campus or factory environment to analyze.

Once an environment is selected, the **Video Feed** (Figure 3.4) displays the real-time or recorded camera streams, offering a comprehensive operational view. From here, users can enter the **Tracking View** (Figure 3.5), which allows for the specific monitoring of individuals across multiple cameras, utilizing the system's re-identification capabilities.

For deeper insights, the **Analytics Dashboard** (Figures 3.6, 3.7, 3.8) provides data-driven visualizations of movement patterns, detection statistics, and other key metrics. Finally, the **Help** section (Figure 3.9) offers documentation and support resources to assist users in navigating the system capabilities.

3.6 Target and Development System

This section outlines the technical requirements for the development and operation of our multi-camera tracking system, focusing on the capabilities needed to support the user stories and use cases identified earlier.

3.6.1 Data Handling and Input Capabilities

To effectively analyze the multi-camera tracking scenarios, the system must process recorded video data from the MTMMC dataset (4.5.3). Key requirements include:

- **Scalable Data Storage Access:** The system must interface with storage solutions capable of holding significant volumes of video data. It requires efficient access mechanisms to retrieve data based on time ranges and camera identifiers, ensuring recorded footage is readily available for retrospective analysis.
- **Real-Time Data Processing:** The system needs to process live video feeds from multiple cameras in real-time. This capability is essential for immediate analysis of current events and synchronized movement tracking across the facility.
- **Flexible Data Source Configuration:** The system needs a configuration-driven mechanism to specify the location of the input data sources, select specific scenarios or time windows for analysis, and identify the relevant cameras.

3.6.2 Core Tracking and Analysis Engine Requirements

The central function of our multi-camera tracking system relies on sophisticated algorithms capable of tracking individuals across multiple camera views. This necessitates several core analytical capabilities:

- **Robust Person Detection:** The system requires the ability to accurately detect individuals within diverse single-camera

views, handling variations in appearance, lighting, and partial occlusions typical in campus or factory environments (1.1).

- **Intra-Camera Tracking Consistency:** Capability to maintain stable tracking of individuals within the field of view of a single camera over consecutive frames, assigning temporary identifiers.

- **Cross-Camera Re-Identification Logic:** A fundamental requirement is the implementation of logic to associate and match individuals across non-overlapping camera views (1.2). This involves comparing extracted appearance features to maintain a persistent identity even with temporal gaps or changes in perspective.

- **Unified Coordinate System Transformation:** The system must be capable of transforming detected locations from individual camera image planes into a common spatial reference frame (e.g., a top-down map). This is essential for visualizing continuous movement paths across the entire monitored environment (1.3.1).

- **Efficient State and Trajectory Management:** Requires mechanisms for managing the state of numerous tracked individuals (current location, persistent ID, last seen time/camera) and for storing historical trajectory data. This includes both data structures for processing and persistent storage for historical analysis.

3.6.3 User Interface and Visualization Needs

To deliver actionable insights to users like security officers and facility managers (1.4), the system must provide an intuitive interface focused on effective visualization of tracking data:

- **Multi-Feed Presentation:** The user interface must be designed to display multiple camera feeds concurrently, allowing operators to monitor different areas simultaneously.
- **Integrated Data Overlay:** Requires the capability to overlay tracking information (such as bounding boxes, persistent IDs, status indicators) directly onto the corresponding camera feeds in a clear and understandable manner.
- **Spatial Map Display:** A core requirement is the visualization of movement on a unified map (1.3.1). The interface must display current positions and historical paths of tracked individuals on this map to provide a comprehensive view of movement patterns.
- **Time-Based Navigation:** Users need the ability to select specific time periods for review and analysis, with controls to navigate through the recorded footage across multiple cameras simultaneously.
- **Responsive Display Framework:** The frontend component must be built using technologies that support efficient rendering of tracking data and provide a clear presentation suitable for monitoring and analysis tasks.

Chapter 4

Software Architecture Design

4.1 Software Architecture

The SpotOn backend is built upon a **Layered Architecture** pattern using the **FastAPI** framework. This design promotes separation of concerns, maintainability, and scalability. The system is structured into clearly defined layers, each responsible for specific aspects of the data processing pipeline, from handling HTTP/WebSocket requests to executing complex AI logic and managing data persistence.

The architecture consists of the following key layers:

4.1.1 API Layer (Presentation)

The API Layer is the entry point for all external interactions. It is responsible for routing requests, validating input data using Pydantic schemas, and handling authentication/authorization.

- **Technology:** FastAPI Routers (‘APIRouter‘), WebSockets.
- **Responsibilities:**
  - **HTTP Endpoints:** RESTful APIs for environment management, analytics queries, and system control.
  - **WebSocket Endpoints:** Real-time bi-directional communication for pushing detection metadata and receiving client heartbeats.
  - **Middleware:** Handles cross-cutting concerns such as CORS, logging, and security headers.

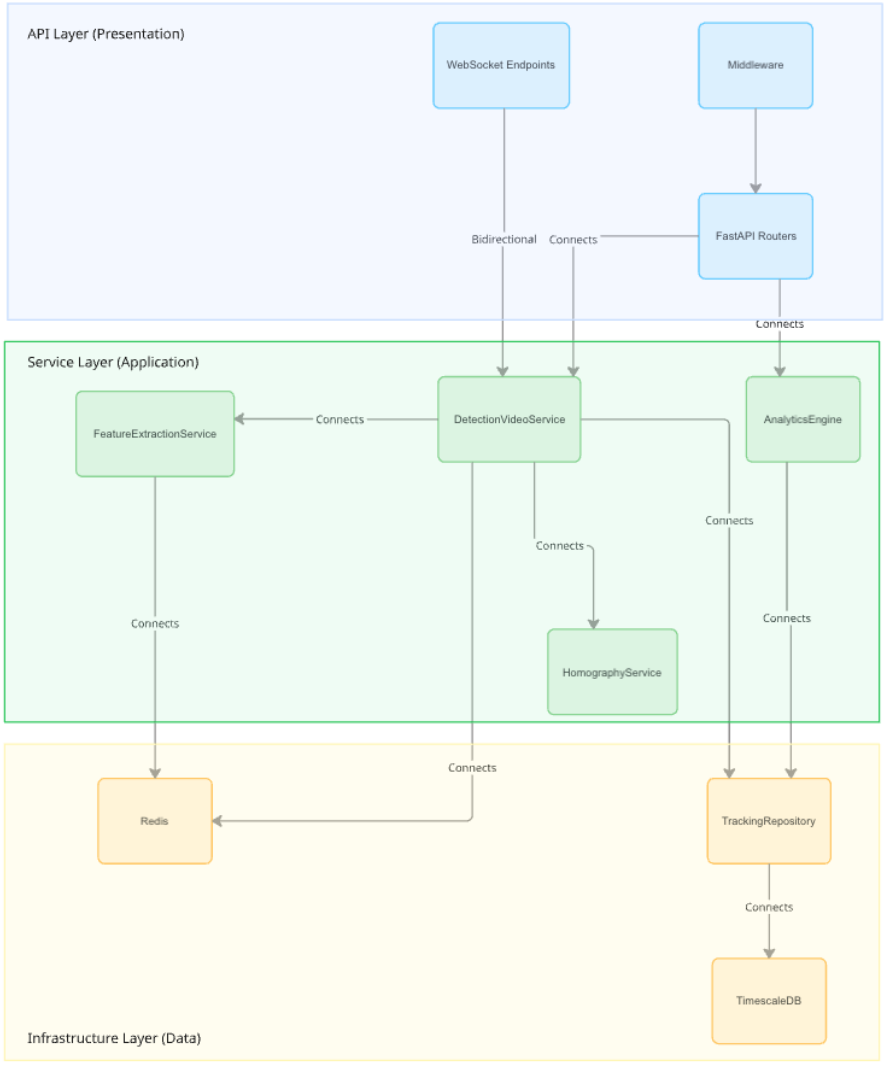

Figure 4.1: Layered Architecture Diagram

4.1.2 Service Layer (Application Logic)

The Service Layer contains the core business logic and orchestrates the data flow between the API and Infrastructure layers. This is where the AI processing pipeline resides.

- **Key Components:**

  - **DetectionVideoService:** The central orchestrator that manages video stream processing, invoking object detectors (RT-DETR) and trackers.
  - **HomographyService:** Manages the transformation of 2D camera coordinates to 2D map coordinates using pre-computed homography matrices.
  - **FeatureExtractionService:** Wraps the FastReID model to extract appearance embeddings for person re-identification.
  - **AnalyticsEngine:** Aggregates real-time data into meaningful metrics (e.g., people counting, dwell time).

4.1.3 Infrastructure Layer (Data Access)

The Infrastructure Layer abstracts the underlying data storage technologies, providing a clean API for the Service Layer to persist and retrieve data.

- **Components:**
  - **Repositories:** The ‘TrackingRepository‘ manages CRUD operations for tracking data, abstracting SQL queries.
  - **Database: TimescaleDB** (PostgreSQL extension) is used for efficient storage and querying of time-series tracking data.
  - **Cache: Redis** is utilized for high-speed caching of Re-ID embeddings and transient application state.

4.1.4 Domain/Model Layer

This layer defines the data structures and business entities used throughout the application, ensuring type safety and consistency.

- **Pydantic Models:** Used for data validation and serialization in the API layer (e.g., ‘Detection‘, ‘Track‘, ‘AnalyticsSummary‘).
- **SQLAlchemy Models:** Mapped to database tables for ORM-based data access.

4.2 Sequence Diagram

The sequence diagram visualizes the real-time interaction between system components during the tracking process. It illustrates how data flows from the video source through the AI processing pipeline and finally to the user interface.

4.2.1 Real-Time Tracking Process

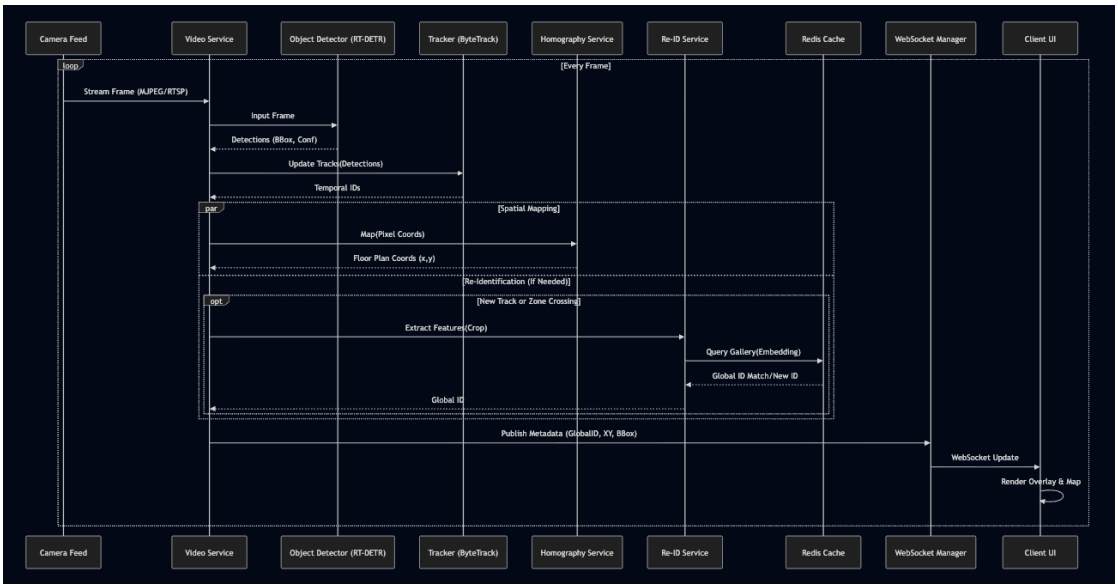

Figure 4.2: Sequence Diagram for Real-Time Multi-Camera Tracking

Figure 4.2 depicts the flow when a detection task is active: 1. The **Video Service** ingests frames from the camera feed. 2. Frames are passed to the **Object Detector** (RT-DETR) to identify individuals. 3. Detections are sent to the **Tracker** (ByteTrack) to assign temporal IDs. 4. The **Homography Service** maps pixel coordinates to the 2D floor plan. 5. If a track crosses into a new zone or camera view, the **Re-ID Service** extracts features and queries the **Redis** cache for a global identity match. 6. The aggregated tracking metadata (Global ID, Location, Bounding Box) is pushed to the **WebSocket Manager**. 7. The **Client UI** receives the update and renders the bounding boxes and map markers in real-time.

4.3 Pipeline

4.3.1 AI Processing Pipeline

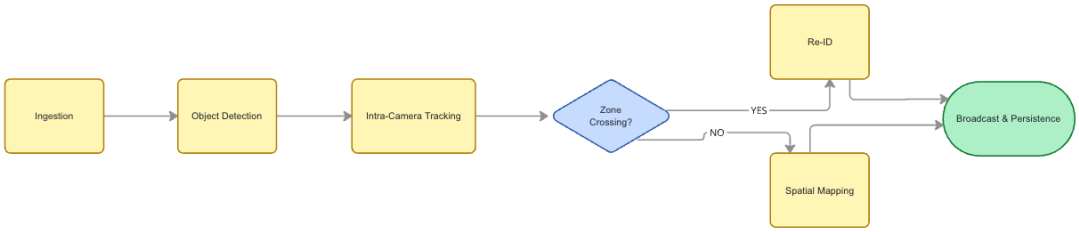

Figure 4.3: Data Processing Pipeline for SpotOn

The SpotOn system employs a strictly ordered pipeline to transform raw video data into actionable tracking insights. As shown in Figure 4.3, the pipeline consists of the following stages:

1. **Ingestion:** Video frames are extracted from source files or streams (RTSP/MJPEG) and pre-processed (resized, normalized) for model input.
2. **Object Detection:** The RT-DETR model analyzes each frame to output bounding boxes and confidence scores for all detected persons.
3. **Intra-Camera Tracking:** The ByteTrack algorithm processes detections to maintain identity consistency across consecutive frames within a single camera view, handling short-term occlusions.
4. **Spatial Mapping:** The center point of each tracked person's foot-box is transformed using a homography matrix to obtain real-world (x, y) coordinates on the facility's floor plan.
5. **Re-Identification (Re-ID):**
   - **Feature Extraction:** For tracks crossing camera boundaries, the FastReID model extracts high-dimensional appearance embeddings.

- **Matching:** These embeddings are compared against a gallery of active tracks in Redis (using cosine similarity) to determine if the person matches a known Global ID.

6. **Broadcast & Persistence:** The final resolved track data (Global ID, Map Coordinates) is persisted to TimescaleDB for history and broadcast via WebSocket for real-time visualization.

4.4 Schemas

To ensure consistent data exchange between the Python backend, the AI models, and the React frontend, SpotOn defines strict data schemas. The following are the core schemas used in the system:

4.4.1 Person Track

The fundamental unit of tracking data, representing a detected individual at a specific point in time.

```
{
    "track_id": int,
    "global_id": string,
    "bbox_xyxy": [
        float, float, float, float
    ],
    "confidence": float,
    "map_coords": [
        float, float
    ],
    "is_focused": boolean
}
```

4.4.2 Environment Configuration

Defines the setup for a tracking location (e.g., Campus or Factory).

```
{
    "environment_id": string,
    "name": string,
    "camera_count": int,
    "zone_count": int,
    "has_data": boolean,
    "last_updated": string
}
```

4.4.3 WebSocket Tracking Update

The payload sent to the frontend for real-time visualization.

```
{
    "type": "tracking_update",
    "task_id": "uuid",
    "camera_id": "string",
    "timestamp": "ISO 8601 String",
    "camera_data": {
        "tracks": [ PersonTrack, ... ]
    }
}
```

4.5 AI Component

4.5.1 Overview of the AI Component

The AI component of our system is designed to handle object detection, tracking, and re-identification (Re-ID) for **real-time** processing of video feeds. This component serves as the core intelligence of the application, enabling it to recognize and track objects across multiple camera views live. The system prioritizes **low-latency real-time processing** to support immediate situational awareness, optimized for high-throughput concurrency using our **One-to-Many** architecture. The AI component integrates with other system modules by streaming analysis through the following primary communication methods:

- **MJPEG Streaming:** For delivering raw video frames to the frontend with minimal latency.
- **WebSocket:** For pushing real-time detection and tracking metadata overlays.
- **REST API:** For handling control commands and configuration requests.

This multi-protocol approach supports the system's overall objectives of accurate object tracking and identification in real-time.

4.5.2 AI Model Description

Object Detection

- **Detection Model:** RT-DETR (Real-Time Detection Transformer) - Selected for its balance of speed and accuracy, specifically its ability to handle crowded scenes better than traditional YOLO-based approaches. This model is responsible for identifying "person" objects within each camera frame.
- **Key Features:** End-to-end detection without NMS post-processing, flexible speed tuning via decoder layers
- **Function:** Responsible for identifying and localizing objects (people) within video frames

Tracking System

- **Model Type:** Custom ByteTrack Implementation (Multi-Context)
- **Architecture:** Modified ByteTrack algorithm designed for One-to-Many scalability
- **Key Features:** Handles multiple camera streams using a single detector instance. It acts as a centralized state manager, maintaining independent tracking histories for infinite camera contexts simultaneously without instantiating separate trackers for each feed.

- **Function:** Maintains identity consistency of detected objects across consecutive frames for all active cameras efficiently.

Space-Based Location Prediction

- **Type:** Algorithmic approach (not pure ML)
- **Purpose:** Handle overlapping camera views where multiple cameras observe the same physical space from different angles
- **Method:** Convert 3D location from Camera A into 2D floor plan coordinates using homography, then map back to 3D coordinates in Camera B to predict the person's location
- **Function:** Enables intra-frame global ID assignment for persons visible in overlapping camera regions

Re-Identification (Re-ID) System

- **Model Type:** FastReID deep convolutional neural network
- **Architecture:** Specialized deep learning network that extracts discriminative features for person re-identification
- **Key Features:** Supports open-set Re-ID methods, critical for real-world applications where previously unseen identities may appear
- **Function:** Associates identities across different camera views and non-consecutive frames (across-frame matching)

The selection of the Re-Identification model was informed by comparative evaluation using standard Re-ID benchmarks on the MTMMC dataset, including Rank-1 accuracy and mean Average Precision (mAP).

Based on this evaluation, FastReID was chosen for its strong discriminative performance and modular design, allowing it to be reliably integrated as a conditional identity confirmation component within the overall tracking pipeline.

4.5.3 Dataset Information

Source

The MTMMC (Multi-Target Multi-Camera) dataset is utilized for training and evaluating our AI models. This dataset is collected from two distinct environments: a campus and a factory, designed to provide a challenging testbed for real-world surveillance and tracking scenarios.

Size and Composition

- **Total Scenarios:** 25 scenarios (13 campus, 12 factory)
- **Number of Cameras:** 16 multi-modal cameras (RGB and thermal)
- **Total Frames:** Approximately 3.05 million frames
- **Frame Rate:** 23 frames per second
- **Video Duration:** Approximately 5.5 minutes per scenario
- **Dataset Split:**
  - *Training Set:* 14 scenarios (7 campus, 7 factory)
  - *Validation Set:* 5 scenarios (3 factory, 2 campus)
  - *Test Set:* 6 scenarios (2 factory, 4 campus)

Technical Specifications

- **Resolution:** RGB: 1920x1080 pixels, Thermal: 320x240 pixels
- **Annotation Process:** Semi-automatic labeling with manual correction
- **Multi-Modal Data:** RGB and thermal data are spatially aligned and temporally synchronized

Key Features

- **Multi-Camera Views:** Multiple camera angles to train models for cross-view recognition
- **Environmental Diversity:** Different times of day, weather conditions, and seasons
- **Challenging Scenarios:** Occlusions, varying lighting conditions, and noisy detections
- **Thermal Data:** Enhances tracking accuracy in scenarios with limited visibility

4.5.4 Training Process

Training Environment

- **Hardware:** NVIDIA RTX 4060 GPU
- **Software:** PyTorch deep learning framework
- **Computing Resources:** Local workstation for model fine-tuning

Hyperparameters

- **Learning Rate:**
  - RT-DETR: 0.0001 (default)
  - ByteTrack and FastReID: 0.001
- **Batch Size:**
  - RT-DETR: 16
  - ByteTrack and FastReID: 16
- **Number of Epochs:**
  - RT-DETR: Fine-tuned on MTMMC dataset
  - ByteTrack and FastReID: 50 epochs
- **Optimizer:** AdamW with weight decay of 0.0001

Data Augmentation

- **Techniques:** Random cropping, horizontal flipping, color jittering, random erasing
- **Purpose:** Prevent overfitting and improve model generalization
- **Implementation:** Applied during training using PyTorch transformations

4.5.5 Evaluation Metrics

Object Detection Metrics

- **Precision:** Measures the accuracy of positive predictions
- **Recall:** Measures the ability to find all relevant instances
- **Average Precision (AP):** Summarizes the precision-recall curve
- **Confidence Threshold:** 0.5 for determining true positives

Tracking Metrics

- **Multiple Object Tracking Accuracy (MOTA):** Measures overall tracking accuracy, considering false positives, missed detections, and identity switches
- **Multiple Object Tracking Precision (MOTP):** Measures the precision of tracking results
- **ID F1 Score (IDF1):** Measures the identity preservation capabilities
- **Identity Switches (IDSW):** Counts the number of identity switches

Re-Identification Metrics

- **Rank-1 Accuracy:** Percentage of queries where the correct match is the top result
- **Mean Average Precision (mAP):** Average precision across all ranks

- **Cumulative Matching Characteristic (CMC) Curve:** Measures the probability of finding the correct match within the top-k matches

4.5.6 Best Practices

Model Versioning

- **Version Control:** Each trained model is versioned and stored in a centralized repository
- **Documentation:** Detailed documentation maintained for each model version, including:
  - Hyperparameters used for training
  - Training and validation performance metrics
  - Dataset version and preprocessing details
  - Dependencies and environment specifications
- **Backup Strategy:** Regular backups of model weights and checkpoints to prevent data loss

Continuous Learning

- **Data Updates:** Periodic dataset updates with new data to keep models current
- **Model Fine-tuning:** Regular model adjustments to adapt to changes in data distribution
- **Performance Monitoring:** Continuous evaluation of model performance on validation data
- **Feedback Loop:** Retrieval accuracy will be improved using **measurement and feedback loops** from system operators.

Ethical Considerations

- **Bias Mitigation:** Regular audits to ensure models do not exhibit biases based on gender, race, or other sensitive attributes

- **Transparency:** Model decision-making processes made transparent to ensure accountability
- **Privacy Protection:** Strict adherence to privacy regulations regarding the use of personal data
- **Security Measures:** Implementation of robust security protocols to prevent unauthorized access to models and data

4.5.7 Integration and Deployment

APIs

- **RESTful APIs:**
  - */api/detection:* Endpoint for object detection requests
  - */api/tracking:* Endpoint for tracking requests
  - */api/reid:* Endpoint for re-identification requests
  - */api/batch-process:* Endpoint for batch processing of video data
- **WebSocket:**
  - *ws://hostname/tracking:* Socket for real-time tracking updates
  - *ws://hostname/notification:* Socket for system notifications
- **Documentation:** Comprehensive API documentation using OpenAPI specification

Deployment Strategy

- **Containerization:**
  - Docker containers for encapsulating each AI model
  - Docker Compose for managing multi-container deployments
- **Orchestration:**
  - To handle real-time processing of multiple camera feeds at scale, the system uses **Kubernetes** for service orchestration.

  - The system features a **modular architecture** where detection and tracking services are containerized using **Docker**.
  - We conducted a **risk and impact assessment** regarding the privacy implications and computational bottlenecks of multi-camera tracking.
  - Horizontal Pod Autoscaling for handling varying loads
- **CI/CD Pipeline:**
  - Automated testing and deployment using GitHub CI/CD
  - Development will follow an **evaluation-driven development** approach, incorporating automated test cases, **regression testing** for the models, and **load/performance tests** to ensure real-time latency.

Database and Caching

- **Scalable Knowledge Base:**
  - The system implements a scalable knowledge base using hybrid search, combining keyword filtering (time/location) with vector search (FastReID embeddings stored in a vector database).
  - Stores time-series data such as tracking events.
- **Data Governance:**
  - The system enforces strict metadata governance and maintains data lineage for all captured video segments to ensure security compliance.
- **Redis:**
  - Implements real-time caching for tracking results
  - Configured with appropriate eviction policies and persistence
- **Data Retention:**
  - Configurable retention periods for different types of data
  - Automated data archiving for long-term storage

4.5.8 Conclusion

The AI component is a critical element of our system, enabling sophisticated **real-time** processing of video feeds for object detection, tracking, space-based location prediction, and re-identification. By leveraging state-of-the-art models like RT-DETR, ByteTrack, and FastReID, trained on the comprehensive MTMMC dataset, our system achieves high accuracy even in challenging scenarios. The training methodology using NVIDIA RTX 4060 GPU, combined with continuous learning approaches and ethical considerations, ensures that the AI component remains effective and responsible. The space-based location prediction algorithm enables intra-frame global ID assignment for overlapping camera views, complementing the Re-ID model for across-frame matching. The seamless integration with other system components via MJPEG, WebSocket, and REST APIs provides a scalable and reliable real-time solution for our users.

Chapter 5

AI Component Design

5.1 Business Context and AI Integration

SpotOn tackles the complex challenge of tracking individuals across multiple camera views in environments like campuses and factories. The core task requires recognizing and associating individuals as they move between potentially non-overlapping camera fields, often amidst changing appearances, viewpoints, lighting conditions, and occlusions. This inherent complexity and real-world variability make manual tracking labor-intensive, error-prone, and difficult to address effectively with traditional rule-based programming. Artificial Intelligence is therefore central to SpotOn, providing the automation necessary to significantly improve this process.

The primary AI-driven workflow resides within the system's backend processing core, designed as a service-oriented architecture. This backend, built with FastAPI, orchestrates a pipeline for **real-time analysis of live video streams**. Video data allows for immediate ingestion from IP cameras (e.g., via RTSP) and processed in-memory for low-latency analysis. The AI pipeline involves several key steps for each frame extracted from these real-time feeds:

1. **Person Detection:** Identifying individuals within each camera frame using high-performance detection models (e.g., RT-DETR).

2. **Intra-Camera Tracking:** Maintaining consistent temporary IDs for detected individuals within a single camera's view over consecutive frames, utilizing robust tracking algorithms (e.g., ByteTrack).

3. **Space-Based Matching (Continuous):** For every tracked individual, transforming 2D pixel coordinates to a unified 3D map location using homography. This "always-on" step handles overlapping views by geometrically predicting location, allowing the system to bridge tracks spatially before resorting to visual features.
4. **Feature Extraction:** Generating unique appearance embeddings (features) from detected person images using models like FastReID, preparing them for potential identification needs.
5. **Cross-Camera Re-Identification (Conditional):** Triggered only when a target leaves one camera view and enters another (non-overlapping) view. This step matches appearance embeddings against the gallery to confirm identity when spatial continuity alone is insufficient.
6. **Spatial Mapping & Visualization:** The final 2D map output which visualizes the consolidated tracks, derived from the continuous Space-Based Matching data.

These AI components perform the essential visual analysis and identity association. Tracking metadata (bounding boxes, global IDs, map coordinates, frame timestamps) is then compiled and can be streamed to a frontend client via WebSockets for visualization, while historical data is stored in TimescaleDB. Figure 5.1 provides a high-level illustration of this system architecture and the integration points for these AI modules.

In real-world surveillance environments, maintaining consistent identification of individuals across multiple cameras cannot be achieved through manual monitoring or rule-based systems due to scale, appearance variation, occlusions, and temporal gaps. Artificial Intelligence therefore serves as a core enabler of the SpotOn system rather than an auxiliary feature. By automating person detection, tracking, and cross-camera re-identification, the system transforms fragmented video feeds into coherent identity trajectories and spatial representations. This AI-driven process significantly reduces cognitive load on operators, shortens investigation time, and enhances situational awareness, making AI essential for practical deployment and real-world adoption.

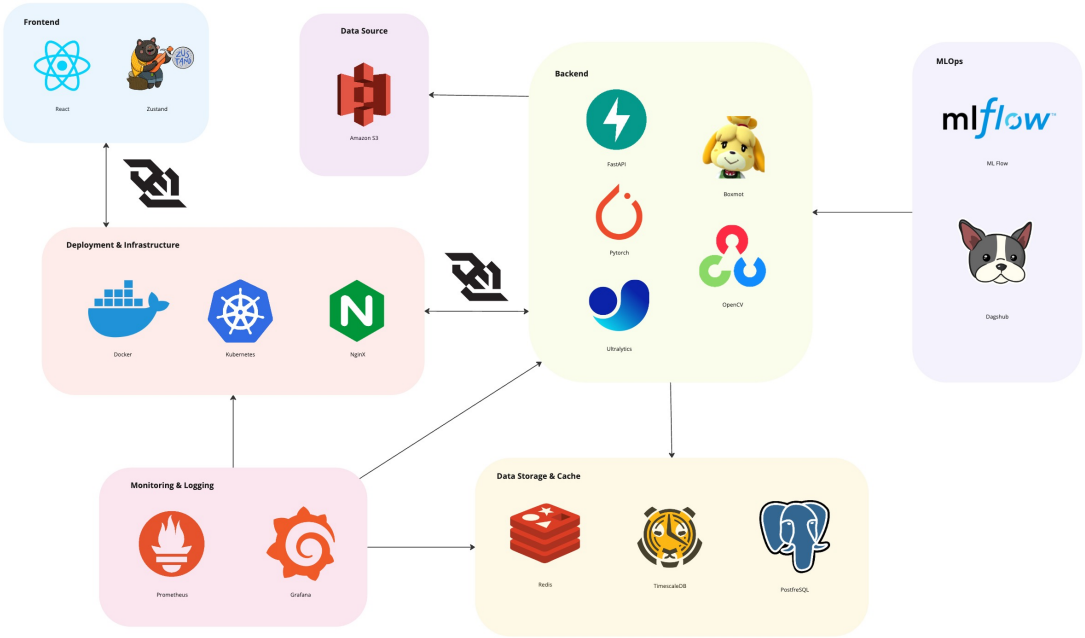

Figure 5.1: System Architecture Overview Highlighting AI Component Integration

The justification for employing AI stems from the nature of the problem itself:

- **Complexity Beyond Rules:** Recognizing and matching people under diverse visual conditions involves intricate pattern recognition that surpasses the capabilities of predefined rules. AI models excel at learning these complex patterns directly from data.
- **Scalability Requirement:** Manually monitoring and correlating feeds from numerous cameras is impractical. AI offers an automated solution capable of handling large data volumes and multiple camera streams for retrospective analysis.
- **Adaptability to Dynamic Environments:** Real-world surveillance scenarios are constantly changing. AI, particularly deep learning, offers better generalization and adaptation to variations in lighting, crowds, and individual appearances compared to static algorithms.
- **Value Despite Imperfection:** Achieving flawless accuracy in cross-camera tracking is exceptionally difficult due to the inherent ambiguities and challenges (e.g., severe occlusions, drastic appearance changes). However, an AI system achieving high, albeit imperfect, accuracy still delivers significant value. It drastically reduces the manual workload for operators and provides a level of situational awareness unattainable through manual means or simpler systems. Even with occasional errors, such as false positive matches or missed detections, the system's ability to automatically correlate identities across most views aligns with the primary goal of enhancing operational efficiency and speeding up investigations. The objective is a substantial improvement over the baseline, accepting a trade-off where occasional inaccuracies are outweighed by the overall gains in automation and insight.

5.2 Goal Hierarchy

Ensuring the AI components effectively contribute to the overall success of SpotOn requires aligning goals across multiple levels. This hi-

erarchical approach clarifies the purpose of each component and provides measurable targets for development and evaluation, moving from broad organizational objectives down to specific AI model performance.

- **Organization Level Goals:**
  - Enhance overall campus/factory safety and security posture.
  - Improve operational efficiency for security and facility management teams.
  - Optimize resource allocation (e.g., personnel deployment, space utilization).
  - Reduce costs associated with manual surveillance monitoring and incident investigation time.

  Measurement: Success at this level is measured via key performance indicators such as reduction in security incidents, faster incident resolution times, documented improvements in space utilization, and potential reduction in monitoring personnel hours.

- **System Level Goals (SpotOn):**
  - Provide accurate and continuous tracking of individuals across multiple, potentially non-overlapping, camera views from recorded footage.
  - Maintain persistent identity of individuals even through temporary disappearances or view changes in the analyzed data.
  - Visualize individual movement paths clearly on a unified spatial map.
  - Enable efficient retrospective analysis of movement patterns and incidents.
  - Offer a reliable and usable interface for target users.

  Measurement: System success is assessed through end-to-end tracking accuracy metrics (e.g., overall MOTA/IDF1 on test sequences), system processing throughput for batch analysis, task completion time for key user scenarios, and user satisfaction surveys.

- **User Level Goals:**
  - *Security Officer:* Faster POI location/monitoring during investigations, efficient evidence gathering from historical data, reduced manual correlation effort. (3.2.1, 3.2.2, 3.2.5)
  - *Facility Manager:* Understanding pedestrian flow from past data, identifying bottlenecks, optimizing space. (3.2.3)
  - *Emergency Coordinator:* Rapid location finding during post-incident analysis, reviewing evacuations. (3.2.5)
  - *Analytics Specialist:* Accessing historical data for trend analysis. (3.2.4)

  Measurement: User-level success is evaluated by measuring time savings for specific analytical tasks (e.g., time to locate a person's full path in historical data), task success rates, positive feedback on usability and effectiveness, and the perceived accuracy of generated movement analyses.

- **AI Model Level Goals:**
  - *Detection Model:* Achieve high precision and recall for person detection across diverse conditions in recorded video.
  - *Tracking Model:* Minimize identity switches (IDSW) and fragmentation within single cameras, achieving high MOTA and IDF1 scores on benchmark datasets.
  - *Re-Identification Model:* Generate highly discriminative appearance features, achieving high Rank-1 accuracy and mAP for cross-camera matching on benchmark datasets.
  - *Spatial Mapping Component:* Accurately transform image coordinates to map coordinates with minimal projection error using pre-supplied calibration data.

  Measurement: AI model performance is quantified using standard computer vision benchmarks on relevant datasets (like the MTMMC test split), including metrics such as Average Precision (AP), Recall, MOTA, MOTP, IDF1, IDSW, Rank-1 Accuracy, mAP, CMC curves, and coordinate projection error where applicable.

To comprehensively evaluate the system and ensure it meets its objectives, a variety of data will be collected. This includes quantitative performance metrics from the AI models themselves (such as detection accuracy, tracking precision, and re-identification success rates on benchmark datasets like MTMMC), system-level operational data (like processing speed for batch tasks, error logs), and qualitative user feedback (gathered through surveys, interviews, and observation of users interacting with the system during pilot phases or usability testing). We will know if the system is doing a good job by analyzing these collected data points against the predefined success criteria outlined at each level of the goal hierarchy. Specifically, success will be indicated by strong AI model performance on technical benchmarks, efficient system operation for analytical tasks, and demonstrable improvements in user task efficiency, coupled with positive user satisfaction regarding the system's effectiveness and usability in achieving their respective goals. Continuous monitoring of these aspects will guide further development and refinement.

5.3 Task Requirements Analysis Using AI Canvas

This section breaks down the core AI task of multi-camera person tracking using the AI Canvas framework to clarify requirements, inputs, outputs, and evaluation criteria.

5.3.1 AI Task Requirements

The AI components must meet the following requirements within the specified operational environment and constraints for retrospective analysis:

- **Requirements (REQ):** The fundamental goals the AI must achieve.
  - Accurately detect individuals in diverse video frames from multiple cameras sourced from recorded footage.
  - Reliably track detected individuals within the field of view of a single camera over time.

  - Correctly associate, or re-identify, the same individual when they appear across different camera views in the recorded data, potentially after time gaps or significant appearance changes.
  - Provide the spatial location of tracked individuals within a unified coordinate system (map).

- **Specifications (SPEC):** The necessary technical capabilities.
  - Implement high-performance person detection capabilities suitable for surveillance footage.
  - Employ robust tracking algorithms capable of handling short-term occlusions and maintaining identity within a single camera view.
  - Utilize methods to generate discriminative appearance features from person images, enabling effective re-identification.
  - Apply appropriate similarity metrics to compare appearance features for matching individuals across views.
  - Incorporate techniques for transforming image coordinates to a common spatial map reference frame using provided calibration data.

- **Environment (ENV):** The operational context, including assumptions and limitations.
  - Input consists of sequential image frames extracted from recorded video segments from a network of potentially non-overlapping cameras operating in real-world campus or factory settings.
  - The system is designed to operate under typical environmental challenges found in such recordings, but performance relies on certain assumptions and is subject to limitations:
    * **Appearance Consistency:** Assumes individuals do not drastically change their core appearance (e.g., changing distinct clothing) between consecutive camera views within the tracking period relevant to the analysis. Re-ID relies heavily on visual similarity.

    - **Lighting Variations:** While robust to moderate lighting changes, extreme variations present in the recordings can significantly degrade detection and Re-ID performance.
    - **Occlusion Handling:** Tolerant to short-term, partial occlusions. However, prolonged periods where an individual is completely hidden from all camera views in the footage may lead to track fragmentation or loss.
    - **Crowding:** Performance may degrade in extremely dense crowds where individuals are heavily occluded for extended durations.
    - **Viewpoint Changes:** Designed to handle viewpoint variations, but extreme differences can impact feature distinctiveness for Re-ID.
  - Processing occurs within the backend system, requiring adequate computational resources (potentially including GPUs) to handle the batch processing workload.

5.3.2 AI Canvas Development

Applying the AI Canvas framework helps to structure the analysis of the primary AI task: assigning a consistent global identity across multiple camera views from recorded footage. Figure 5.2 illustrates this.

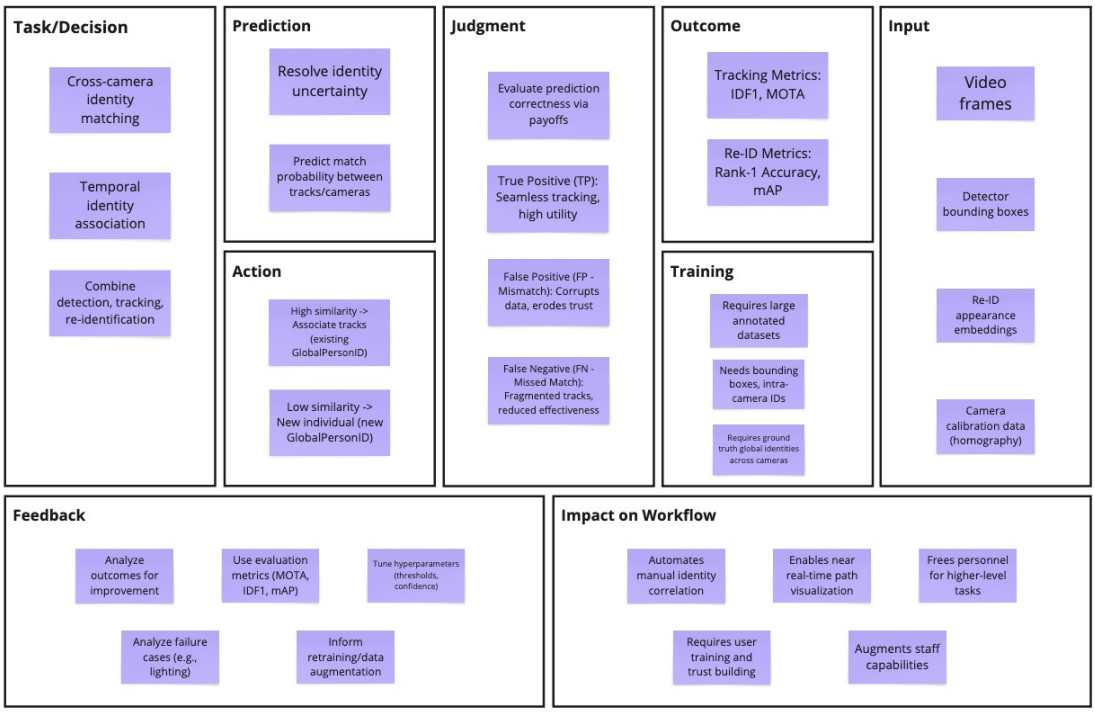

Figure 5.2: AI Canvas for Multi-Camera Person Tracking

- **Task/Decision Examined:** The core task is to determine if a person detected in one camera view at a certain time is the same individual as a person detected in another (or the same) camera view at a later time within the analyzed video data. This involves combining detection, tracking, and re-identification.
- **Prediction:** The key uncertainty the AI needs to resolve is: "Given appearance features (embedding) extracted from track A in camera X and features from track B in camera Y, what is the probability that A and B represent the same unique individual?"
- **Judgment:** Evaluating the correctness of the prediction involves considering the payoffs:
  - *Correct Match (True Positive):* Enables seamless tracking, accurate path reconstruction. Payoff: High system utility, user trust in analytical results.
  - *Incorrect Match (False Positive - Mismatch):* Assigns the same ID to different people. Payoff: Negative - Corrupts historical data, misleads users, significantly erodes trust.
  - *Missed Match (False Negative):* Fails to link the same person across views. Payoff: Negative - Creates fragmented tracks, reduces system effectiveness for analysis, may require manual reconciliation if possible.

  Minimizing false positive mismatches is often critical for maintaining data integrity and user confidence, even if it leads to slightly more missed matches (fragmented tracks).
- **Action:** Based on the prediction (similarity score and threshold):
  - If similarity is high (above threshold): Associate the tracks by assigning the existing GlobalPersonID.
  - If similarity is low (below threshold): Treat as a different individual, assign a new GlobalPersonID.
- **Outcome:** Performance is measured using standard metrics on benchmark datasets:

  - *Primary Tracking Metrics:* IDF1 Score (identity preservation), MOTA (overall tracking accuracy).
  - *Re-ID Specific Metrics:* Rank-1 Accuracy, mean Average Precision (mAP).
  - *System-Level Metrics:* Reduction in manual effort for historical analysis, time-to-completion for finding a person's full path in recorded data.

- **Training:** To train the AI components (especially detection and Re-ID models):
  - Need large datasets of annotated video sequences (like MTMMC train/val splits).
  - Annotations must include bounding boxes per frame and consistent person IDs *within* each camera view.
  - Crucially, need ground truth *global identities* linking person appearances across different cameras to train and evaluate the Re-ID model effectively.

- **Input (for Prediction during retrospective analysis):** Once trained, the system needs:
  - Incoming video frames extracted from sub-video segments fetched from cloud storage (e.g., S3).
  - Bounding boxes generated by the person detector.
  - Appearance embeddings generated by the Re-ID feature extractor for relevant detections/tracks.
  - Camera calibration data (e.g., homography matrices) for spatial mapping, specific to each camera.

- **Feedback:** Improving the AI relies on analyzing outcomes:
  - Use evaluation metrics (MOTA, IDF1, mAP, etc.) on test data to identify weaknesses.
  - Tune hyperparameters (e.g., Re-ID matching threshold, detection confidence) based on performance trade-offs.

  - Analyze failure cases to inform retraining strategies or data augmentation techniques.

- **Impact on Workflow:**
  - Automates the time-consuming and error-prone task of manually correlating identities across camera feeds in historical footage.
  - Enables visualization of movement paths across the entire monitored area from past recordings.
  - Frees up security/facility personnel to focus on higher-level tasks like incident investigation, trend analysis, and strategic planning rather than low-level video review.
  - Requires user training on the interface for retrospective analysis and building trust in the AI's associations. It augments staff capabilities for investigation and analysis.

Figure 5.3 illustrates the comprehensive ML Canvas design specifically tailored for the Intelligent Multi-Camera Person Tracking and Analytics System. This canvas provides a detailed blueprint for the ML components, building upon the general AI task analysis.

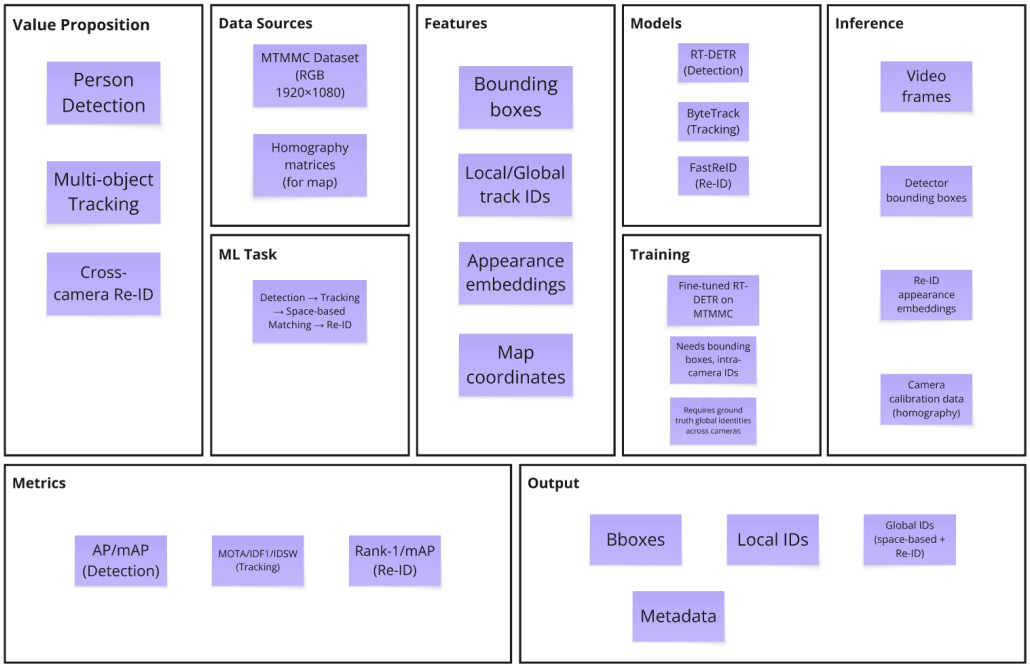

Figure 5.3: ML Canvas for Intelligent Multi-Camera Person Tracking and Analytics System

- **Problem Statement & Business Objectives (ML Canvas):** The system directly addresses the significant challenges security and facility personnel face in tracking individuals across complex, multi-camera environments using recorded data. It tackles the limitations of manual review—its labor intensity, error propensity, and fragmented views—and the core technical hurdle of maintaining consistent identity (Re-Identification) despite variations in appearance, lighting, occlusions, and camera viewpoints present in historical footage. The overarching business objectives are to enhance safety through better post-incident analysis, improve operational efficiency by automating tracking in recorded data, provide unified spatial visualization of movement, reduce incident investigation time, and optimize resource allocation based on analytics from past events.

- **Input Data Requirements (ML Canvas):** The necessary components for the system primarily involve recorded RGB video feeds (pre-split into sub-videos) or image sequences from multiple cameras, initially using datasets like MTMMC stored on cloud storage (e.g., AWS S3). These inputs encapsulate real-world challenges. Pre-calculated camera calibration data (Homography matrices) are essential inputs for spatial mapping, which is performed by the backend system using outputs from the ML pipeline.

- **Processing Approach & Model Selection (ML Canvas):** Central to the design is a backend-driven batch processing pipeline orchestrated by FastAPI. This pipeline integrates key AI steps in sequence:

  (A) Person Detection, utilizing RT-DETR (Real-Time DEtection TRansformer) for robust object localization.

  (B) Intra-Camera Tracking, employing ByteTrack algorithm to maintain temporary IDs within single camera views.

  (C) Space-Based Location Prediction, an algorithmic approach for overlapping camera views that converts 3D locations between cameras via 2D floor plan coordinates using homography to assign intra-frame global IDs.

(D) Feature Extraction for Re-ID, using FastReID to generate discriminative embeddings.

(E) Cross-Camera Re-Identification, which compares embeddings against a gallery (managed via Redis for recent/active data and TimescaleDB for historical data) to assign persistent GlobalPersonIDs for across-frame matching.

The ML pipeline provides track data (including pixel coordinates) which can then be used by the backend for spatial mapping using tools like OpenCV with pre-supplied camera calibration data. The justifications for model choices emphasize accuracy, robustness, and efficiency for retrospective analysis.

- **Expected Outputs (ML Canvas):** The system's outputs encompass both user-facing visualizations and stored data for analysis.
  - *For the UI (from backend metadata):* Annotated video feeds (client-side rendering of bounding boxes and GlobalPersonIDs based on WebSocket metadata) alongside a unified map interface showing current locations and historical paths (achieved by backend mapping of ML outputs).
  - *For historical analysis:* Structured tracking events (timestamp, IDs, coordinates including map coordinates derived by the backend, optional embeddings) stored in a TimescaleDB database.
- **Success Criteria & Evaluation Metrics (ML Canvas):** Evaluation primarily focuses on AI Model-Level Metrics through offline assessment:
  - *AI Model-Level Metrics (Offline Evaluation):* Specific metrics such as AP (Average Precision) for detection; MOTA (Multiple Object Tracking Accuracy), IDF1 (ID F1-Score), and IDSW (ID Switches) for intra-camera tracking; and Rank-1 accuracy, and mAP (mean Average Precision) for re-identification, using dedicated test sets from datasets like MTMMC. Spatial mapping accuracy can be measured by projection error if ground truth map coordinates are available.

Figure 5.4 visually summarizes the core AI processing stages detailed in the "Processing Approach & Model Selection (ML Canvas)" item. It outlines the sequential flow from Person Detection, Intra-Camera Tracking, and continuous Space-Based Matching, to conditional Cross-Camera Re-ID and Spatial Mapping.

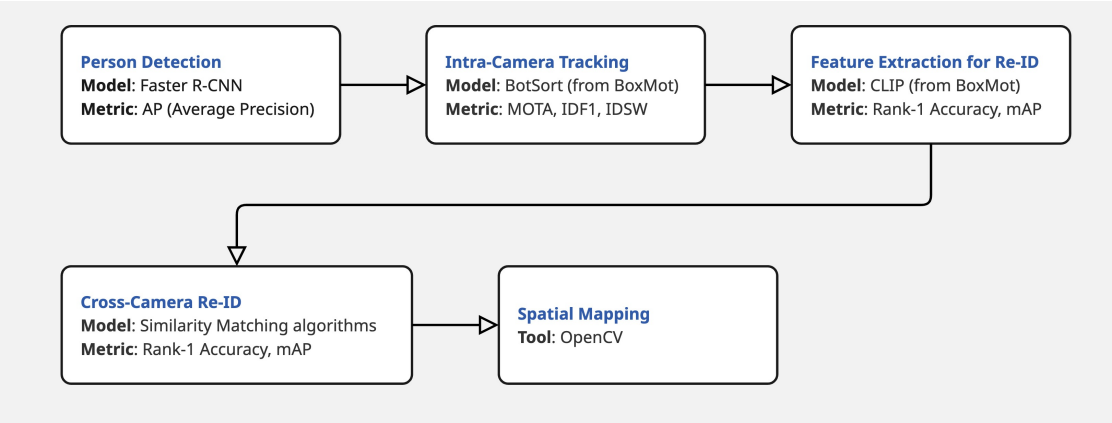

Figure 5.4: Core AI Processing Pipeline Stages, Models, and Metrics. This diagram illustrates the flow from Person Detection and Intra-Camera Tracking to continuous Space-Based Matching, followed by conditional Cross-Camera Re-ID, highlighting the primary model/tool and evaluation metrics for each stage.

This detailed ML Canvas provides a comprehensive blueprint for the development and evaluation of this intelligent surveillance analysis system.

5.3.3 (Optional) Innovation

While SpotOn builds upon established techniques in person detection, tracking, and re-identification, its creative contribution lies in how these components are conceptualized and orchestrated as a unified, system-level experience for multi-camera identity persistence. Rather than treating detection, tracking, and re-identification as isolated technical processes, SpotOn reframes continuous identity preservation across distributed cameras as a human-centered, integrated problem involving spatial consistency, temporal reasoning, and operational usability. This design-driven perspective shifts the focus from isolated model outputs to end-to-end identity continuity expressed through visual trajectories, spatial maps, and annotated views, enabling users to intuitively interpret complex surveillance data. As a result, the system creatively transforms established computer vision techniques into an investigation-oriented, visually grounded tracking experience tailored for real-world campus and factory environments.

5.4 AI Model Development and Evaluation

This section outlines the strategy for developing, training, and evaluating the core AI models that underpin the SpotOn system, focusing on the person detection component as a critical foundation.

5.4.1 Model Development and Training Strategy

The primary AI model for person detection is RT-DETR (Real-Time Detection Transformer), developed using the PyTorch deep learning framework. The development process prioritized robustness and accuracy for surveillance footage analysis. Key aspects of the strategy include:

- **Dataset Utilization:** The MTMMC dataset served as the primary data source for training and fine-tuning. This dataset's diversity in campus and factory environments, lighting conditions, and occlusion levels provides a realistic foundation for developing models intended for real-world scenarios.
- **Fine-Tuning Approach:** The RT-DETR model was initialized with default weights and fine-tuned on the MTMMC dataset. The model was trained using an NVIDIA RTX 4060 GPU to specifically address the 'person' class detection relevant to the system's objectives.
- **Data Preprocessing:** A standardized preprocessing pipeline was established. This included image resizing to a consistent input dimension for the model, color space conversion (e.g., BGR to RGB), normalization using established statistics, and conversion to tensor formats suitable for PyTorch. Data augmentation techniques, such as random horizontal flipping, were applied during training to enhance model generalization.
- **Experiment Management:** A systematic approach to tracking experiments, model configurations, hyperparameters, and performance metrics was employed, often facilitated by tools designed for experiment management (e.g., MLflow). This allowed for methodical comparison of different training runs and model variants.

This structured approach ensured that the model development lifecycle was organized, reproducible, and geared towards optimizing performance on the target task.

5.4.2 Testing and Validation Approach

Rigorous testing and validation were integral to ensure the reliability and effectiveness of the AI models. The approach encompassed several layers:

- **Quantitative Evaluation on Benchmark Data:** The primary method for assessing model performance involved testing against a dedicated

validation split of the MTMMC dataset. Standard COCO-style metrics were employed, particularly mean Average Precision (mAP) at various Intersection over Union (IoU) thresholds (e.g., mAP@.50, mAP@[.50:.95]). The Average Precision (AP) specifically for the 'person' class served as a critical performance indicator.

- **Comparative Model Analysis:** To select the most suitable baseline detection model for the system, a comparative analysis of different candidate architectures was performed. This involved training and evaluating several models (e.g., Faster R-CNN, RT-DETR variants) under consistent conditions. Performance was assessed based on quantitative accuracy metrics (like mAP), inference speed (Frames Per Second - FPS), and qualitative observations of detection stability on representative video data. The process of comparing models, including visualization of performance trade-offs (as conceptually illustrated in Figure 5.5) and tabular summaries of key metrics (conceptually represented by Figure 5.6), guided the selection.
- **Unit Testing of Core Components:** Key software components within the training and evaluation pipeline, such as dataset loading mechanisms, annotation parsing utilities, metric calculation functions, and model instantiation logic, were subjected to unit tests. This helped ensure the correctness and reliability of the underlying codebase supporting the AI model lifecycle.
- **Fairness Considerations:** An initial analysis of potential biases in the training data (MTMMC) and model behavior was considered. This involved identifying potential sources of bias (e.g., demographic imbalances, environmental variations) and discussing conceptual mitigation strategies (e.g., targeted data augmentation, fairness-aware training if feasible, transparency). While a full quantitative fairness audit requires specific demographic ground truth often unavailable, this proactive consideration is crucial for responsible AI development.

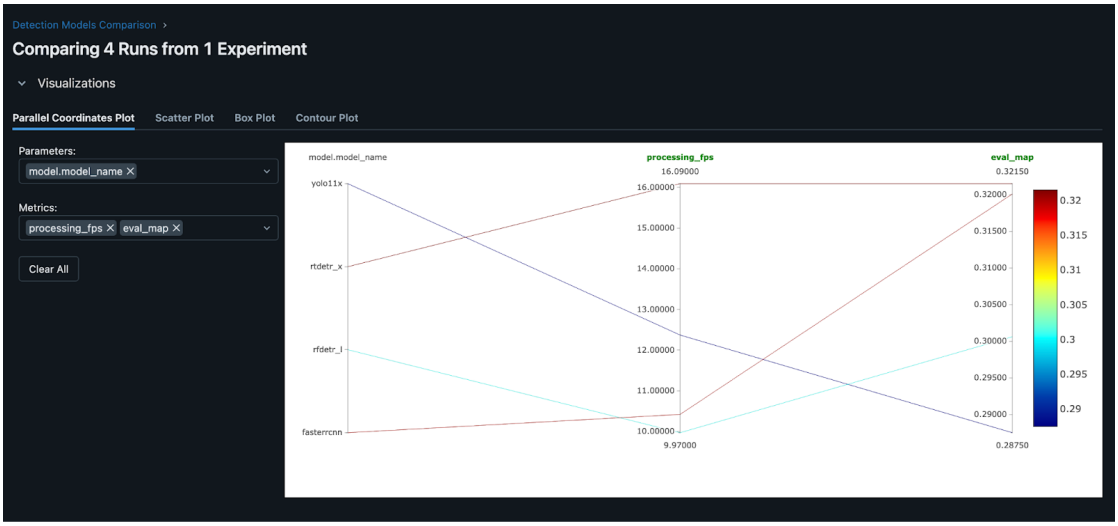

Figure 5.5: Conceptual Illustration: Chart Comparing Performance Metrics of Different Detection Models

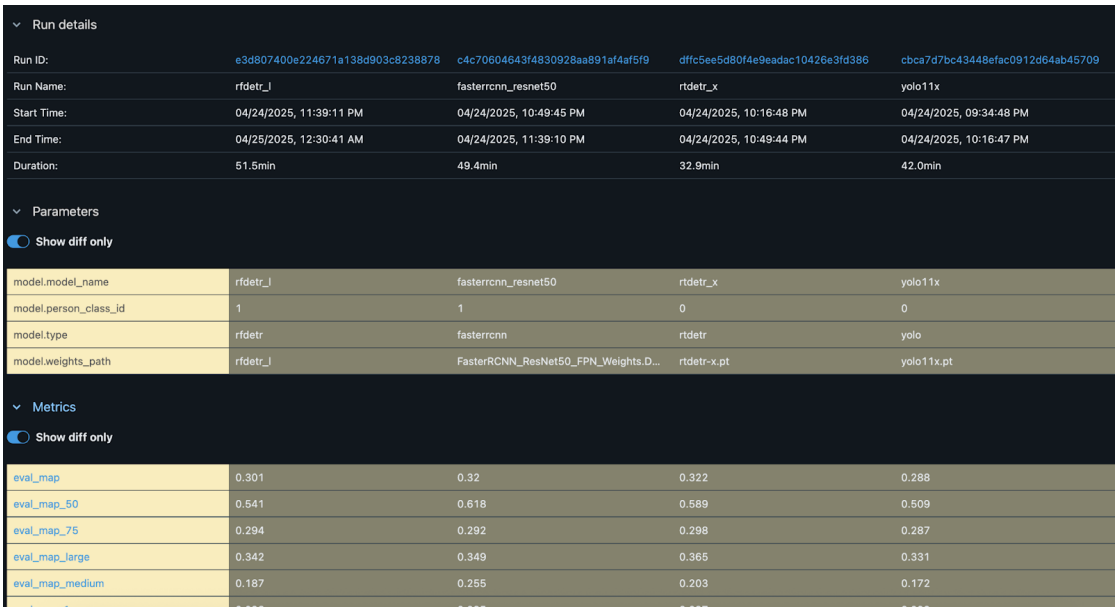

Figure 5.6: Conceptual Illustration: Table Summarizing Key Performance Metrics for Model Comparison

5.4.3 Performance Results and Justification

The comparative analysis of various person detection models, aided by experiment tracking tools, led to the selection of RT-DETR as the baseline architecture for the SpotOn system. After evaluating candidates including Faster R-CNN and RT-DETR variants, RT-DETR presented a compelling balance of characteristics crucial for the system's application in retrospective surveillance analysis.

Key performance observations and the justification for selecting RT-DETR include:

- **End-to-End Detection:** RT-DETR provides end-to-end detection without requiring Non-Maximal Suppression (NMS) post-processing, reducing the complexity of the detection pipeline and eliminating NMS-related artifacts.
- **Strong Overall Accuracy:** RT-DETR demonstrated competitive mAP scores on the MTMMC validation set, particularly for medium to large-sized persons common in surveillance footage.
- **Flexible Speed-Accuracy Trade-off:** The transformer-based architecture allows for flexible speed tuning via decoder layers, enabling optimization for the batch processing pipeline.
- **Robust Feature Learning:** The hybrid encoder architecture enables robust feature extraction across varying lighting conditions and partial occlusions present in the MTMMC dataset.

5.4.4 Demonstration on Representative Data (Model Explainability)

To build trust and enable diagnostics, techniques were employed to understand and explain the behavior of the selected person detection model (RT-DETR).

Visual Explanation of Model Focus: Gradient-weighted Class Activation Mapping (Grad-CAM) was utilized to generate visual heatmaps. These heatmaps highlight the regions in an input image that were most

influential in the model's decision to classify an area as containing a 'person'. Figure 5.7 shows such a visualization for an image representative of a factory scene. The heatmap often strongly highlights relevant body parts like the torso and upper legs, aligning well with the subject's silhouette and indicating that the model learned to focus on pertinent visual features for its high-confidence detection. Similar interpretability was observed for campus scenes. For instance, Figure 5.8 and Figure 5.9 depict Grad-CAM outputs for persons detected in different campus environments. Even for smaller or more distant individuals, the heatmaps tend to concentrate on the upper body or head regions, demonstrating the model's ability to identify key features contributing to high-confidence scores. These visualizations confirm that the model generally behaves as expected by focusing on semantically relevant parts of a person.

Explainability is a critical factor for trust and adoption in AI-driven surveillance systems. In SpotOn, techniques such as Grad-CAM visualization and interpretable performance metrics provide insight into model behavior and decision-making processes. The team will systematically **communicate evaluation, quality, and risk results** to advisors, ensuring all architectural choices are **evidence-based decisions**. These mechanisms enable operators and system maintainers to understand, validate, and diagnose AI outputs, thereby increasing confidence in automated identity association and supporting responsible real-world deployment.

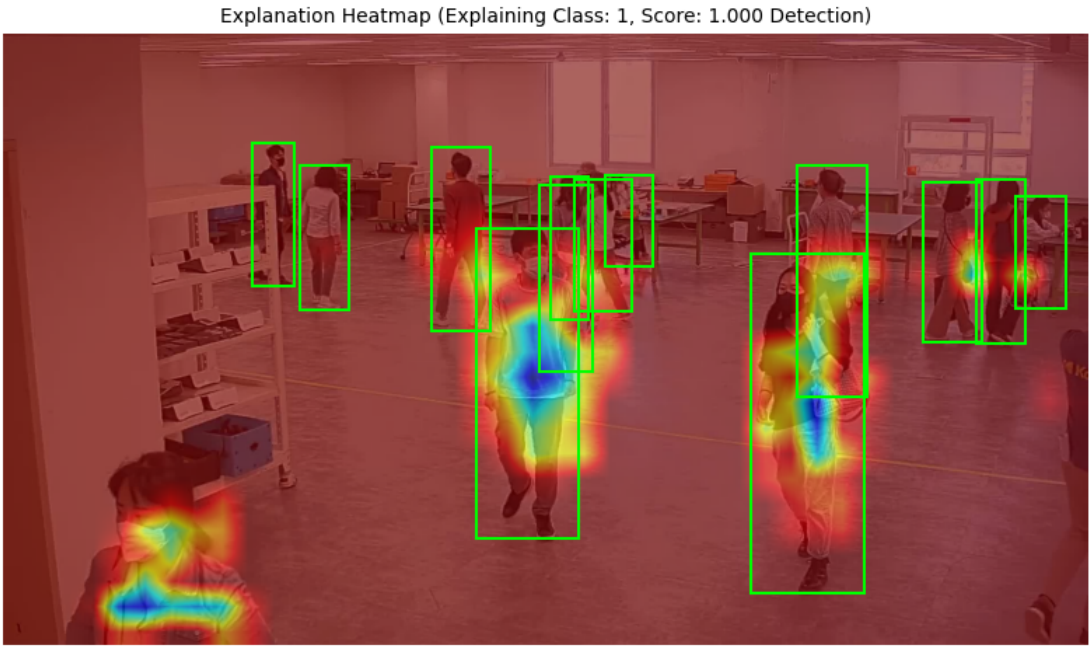

Figure 5.7: Grad-CAM Visualization for a Person Detected in a Factory Scene Image

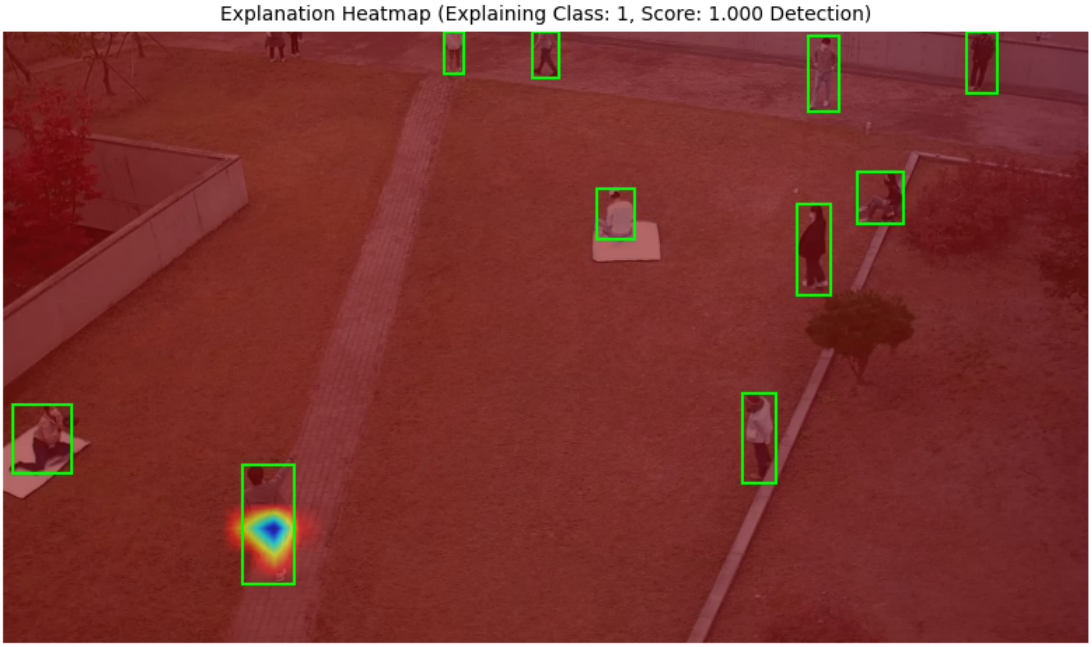

Figure 5.8: Grad-CAM Visualization for a Person Detected in a Campus Scene Image (View 1)

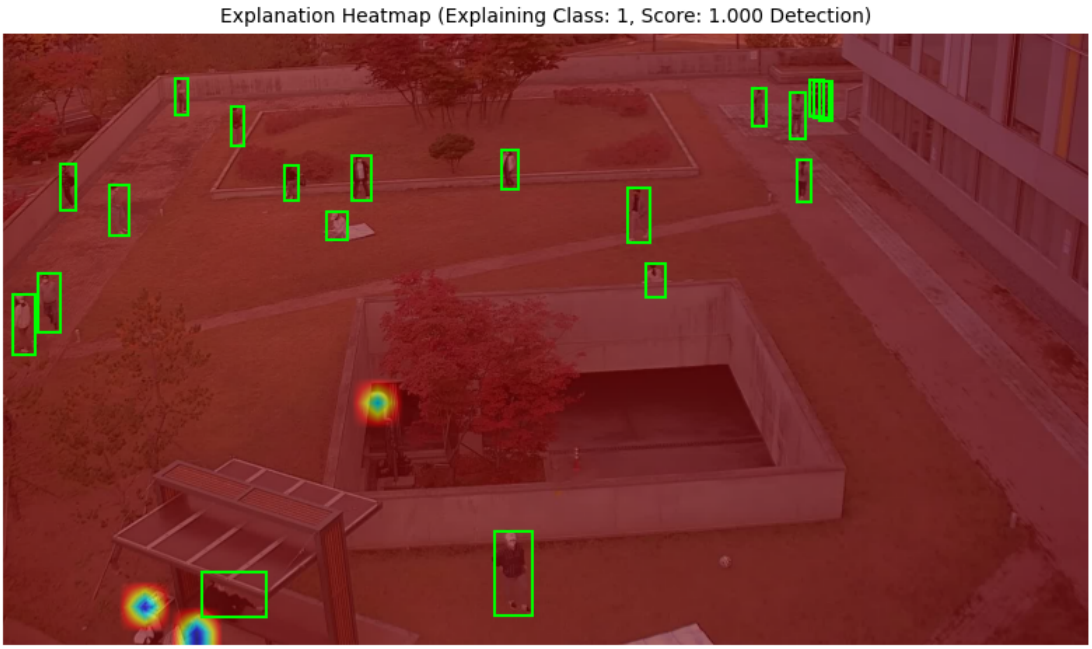

Figure 5.9: Grad-CAM Visualization for a Person Detected in a Campus Scene Image (View 2)

Textual Reasoning for Predictions: Beyond visual heatmaps, a mechanism was designed to generate human-understandable textual reasoning for individual predictions. This reasoning typically combines the model's quantitative output (confidence score) with a qualitative interpretation and a reference to the visual explanation. For example, for a high-confidence detection, the system might generate reasoning such as: "Detected object classified as 'person' with a high confidence score (e.g., 0.99 or higher). This suggests the model found strong visual evidence matching features learned for the 'person' class. The associated visual explanation (e.g., Grad-CAM focus) highlights the specific image regions that most influenced this classification decision. Reviewing this visual explanation can provide further insight into which parts of the object (e.g., head, torso) contributed most to the prediction." This approach provides a practical level of interpretability, allowing users to understand the basis of individual detection decisions by linking the model's confidence to visual evidence of its focus. These demonstrations of model explainability serve as valuable tools for verifying expected model behavior, debugging potential issues where the model might focus on irrelevant features, and ultimately fostering greater trust in the AI system's outputs.

The AI models integrated into SpotOn are selected with a strong emphasis on deployment readiness and operational stability. Well-established architectures such as RT-DETR, ByteTrack, and FastReID are employed to ensure robustness, maintainability, and compatibility with standard deep learning frameworks. The system is designed to operate under realistic hardware constraints and prioritizes batch-based retrospective analysis to avoid fragile real-time assumptions. Together with model versioning, performance monitoring, and retraining considerations, these design choices support reliable long-term use in real-world surveillance environments.

5.5 User Experience Design with AI

Figure 5.10: AI-Annotated Multi-Camera View with Person Detection.

The system leverages Artificial Intelligence primarily through an Annotate interaction style to enhance the user's ability to monitor and analyze activity across multiple camera views. In this specific mockup, the AI processes the video streams from each active camera to perform person detection. When the AI identifies a person within a camera's field of view, it annotates the visual feed by drawing a bounding box (seen as green rectangles around individuals in the composite camera display) around that person. This direct visual annotation serves multiple purposes: it immediately draws the operator's attention to the presence and specific location of individuals, helps in quickly assessing the number of people in an area, and provides a clear visual marker for each detected person. By automatically highlighting these individuals, the AI significantly reduces the cognitive load on the human operator, who no longer needs to manually scan complex scenes. These bounding boxes are fundamental AI-generated annotations that support rapid visual assessment and are a foundational element for more complex tasks like person tracking and

re-identification across different cameras, which are also hinted at by the map visualization and person ID information on the right side of the interface.

Figure 5.11: AI-Generated Trajectory Map for Multi-Person Tracking.

The system utilizes Artificial Intelligence in an Annotate interaction style to provide users with a comprehensive spatial understanding of movement patterns within the monitored area, as depicted in the map visualization. The AI is responsible for several complex tasks that culminate in this annotated map display. Firstly, AI models detect individuals in various camera feeds (as seen on the left). Crucially, the AI then performs cross-camera re-identification, assigning a persistent global ID to each individual and tracking them as they move between different camera views, even if those views don't overlap or if there are temporal gaps. Simultaneously, the AI uses homography or other spatial mapping techniques to transform the 2D pixel coordinates of a detected person in a camera image

into corresponding coordinates on a unified 2D floor plan or map of the environment. The map display itself is an AI-driven annotation. The lines and markers on the map are not manually drawn but are generated by the AI, representing the historical and current paths of tracked individuals. Each distinct color typically corresponds to a unique global person ID, allowing operators to visually differentiate and follow the movements of multiple people simultaneously. The location markers (pins) indicate the last known positions or significant points along these paths. This annotated map provides powerful insights into pedestrian flow, area usage, and individual trajectories, all derived from the AI's continuous processing and interpretation of multi-camera surveillance data. It transforms raw video into a spatially coherent and easily digestible overview of activity.

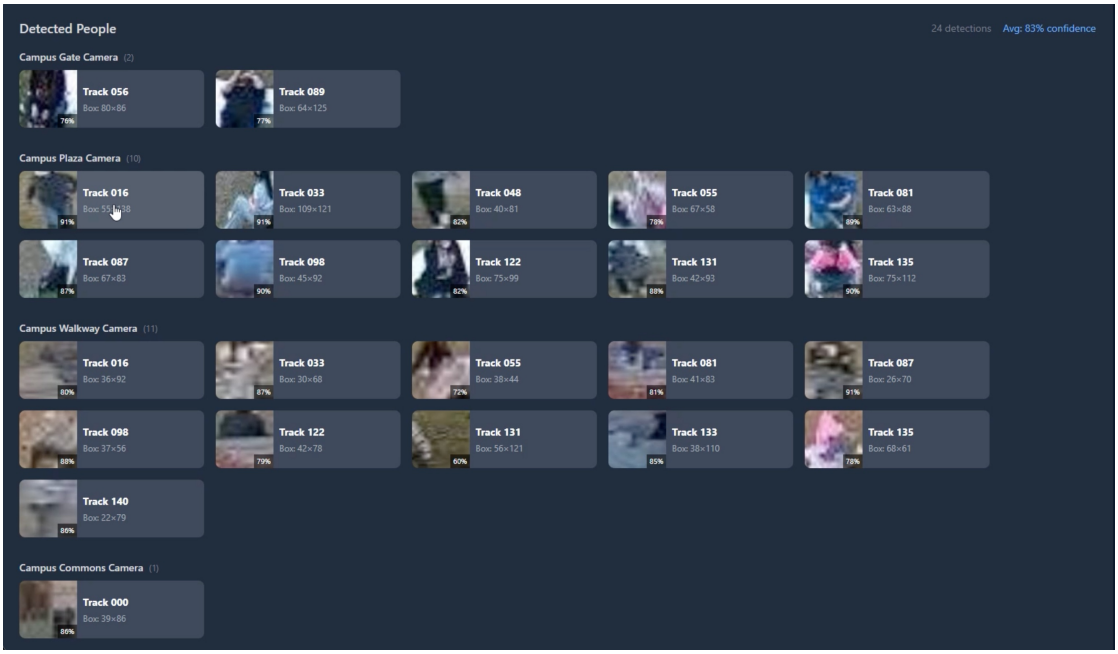

Figure 5.12: Detailed AI-Annotated Tracking Information and Movement Analytics.

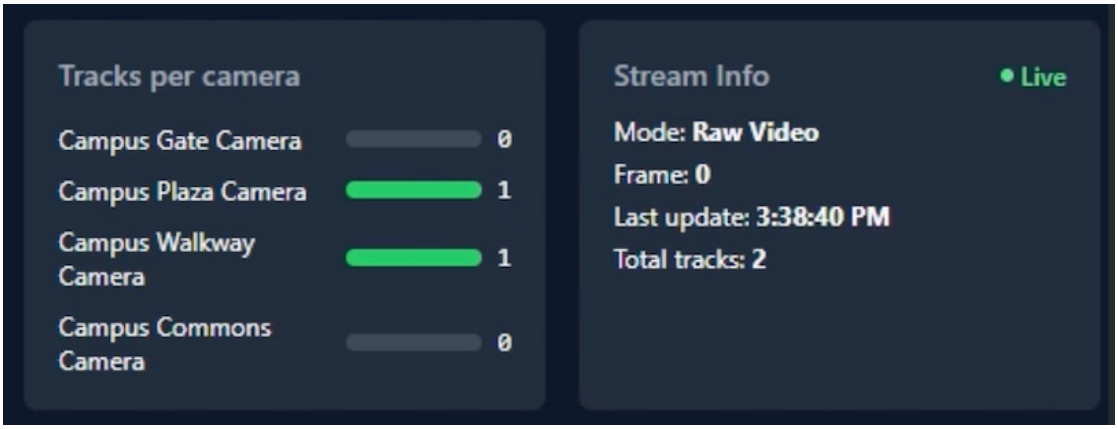

Figure 5.13: AI-Extracted Metadata and Re-Identification Features.

The system further enhances understanding through detailed metadata extraction, as illustrated in Figure 5.13. The AI analyses the visual appearance of detected individuals to extract attribute metadata such as **upper body clothing color** (e.g., "black") and **lower body clothing color** (e.g., "blue"). These attributes, along with the person's estimated **gender** (e.g., "Male"), are automatically cataloged and associated with the tracked identity. This granular level of detail supports advanced query capabilities, allowing operators to search for individuals based on physical descriptions (e.g., "Show all males wearing black shirts"). This feature relies on the AI's ability to not only detect and track but also to semantically understand the visual content within the bounding boxes.

5.6 Deployment Strategy

5.6.1 Deployment Plan

- **Deployment Location:** The AI components (e.g., person detection using RT-DETR, intra-camera tracking with ByteTrack, space-based location prediction algorithm for overlapping views, Re-ID feature extraction with FastReID) are integrated within the system's core **Backend** service. This Backend service, built with FastAPI, along with the entire system infrastructure (Frontend server, databases, caching, reverse proxy), is designed to be deployed in a **Cloud** environment.

This leverages cloud scalability for computational resources and centralized data storage.

- **AI Communication with System Components:** The AI models and algorithms run embedded within the Python-based FastAPI Backend process. Communication occurs as follows:
  - **Internal Calls:** The main FastAPI application logic directly invokes the AI models for detection, tracking, space-based matching, feature extraction, and homography transformations using standard Python function calls.
  - **Backend to Data Storage/Cache:** The Backend interacts with Redis (for caching Re-ID embeddings/gallery) and TimescaleDB/PostgreSQL (for storing historical tracking events and persistent embeddings) through database client libraries. It fetches video segments from cloud storage (e.g., AWS S3) and caches them locally for processing.
  - **Backend to Frontend (Real-time Video & Metadata):**
    - **MJPEG Streaming:** The backend streams raw video frames directly to the frontend using the **MJPEG** protocol. This ensures low-latency video playback without requiring intermediate cloud storage for live feeds.
    - **WebSocket:** Processed tracking metadata (bounding boxes, local IDs, global IDs from space-based and Re-ID matching, map coordinates, frame timestamps) are pushed from the FastAPI Backend to the React Frontend using **WebSocket** connections. This enables real-time updates for client-side visualization overlays on top of the MJPEG video stream.
  - **Frontend to Backend (Control/Queries):** User interactions (e.g., initiating analysis tasks, requesting historical data, system configuration) are sent from the React Frontend to the FastAPI Backend via **RESTful APIs**. External communication passes through a reverse proxy like Nginx.
- **Tools and Frameworks Used:** The deployment relies on the following key technologies:

  - *Containerization & Orchestration:* **Docker** for containerizing all services. **Docker Compose** for orchestration.
  - *Backend Framework:* **FastAPI** for REST APIs, WebSocket handling, and orchestrating the AI processing pipeline.
  - *AI Models & Libraries:* **PyTorch** as the deep learning framework. Specific models like **RT-DETR** (detection), **ByteTrack** (tracking), **FastReID** (Re-ID feature extraction). Space-based location prediction uses homography transformations. **OpenCV** for video processing and image manipulation.
  - *Frontend Framework:* **React** for the user interface.
  - *Data Storage & Caching:* Cloud storage like **AWS S3** for video segments. **Redis** for caching. **TimescaleDB** for historical tracking data and embeddings.
  - *Reverse Proxy:* **Nginx**.
  - *Monitoring & Logging:* Tools like Prometheus, Loki, Grafana.
- **System Qualities (Reliability, Security, Maintainability, Scalability):** The deployment strategy addresses system qualities as follows:
  - *Reliability:* Orchestration tools provide container health management and restarts. Monitoring tools offer visibility.
  - *Security:* Nginx acts as a gateway for security measures (HTTPS, etc.). Containerization provides isolation. Proper cloud security configurations (IAM roles, network policies) are essential.
  - *Maintainability:* Docker ensures consistent environments. Modular backend services and clear API definitions aid management. Centralized logging/metrics simplify troubleshooting.
  - *Scalability:* Cloud deployment allows resource scaling. Stateless FastAPI backend instances can be scaled horizontally. Databases and caches can be scaled using cloud services or native features. The processing pipeline is designed for batch analysis of pre-split video segments.
- **MLOps & Governance:**

– **Observability:** The system includes production-grade **observability and monitoring** to track model accuracy and concept drift.
– **Risk Management:** Deployment includes **guardrails** for low-confidence detections and **incident playbooks** for handling tracking failures.

5.6.2 Proof of Concept (Service Demonstration)

The Proof of Concept (PoC) for the SpotOn system focused on demonstrating the end-to-end functionality of the AI-driven backend service, particularly its capability to process video data and stream tracking metadata for retrospective analysis.

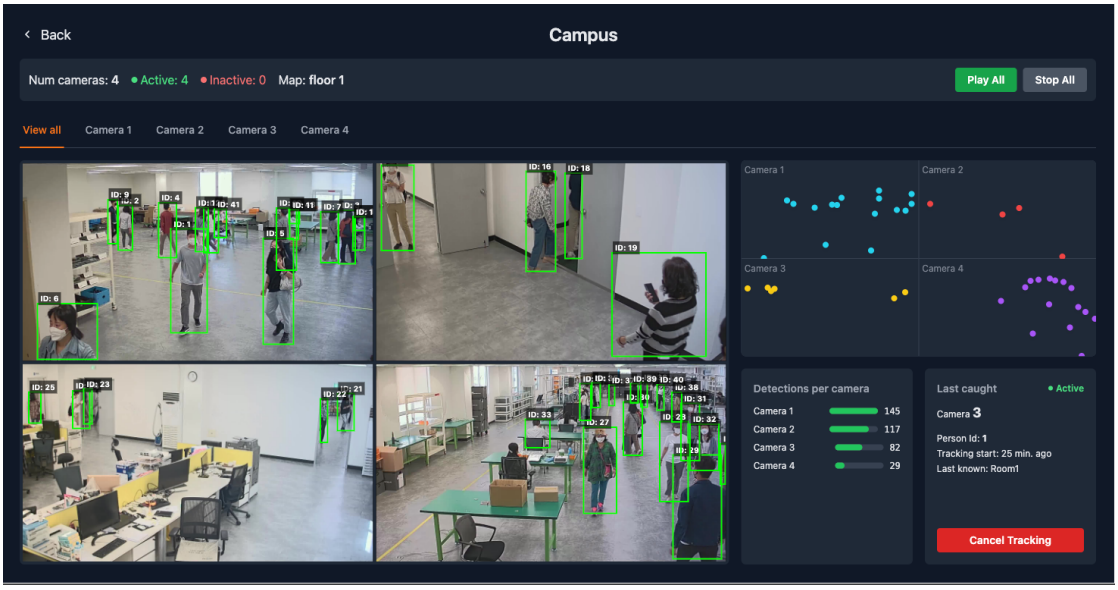

Figure 5.14: Frontend Proof of Concept

The PoC involved the following key steps:

1. **Backend Deployment:** The backend application, containerized using Docker and orchestrated with Docker Compose (including dependencies like Redis and TimescaleDB), was started. This made the system's REST APIs and WebSocket endpoint accessible.

2. **Task Initiation:** A client (simulated by a test script) initiated a retrospective processing task by sending a POST request to the backend's REST API (e.g., '/api/v1/processing-tasks/start'). This request specified the target environment (e.g., "factory" or "campus") for analysis. The backend responded with a unique task ID and the WebSocket URL for tracking updates.
3. **WebSocket Connection:** The client then established a WebSocket connection to the provided URL to receive real-time metadata streams.
4. **Data Processing and Streaming Observation:** As the backend processed the video segments for the specified environment (fetching from S3, extracting frames, performing detection, tracking, Re-ID, and homography).
5. **Client-Side Interpretation:** The test client logged these incoming messages, demonstrating that the backend correctly performed the AI pipeline (detection, tracking, Re-ID, homography) and formatted the results into the specified JSON structure for WebSocket delivery.

This PoC successfully validated that the backend service could manage a processing task, execute the AI pipeline on video data, and deliver structured, real-time tracking metadata suitable for a frontend application to consume for visualization (e.g., rendering bounding boxes on video playback and plotting trajectories on a map).

5.7 (Optional) Reflection and Future Development

This section is designated for a comprehensive reflection on the project's development process, outcomes, and lessons learned, alongside an outline of potential avenues for future development and enhancements. The content for this section will be thoughtfully prepared and included upon the final completion of the project.

Chapter 6

Software Development

6.1 Software Development Methodology

Project management and version control are handled via **GitHub Issues and PRs** to ensure transparency. Algorithm tuning will rely on strict **data and quality metrics**, and the team will improve the system through formal **AAR-style (After-Action Review) reflections** at the end of each development sprint.

6.2 Technology Stack

6.3 Coding Standards

6.4 Progress Tracking Report

Chapter 7

Deliverable

7.1 Software Solution

Soon.

7.2 Test Report

Soon.

Chapter 8

Conclusion and Discussion

Soon.

Reference

Bibliography

[1] X. Wang, S. Zhang, L. Qing, Z. Liu, and Y. Gao, "Multi-camera tracking by multi-task assignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3891–3905, 2021.

[2] Y. Zhang, P. *Sun*[2], Y. Jiang[3], D. *Yu*[3], F. Weng[1], Z. *Yuan*[3], P. *Luo*[2], W. *Liu*[1], and X. Wang[1], "Bytetrack: Multi-object tracking by associating every detection box," *arXiv preprint arXiv:2110.06864*, 2022.

[3] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, "Fastreid: A pytorch toolbox for general instance re-identification," *arXiv preprint arXiv:2006.02631*, 2020.

[4] S. Woo, K. Park, I. Shin, M. Kim, and I. S. Kweon, "Mtmmc: A large-scale real-world multi-modal camera tracking benchmark," *arXiv preprint arXiv:2403.20225*, 2024.

[5] L. Bredereke, Y. Hartmann, and T. Schultz, "A modular pipeline for 3d object tracking using rgb cameras," *arXiv preprint arXiv:2503.04322*, 2025.

[6] V. Gautam, S. Prasad, and S. Sinha, "Yolore-idnet: An efficient multi-camera system for person-tracking," *arXiv preprint arXiv:2309.13387*, 2023.

[7] C.-C. Wang, C.-Y. Yang, R. Jain, M.-C. Hu, and W.-H. Cheng, "Tracking with mixture of realistic and synthetic knowledge," *CVPR Workshops*, 2023.

[8] Z. Wu, S. Dong, and H. Tian, "Online tracking with geometric consistency and state-aware re-id," *CVPR Workshops*, 2024.

[9] R. Hachiuma, T. Nakayama, and H. Saito, “Trajectory-based tracking in the operating room,” *IEEE Access*, 2022.

Appendix

Soon.