HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

Multi-Camera Tracking for Employee Behavior Monitoring

TRẦN QUỐC LẬP
lap.tq194443@sis.hust.edu.vn

Major: Data Science
Specialization: Data Science

Supervisor: Ph.D. Nguyễn Nhật Quang _____
Department: Computer Science
School: School of Information and Communications Technology

Co-Supervisor: Phạm Công Mạnh _____
Department: Research and Development
Company: *The company name is hidden from public*

HANOI, 08/2023

ACKNOWLEDGMENT

ABSTRACT

Multi-camera tracking (MCT) aims to associate people across different cameras. Existing solutions primarily rely on Person Re-Identification (Re-ID) to match people's visual features. However, this approach fails to track people with similar appearances, such as employees wearing uniforms in workplaces. This thesis investigates spatio-temporal association (STA) as an alternative solution that does not use visual cues.

The proposed method matches individuals across time-synchronized cameras with overlapping views by projecting and comparing their foot point locations using homography. Extensions are incrementally added to address limitations: FP filtering and window-based mapping to handle missing detections, and pose estimation to improve foot point accuracy. Various experiments have been conducted to provide deep insights into the proposed solution's underlying issues and applicable scenarios. An application software system is implemented to demonstrate real applicability of the method in monitoring employee behavior.

The key contributions are: (1) A novel spatio-temporal association solution overcoming limitations of appearance-based methods; (2) Analysis of when (under what conditions) the proposed MCT method works and when it does not; (3) An application system meeting practical requirements. Overall, this thesis opens new possibilities for multi-camera tracking research by demonstrating the potential of solely using spatio-temporal information. Future work can build upon these findings to enhance robustness and broaden applicability.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
FN	False negative
FOV	Field of view
FP	False positive
IQR	Inter-quantile range
MCT	Multi-Camera Tracking
Re-ID	Person Re-identification
SCT	Single Camera Tracking
STA	Spatial-Temporal Association
TP	True positive

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

In today's life, *multi-camera systems* play an important role in our safety and security, as well as providing valuable insights into our world. Multi-camera systems consist of multiple cameras installed in various locations within a certain space. With their widespread use in homes, businesses, schools, public spaces, and more, multi-camera systems have become an integral part of our daily lives.

In the past, multi-camera systems were mainly used to ensure security and record past events for future reference. Nowadays, with the development of technology, one of the interesting potentials of multi-camera systems is *people tracking*, which offers many other benefits. For example, in retail stores, they can track customers' movements and interactions with products to help owners understand which products are popular and where they should be displayed. In sports, multi-camera systems can analyze players' performance and tactics to support coaches plan personalized training programs. Multi-camera systems can also perform important real-time tasks such as issuing alerts during unusual events in public places. To associate (i.e., re-identify) people moving across different cameras, most multi-camera systems today base their solutions on visual features similarity.

Therefore, particularly in management, managers have recognized the potential benefits of multi-camera systems and are urgently finding ways to apply them in companies, stores, etc. to track their employees across different workplaces. By tracking employees, managers expect from multi-camera systems the following application scenarios:

- Calculating employees' productivity.

Figure 1.1: An example of multi-camera tracking in retails to track customers' movements and interactions with products.

- Monitoring employees' movements and sending immediate alerts to managers if they have an accident, disappear for too long or enter an unauthorized area.
- Assigning work among employees in a balanced way and coordinating work automatically.

However, there are some unique aspects in the management field that make it more difficult to apply multi-camera tracking systems than other fields. While popular methods rely on appearance similarity to associate individuals across cameras, most employees look very similar because they wear uniforms in their workplace, causing the multi-camera tracking system using visual features similarity to easily match people incorrectly (see Figure 1.2). In this thesis project work, I **1)** develop a solution to track people in multiple cameras which can work well when people wear uniforms or have similar appearance. **2)** conduct experiments to verify research hypotheses that show when (under what conditions) the proposed solution work. **3)** develop an application system that shows the practical application of the proposed solution to the problem of monitoring employees' work behavior.

1.2 Background and Problems of Research

To meet the given requirements, the research problem to be solved is *Multi-Camera Tracking* (MCT), one of the sub-problems of *Computer Vision*. Many conferences dedicated to MCT have been held annually, such as the AI City Challenge

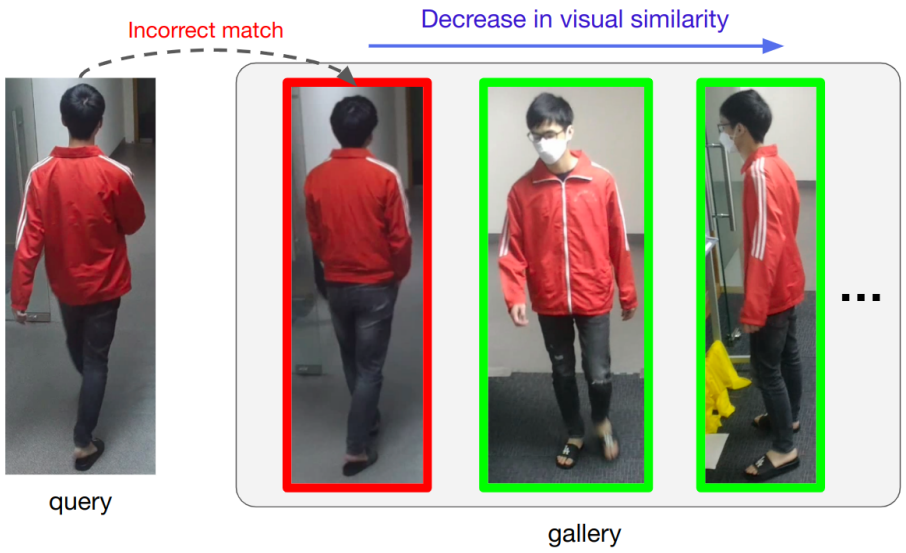


Figure 1.2: An example of people with similar looks that poses a challenge to Re-ID application. The top individuals in the gallery who are most similar to the query person are shown from left to right in decreasing order of similarity. The images with blue border belong to the same person as the query. The images with red border belong to different people than the query.

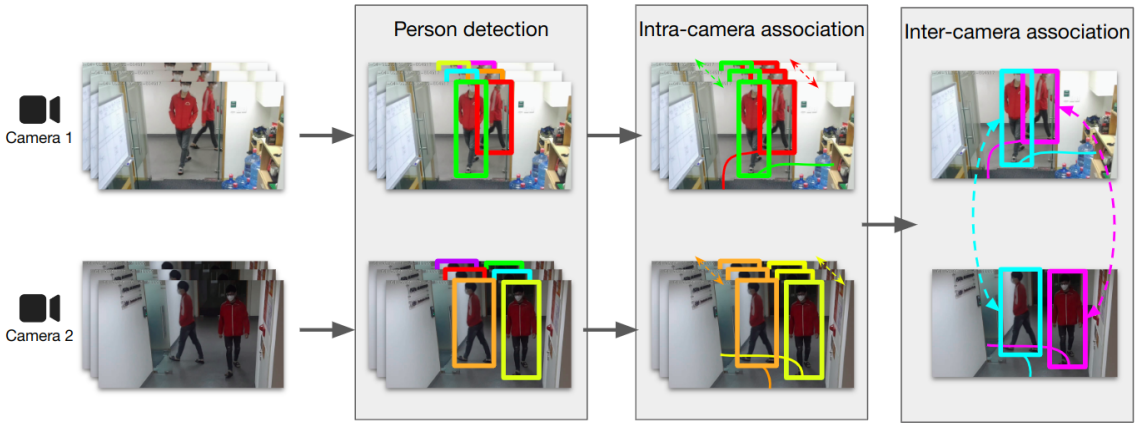


Figure 1.3: General pipeline for multi-camera tracking.

and MMP-Tracking Challenge, to encourage progress in this field. The MCT problem takes videos captured from multiple cameras as input and outputs the person detection in those videos, as well as determining which people are of the same identity across the videos. The widely-used pipeline for MCT involves three main steps: person detection, intra-camera association, and inter-camera association (as illustrated in Figure 1.3).

Person detection belongs to mid-level Computer Vision. With the raise of Deep Learning, it has been one of the active research areas in recent years. Many deep learning models have been introduced such as R-CNN family [13, 12, 25], YOLO family [24, 22, 23, 27], etc. that have achieved impressive performance, not only with high accuracy but also with real-time processing speed. The input of these

models is an image, and the output is a set of rectangle bounding boxes.

The next step in the MCT pipeline is *intra-camera association*, which involves associating detections through consecutive video frames within each camera and grouping detections of the same person into one track. Several popular methods, including SORT [3], ByteTrack [30], and DeepSORT [28], have been developed to accomplish this task. To associate detection between consecutive video frames, these methods generally compare the closeness between bounding boxes of the current frame and the previous in terms of either positions or visual appearance. The combination of person detection step and intra-camera association step is called *Single Camera Tracking* (SCT).

The final step in MCT is *inter-camera association*, which involves grouping tracks of the same person across multiple cameras. The cameras may have different time synchronization and overlapping features. Most current methods for inter-camera association rely on *Person Re-identification* (Re-ID) which involves finding a query person in a gallery of images captured from multiple cameras. A neural network is used to extract visual embeddings from query images and match them with visual embeddings of the same person in the gallery.

However, those methods still have some limitations that have yet to be fully resolved. Regarding person detection, deep learning models may miss or detect a person with low confidence if they are occluded by an object or by another person. For intra-camera association, ID switch is a common problem with tracking methods such as SORT [3], DeepSORT [28], and ByteTrack [30], mainly due to missing detections or when people move close to each other optically. In inter-camera association, common issues like viewpoint variation, deformation, partial occlusion, and illumination may cause the visual features of two tracks belonging to the same person to have a large distance, while the visual features of two different people may have a small distance, resulting in incorrect matching (Figure 1.2). Especially, in the application scenario of this thesis, the fact that employees wear similar uniforms has posed a challenge for visual-based Re-ID.

As mentioned earlier, this thesis focuses on tracking employees wearing similar uniforms, where similar visual appearance poses a challenge for commonly used inter-camera association methods. Thus, the thesis's focus is on solving the inter-camera association problem, which is step 3 in the pipeline diagram. Instead of relying on visual feature appearance like Re-ID methods, this project will develop a solution using *spatio-temporal information* of people's trajectory in overlapping field of views (FOV) (see Figure 1.4).

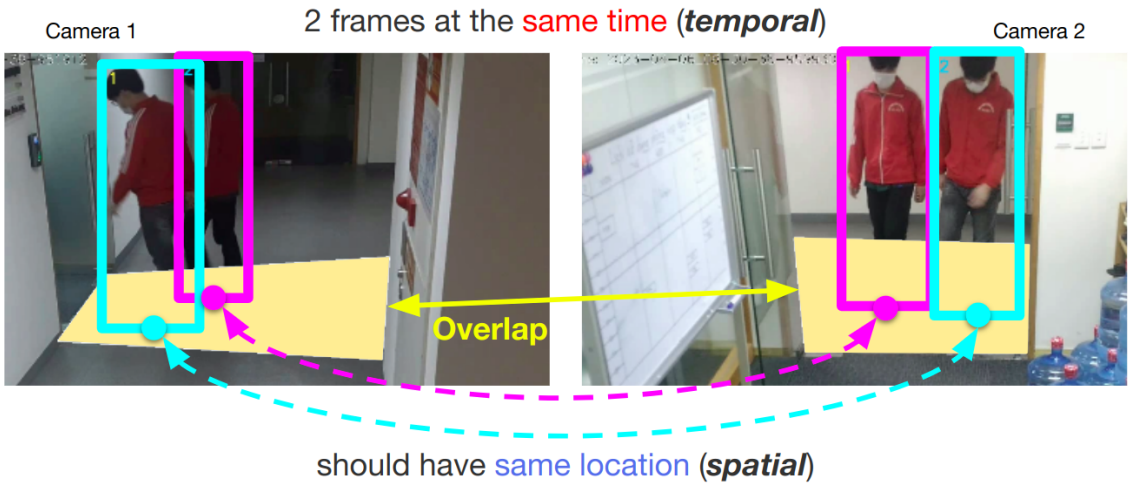


Figure 1.4: Multi-camera tracking with overlapping field of views.

1.3 Objectives and Requirements

My thesis work focus on three main goals:

1. **Research**: Propose a solution that uses spatio-temporal information to map tracks between cameras (step 3 in Figure 1.3) as a replacement for visual feature-based Re-ID, where cameras have overlapping FOV and synchronized time.
2. **Technology**: Master the software frameworks and development processes of deep learning models, from data collection, data preprocessing, model training to conducting experiments.
3. **Application**: Develop a software system that implements the proposed spatio-temporal MCT solution to support monitoring employees' behavior (where they are and for how long) in the workplace, thereby showing the applicability of the proposed spatio-temporal MCT solution.

1.4 Scope of Work

The work conducted within the scope of this thesis project includes:

1. Developing a solution based on STA to match people between overlapping cameras to replace methods based on visual feature Re-ID.
2. Cameras have small overlapping FOV and synchronized time.
3. The subjects to be tracked are employees wearing the same uniform in a store.
4. Employee behavior: Staying in designated work position or moving between

multiple cameras at a normal pace.

5. The solution needs to support a maximum of 4 people in the overlapping FOV at any given time.
6. The implemented system should support a test scenario that comprises 3 cameras covering all areas of the store.

1.5 Organization of Thesis

This thesis is organized into 5 chapters:

Chapter 1 introduces the background, motivation, objectives, and scope for the research. It highlights the challenges of using visual appearance features for multi-camera tracking of employees wearing uniforms. The goals are set to propose and evaluate a spatio-temporal association solution as a replacement.

Chapter 2 reviews related work on person re-identification, spatio-temporal association, image warping, and essential technologies that will be utilized. This provides the theoretical foundation for the research.

Chapter 3 presents the proposed spatio-temporal association method in detail. It starts with a baseline version and incrementally improves the solution through extensions. Each extension is thoroughly analyzed to understand when and why it works or fails. The method is evaluated on a custom dataset and compared with a visual feature-based Re-ID approach.

Chapter 4 describes the development of a software application to showcase the practical applicability of the research solution. It focuses on designing features related to monitoring employee productivity and behavior. The implementation validates that the solution meets real-time requirements.

Chapter 5 summarizes the key findings and contributions of this thesis. It also discusses limitations and potential areas for future work to build upon the research.

The thesis concludes by providing references.

CHAPTER 2

THEORETICAL FOUNDATION

Chapter 1 introduced the research problem and defined the study's goal and scope, which is to find a solution to the inter-camera association problem. This chapter goes deeper into commonly used solutions for inter-camera association.

2.1 Inter-camera association

2.1.1 Person Re-Identification based on visual features

Re-ID aims to identify a query person across multiple cameras. The query person can be represented by a track or a sequence of images, which is the output of the SCT process. Re-ID involves several steps, as shown in Figure 2.1. First, an empty gallery is created to store the identities during person tracking. New tracks obtained from SCT are fed into a pre-trained neural network to extract a visual embedding, which is then compared to the tracks in the gallery. If a similar track already exists in the gallery, the query track is merged with it as an identity. If not, a new identity in the gallery is created.

The neural network for extracting a person's visual embedding must be pre-trained on a Re-ID dataset. It takes an image or image sequence as input and outputs a feature embedding. The network is trained to encourage similar embeddings for the same person and dissimilar embeddings for different people. Some modify the architecture to obtain more discriminative features, such as OSNet [31]. Others focus on designing loss functions, such as cross-entropy, verification, and triplet loss (Figure 2.2) to guide feature learning.

Figure 2.1: An example of query-gallery management in Re-ID.

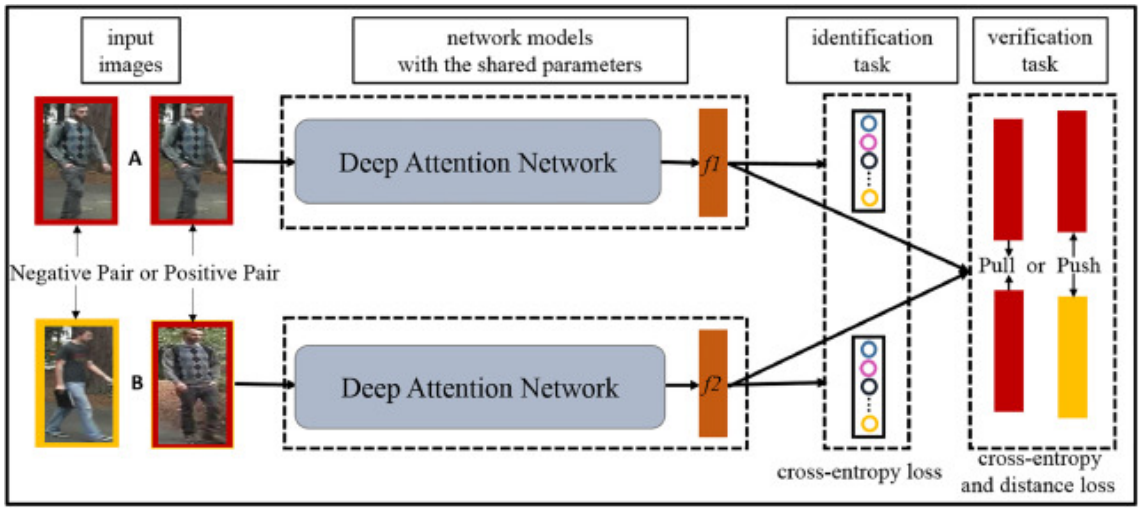


Figure 2.2: An example of training neural network for Re-ID task.

Re-ID using visual features can match people across different non-overlapping camera views. It doesn't require cameras to overlap or be synchronized in time. The intrinsic and extrinsic camera parameters can also be ignored.

However, this method requires a large training dataset to ensure recognize the same person under variations in viewpoints, illumination, poses, occlusions, etc. Similar clothing or unfavorable lighting conditions can also make different people look alike (see Figure 1.2). Thus, an effective neural network requires a diverse dataset with many identities, camera viewpoints, and environmental factors, in addition to a carefully designed training strategy.

Another drawback is that the query person may not appear in the gallery, requiring a person verification task rather than a retrieval task. Instead of finding the most similar person, this task must determine whether the query and gallery images show the same person. Typically, a carefully selected threshold is used, such that two people are considered the same if $\text{sim}(\text{query}, \text{gallery}) > \text{threshold}$.

Given the assumption that employees wear uniforms in this project's application scenario, comparing tracks based on people's appearance is challenging for Re-ID. Instead, comparing tracks based on their positions in space and time of appear-

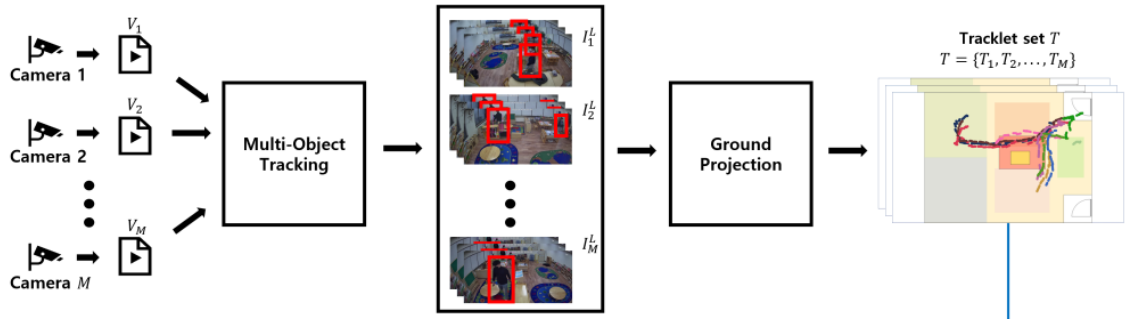


Figure 2.3: An example illustration of spatio-temporal system. The picture was borrowed from Jang et al. [15].

ance is more promising. This leads to the following studies on Spatio-Temporal Association.

2.1.2 Spatio-Temporal Association

Spatio-Temporal Association (STA) links tracks between cameras using timestamp and location in space (see Figure 2.3). Several approaches have been proposed to utilize STA in multi-cameras tracking. Anjum et al. [1] proposed an offline approach to associate trajectories from partially overlapping cameras. They created a virtual ground-plane and projected trajectories from image-planes to the ground-plane using homography. They computed feature vectors for each trajectory segment in overlapping areas, including shape, length, velocity, turns, position, and eigenvalues of the covariance matrix. Trajectory similarity was determined by cross-correlation between these features. The authors evaluated their approach on basketball and soccer match videos.

Zhang et al. [29] developed an online real-time algorithm for MCT with non-overlapping cameras. Their approach primarily used Re-ID for inter-camera matching, with STA to reduce candidate lists. They measured camera distance in advance. When a track disappears in one camera, tracks in other cameras are queried and matched based on appearance time traveling between cameras. The authors evaluated their approach on DukeMTMC Re-ID dataset with provided camera positions.

Chen et al. [5] proposed an online tracker that simultaneously performs intra-camera and inter-camera association. They treated Re-ID and STA independently and combined them to produce the final result. For Re-ID, the authors utilized a Deformable Parts Model to segment body parts, extract color and texture, and create a visual embedding. For STA, image-plane coordinates were projected onto a ground-plane coordinate, and person detection in the current frame was assigned to an active track based on proximity. Results from both branches were fused for

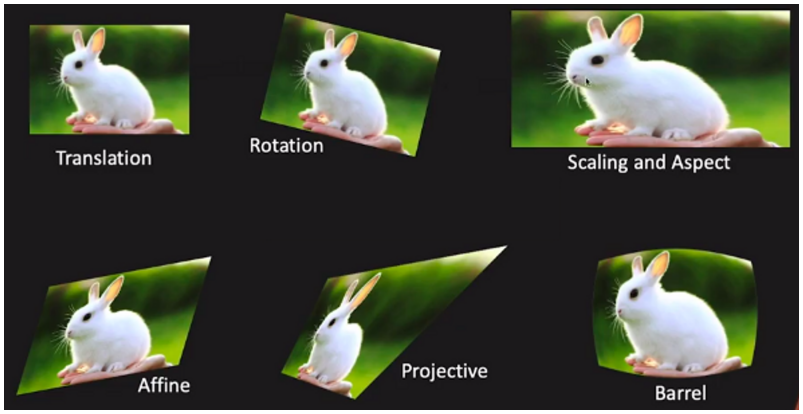


Figure 2.4: Some geometric image transformations

the final associations. The authors evaluated their approach with up to four people in a room, with overlapping camera views.

Jang et al. [15] also used image-to-ground projection with homography. To handle distortion in commercial cameras, they avoided expensive camera calibration by calculating moving direction instead of location. Moving direction was calculated using the Kanade-Lucas-Tomasi algorithm, and track similarity was estimated using Dynamic Time Warping. The authors evaluated their approach in classrooms with up to four overlapping cameras.

Previous studies have provided valuable insights for the proposed approach. However, those studies often consider STA as supplementary to Re-ID or cannot fully utilize this data due to camera limitations. This motivates a deeper investigation of STA as a standalone solution. To maximize STA usage, the proposed approach will focus on synchronized overlapping camera systems.

2.1.3 Image Warping and Homography

In Computer Vision, there is a branch of image processing related to manipulating the geometric shape of an image, called image warping. Image warping includes several linear geometric transformations such as scaling, rotation, skew, translation, affine, and projection (see figure 2.4). These linear transformations can be performed by carrying out a 3×3 matrix multiplication in homogeneous coordinates.

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}, \text{ or } p_2 \equiv \tilde{p}_2 = Hp_1$$

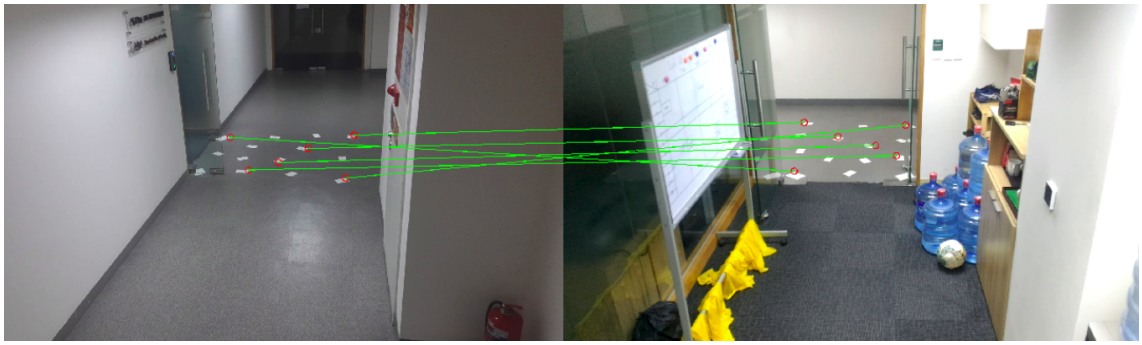


Figure 2.5: Manually selecting corresponding points in the FOV of a camera pair to build a homography matrix.

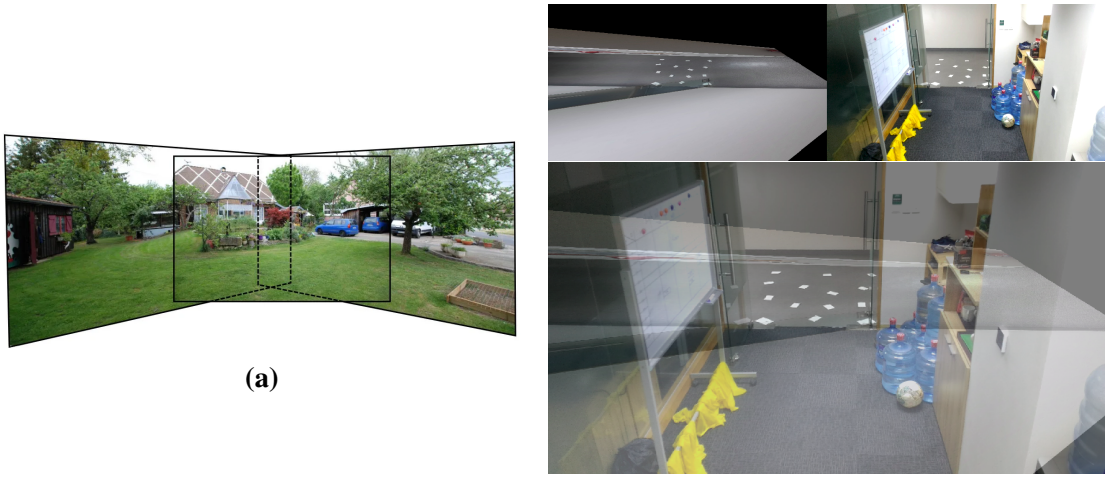
where p_1 is a pixel coordinate in the source image, p_2 is the corresponding pixel coordinate in the target image, $\tilde{p}_2$ is an equivalent to p_2 in homogeneous coordinates, H is called a projective matrix. Any transformation of this form is called *homography*.

To compute the projective matrix H, we need to know several corresponding pairs (p_1, p_2) beforehand (see Figure 2.5). These pairs are used to construct a system of linear equations with unknowns h_{ij} and the coefficients are determined by the known values x_1, y_1, x_2, y_2. While the minimum number of matching points required is 4, having more points is better, as it makes the homography estimate more robust.

The projective matrix H is used to map one plane to another plane through a point. Homography has many applications in photography, with the most common being image stitching in digital cameras (as shown in figure 2.6a). In MCT, homography has been used in many studies to match tracks between multiple cameras by mapping the floor of each camera view to a common plane (figure 2.6b), and then comparing the positions between tracks.

2.2 Single camera tracking

Single camera tracking is a task in computer vision that involves detecting and tracking objects in a video sequence captured by a single camera. Recently, several tracking algorithms have been developed for this task. SORT [3] uses the Kalman filter to anticipate possible boxes in the current frame based on detections in the previous frame. It then calculates the IoU between actually detected boxes and the anticipated ones to map the boxes between the two frames. This approach provides a simple yet effective way to track objects in a video sequence. One drawback of SORT [3] is that it ignores low-confidence detections, such as occluded people,



(a) **(b)**

Figure 2.6: Application of homography in Computer Vision. **a)** Image stitching in digital photography **b)** Ground plane transformation between multiple cameras.

leading to fragmented tracks. SORT [3] also has a fairly simple association criterion that only employs box positions and often suffers from *ID switch*.

ByteTrack [30] attempts to address the issue of low-confidence detections by incorporating them into the tracking process. It first matches the high-score detection boxes to tracks based on motion or appearance similarity as in SORT [3]. It then performs a second matching between the unmatched tracks and the low scores detection boxes using only motion similarity. ByteTrack [30] is more robust with detection threshold over SORT [3].

DeepSORT [28] was proposed to diminish the ID switch problem in SORT [3] by having a pretrained network to extract and compare the visual embeddings of bounding boxes. It mainly relies on appearance similarity to match boxes, but uses motion information to exclude unlikely matches. This approach is robust even in the presence of occlusions or when people reappear in the video after a disappearance. However, DeepSORT [28] is computationally expensive which limits its practical applications.

2.3 Software Frameworks for Development of DNN Models and Application System

The focus of this research is on finding a solution for inter-camera association using STA. Therefore, previous studies will be employed for single camera tracking task.

For person detection, YOLOv7 by Wang et al. [27] has shown impressive per-

formance on the COCO 2017 dataset. The author also provides the YOLOv7-Pose version for Pose estimation task based on the work of Maji et al. [17]. This version not only provides detection as a bounding box but also returns the pose of the person, which can provide a more accurate location of the person in space. Therefore, the pre-trained YOLOv7 and YOLOv7-Pose models will be used for person detection task.

For intra-camera association, SORT [3], DeepSORT [28], StrongSORT [7], and ByteTrack [30] are among commonly used algorithms. In this project's application scenario where employees have similar appearances, visual information cannot be used, so DeepSORT [28] and StrongSORT [7] will be eliminated. ByteTrack [30] is an improvement from SORT [3] that aims to reduce track fragmentation and is well-known for its real-time capability. Therefore, ByteTrack [30] will be used as the tracker for single camera tracking.

To carry out these tasks, Scikit-learn [21], PyTorch [20], OpenCV [4], SciPy [26] will be used to run machine learning algorithms, deep learning models, images processing and optimization respectively. Additionally, Flask [19], a web framework written in Python, will be used to build the application system for the sake of demonstration.

CHAPTER 3

PROPOSED METHOD AND EVALUATION

This chapter presents in detail the theory behind the proposed method, the experimental methodology, and a discussion of the experimental results.

3.1 Overview

As mentioned in section 1.3, three objectives had been set for this graduation thesis: research, technology, and application. Overall, the research problem aims to propose a solution that utilizes spatio-temporal information to map tracks between cameras as a replacement for Re-ID. This is applicable in cases where cameras have overlapping views and synchronized time.

However, in order to truly evaluate the practicality of the proposed solution, it is important to be able to explain the experimental results by answering the following questions:

1. What are the underlying issues with the proposed method at the frame-level?
2. In which scenarios does the method work/fail?
3. When should the method (not) be used?
4. If the method is used, how should it be used?

Additionally, as previously mentioned, in order to perform Multi-Camera Tracking (MCT), the results of Single Camera Tracking (SCT) must first be obtained to generate tracks for each camera. However, a common issue with SCT is *ID switch*, which occurs when the identity of a track is mistakenly switched with another

track due to occlusion, similar appearance, or errors in the tracking algorithm. Researchers are finding ways to address this issue. And, as an extension, another interesting question to be explored in this thesis is: Can the issue of ID switch be mitigated by using the results of MCT?

In general, the experimental procedure in this thesis involves the following loop:

1. Collecting and labeling data.
2. Proposing and presenting the theory behind the solution.
3. Selecting evaluation metric.
4. Analyzing experimental results.
5. Identifying weaknesses of the proposed solution and proposing a better solution
6. Returning to step 4.

3.2 Dataset

According to the the scope of research defined in the Section 1.4, a video dataset used for experiments will follow the below criterias:

- The system consists of 3 cameras covering all areas of the store.
- The cameras are synchronized in time.
- A camera has an overlapping FOV with at least one another camera.
- Maximum of 4 people are in the overlapping view at any given time.
- Employees wear the same uniform.
- Employee behaviors: Staying in designated work positions or moving between pairs of cameras.

Moreover, previous studies on STA have often been conducted on large overlapping areas, such as an entire classroom, and with relatively long periods of time for people to move within the overlapping area. These conditions are not very practical. In real-world scenarios, the overlapping area between cameras is often small, such as in stores or companies, and people move relatively quickly within the overlapping area. With such conditions, there is no publicly available dataset that meets the requirements. Therefore, the video dataset used in this thesis will be recorded and annotated manually. The camera setup is illustrated in Figure 3.1. The filming location is inside a building hallway (see Figure 3.2).

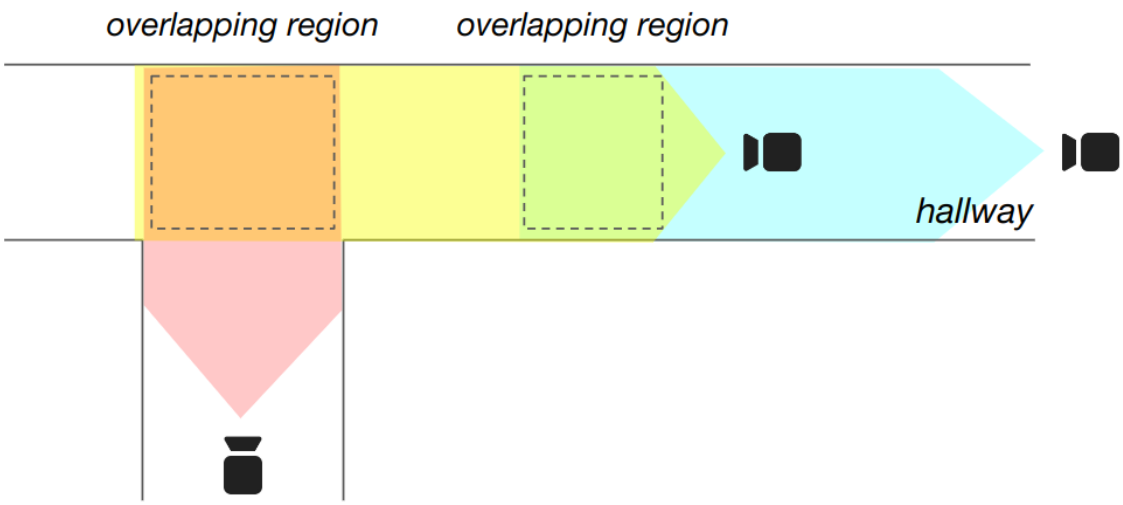


Figure 3.1: The camera setup layout for data collection.

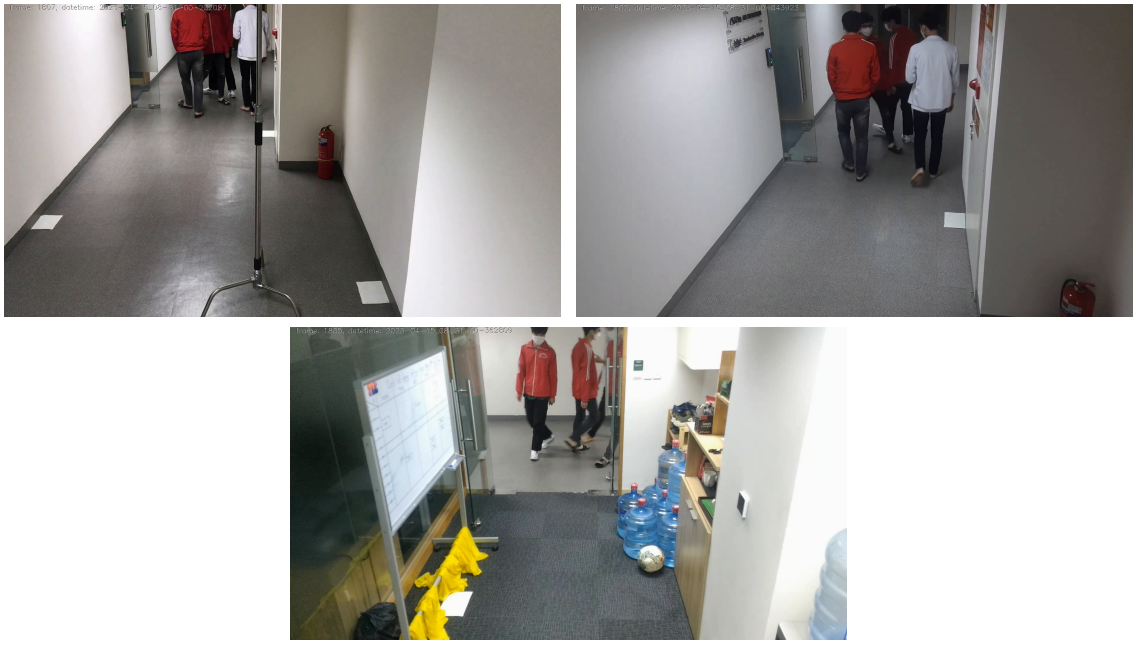


Figure 3.2: Field of view of 3 cameras after setting up.

The area of the overlapping region in the camera setup is approximately $4m^2$. People move within the FOV of the cameras at a normal pace. There are a total of 36 videos divided equally into 3 sets:

- Easy set: 2 people move simultaneously through the overlapping region.
- Medium set: 3 people move simultaneously through the overlapping region.
- Hard set: 4 people move simultaneously through the overlapping region.

Single camera tracking task was performed on a recorded video set using YOLOv7 and ByteTrack. YOLOv7 was pre-trained on COCO2017 with 70.4 million parameters. The HOTA metric [16] was used evaluate the performance of the algorithms,

Video set	ID	Camera 1			Camera 2			Camera 3		
		GTs	IDs	SWs	GTs	IDs	SWs	GTs	IDs	SWs
Easy	1	2	4	2	2	6	4	2	2	0
	2	2	4	2	2	5	3	2	5	3
	3	2	4	2	2	6	4	2	3	1
	4	2	5	5	2	8	6	2	3	1
Medium	5	3	6	3	3	10	7	3	5	7
	6	3	9	6	3	10	7	3	7	5
	9	4	8	7	4	9	7	4	5	1
	10	3	8	5	3	9	7	3	4	1
Hard	7	5	13	10	5	15	13	6	9	5
	8	5	14	9	5	15	11	5	18	16
	11	4	11	7	4	12	11	4	6	3
	12	4	13	9	4	14	14	4	21	23

Table 3.1: Statistics on the number of ground-truth identities (GTs), the number of tracks (IDs) obtained by YOLOv7 + ByteTrack, and the number of ID switches (SWs) determined by CLEAR MOT metric for each video.

which measures both Detection Accuracy and Association Accuracy. The HOTA score obtained was 63.32, with a Detection Accuracy of 66.06 and an Association Accuracy of 61.09.

Table 3.1 also provides other statistics on the performance of the algorithms by showing the number of ground-truth identities, predicted tracks, and ID switches for each video. The number of ID switches was determined using the CLEAR MOT metric [2]. The table proves that as the video set's complexity increased, the occurrence of ID switches also increased.

3.3 The Proposed Method

3.3.1 Frame-level versus Track-level matching

Previous studies on MCT and STA focused on track-level matching. However, track-level matching poses some issues. Firstly, matching tracks between cameras is very complicated due to the ID switch problem. For example, in Figure 3.3 tracks Green, Lime, and Blue on camera 2 need matching to tracks Red and Orange on camera 1. However, track Green has ID switch issue, as it was assigned to both ground-truth people. If it is matched with track Red, then track Lime cannot be matched with track Orange because all tracks Green, Lime, and Orange appear

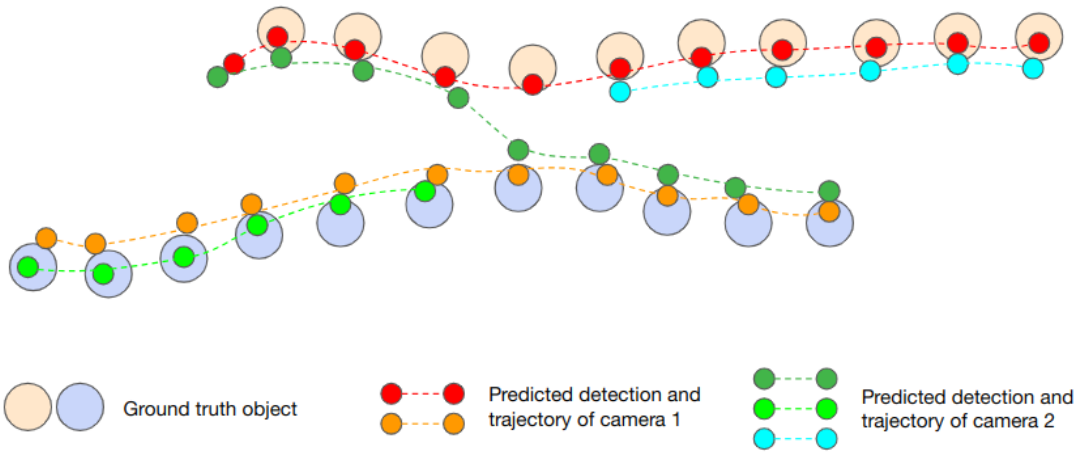


Figure 3.3: A simple example of confusing track-level matching due to ID switch. From the single camera tracking results, camera 1 has tracks for Red and Orange. Camera 2 has tracks for Green, Lime, and Blue.

together for the first half of the time. The same thing happens if we try to match track Green with track Orange. Therefore, it is complicated to perform inter-camera association at the track level.

Secondly, evaluating method reliability requires error analysis at the frame level, which is deeper than track level matching. Also, track-level evaluation may provide unreliable results. For example, in Figure 3.3, suppose track Green is matched with track Orange at the track level. However, but frame-level analysis shows that Orange should have been assigned to Lime for the first half of the time. Consequently, the evaluator will accidentally assess Orange-Lime pairs as false positives (FP) and Green-Orange pairs as false negatives (FN) during this first half.

Matching at the frame level avoids the above issues. It can also produce matching results at the track level if necessary and address SCT issues like detecting and correcting ID switches. Therefore, the proposed method involves frame-level matching.

3.3.2 The baseline version of the proposed method

3.3.2.1 Theory and Evaluation

Figure 3.4 provides an overview of the proposed method, which consists of three main steps: (1) Foot point interpolation and projection, (2) Frame-level timestamp matching, and (3) Frame-level object matching.

In the first step, for each detection box within a frame that represents a person, the midpoint of bottom edge is used as foot point and projected onto a common

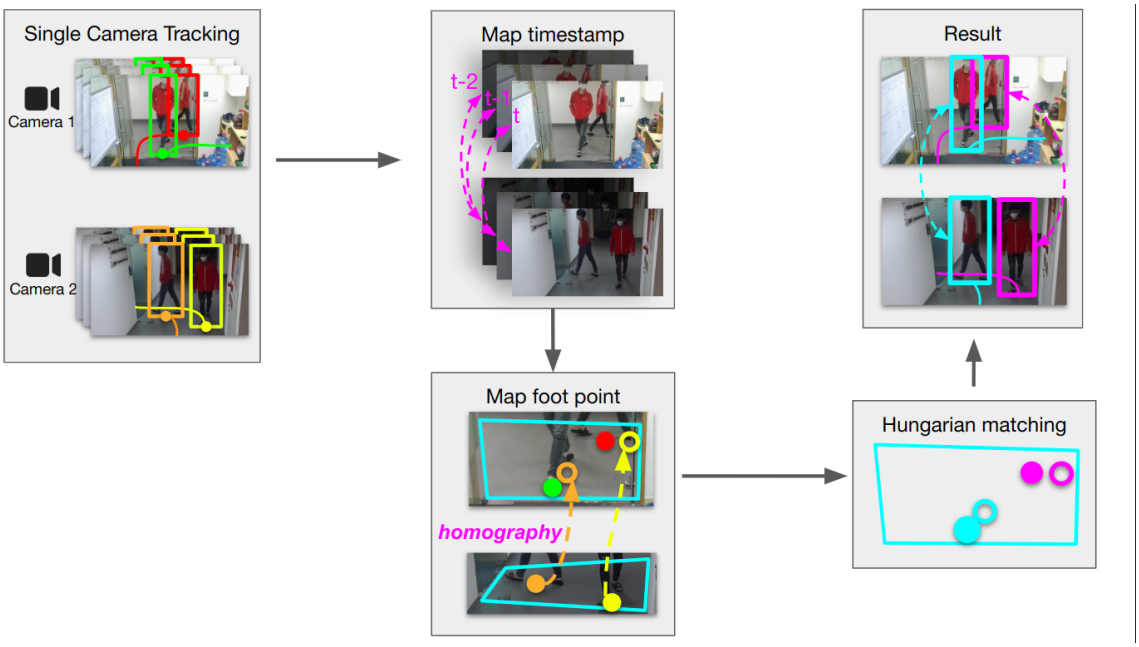


Figure 3.4: Overview of the proposed method for inter-camera association using Spatio-Temporal Association.

image plane using a homography matrix (see Figure 3.5a). The homography matrix is pre-computed by manually selecting corresponding points in the FOV of a camera pair (see Figure 2.5).

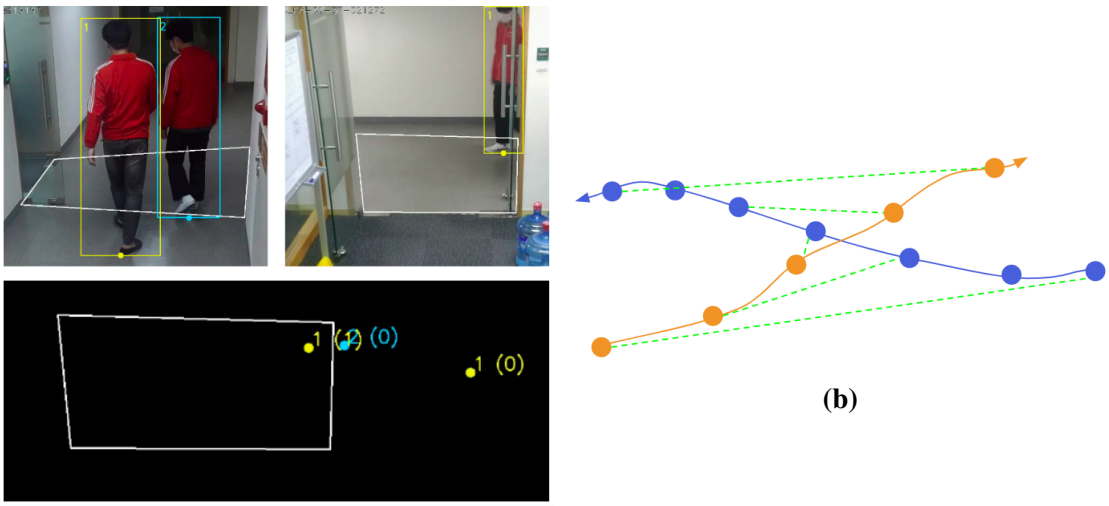
In the second step, the tracks of the two cameras are aligned and synchronized in time. Figure 3.5b illustrates an example of aligning two tracks based on their timestamps, where for each pair of aligned detections, their timestamps should be as close as possible. This alignment is one-to-one, however, due to possible differences in the FPS of the two cameras, some timestamps may not have a corresponding match and will be skipped.

In the third step, for each pair of aligned detections, their Euclidean distance is calculated based on the foot point interpolated in step 1. Finally, Hungarian algorithm is applied to find the best matching pairs.

Precision, Recall, and F1 score are selected to evaluate the algorithm:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP is the number of correctly mapped detection pairs, FP is the number of falsely mapped detection pairs, and FN is the number of missed detection pairs, all accumulated over frames in the video. Table 3.2 summarizes the evaluation results of the proposed method on a dataset of 12 videos, where the single camera tracking



(a) (b)

Figure 3.5: a) An example of foot point interpolation and projection. **b)** Frame-level timestamp matching between 2 tracks.

Video set	#TP	#FP	#FN	F1
Easy	511	7	7	0.986
Medium	662	46	22	0.951
Hard	966	144	41	0.913
Total	2139	197	70	0.941

Table 3.2: Performance of the proposed baseline method on the recorded videos.

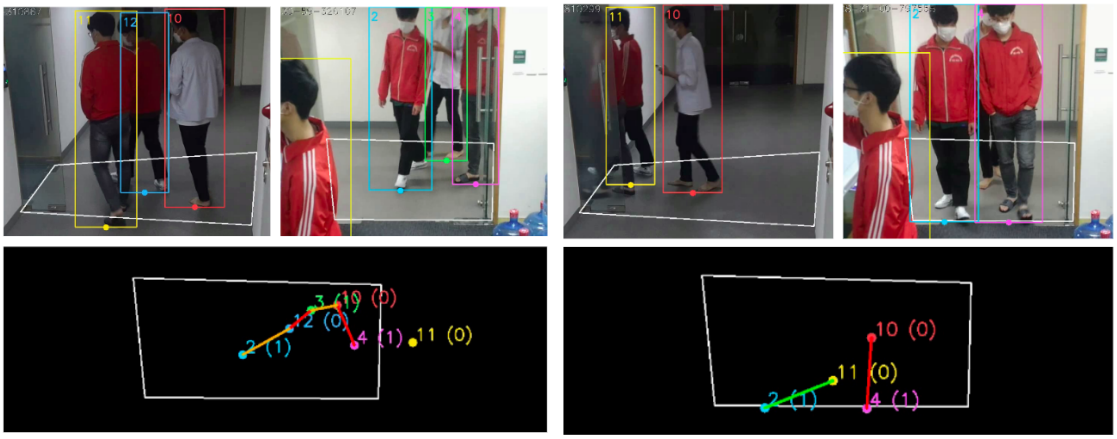
task is performed using the YOLOv7 and ByteTrack models, and the foot point is interpolated as the midpoint of the bottom edge of the bounding box.

The proposed method yields promising results. However, its performance decreases as the difficulty level of the videos increases. Notably, the number of FP is significantly higher than the number of FN, especially in the hard and medium video sets.

3.3.2.2 Error analysis

FP and FN were caused by two main factors. The first factor is loose bounding boxes or inaccurate foot points interpolation (see Figure 3.6a). Accurate foot point is critical to match individuals. However, when a person walks with their legs apart, the interpolated foot point will be the lowest point of their visible body in the camera's FOV. For example, in Figure 3.6a camera 1 (left) shows ID 11 with an inaccurate interpolated foot point, despite the tight bounding box. This inaccuracy places him outside the overlapping region.

The second cause of FP and FN is missing detections due to occlusion. If two



(a) **(b)**

Figure 3.6: Cause of FP and FN **a)** due to incorrect foot point interpolation. **b)** due to missing detection. The green line indicates a TP match, the red line indicates a FP match, the yellow line indicates a FN match.

different people are missed on different cameras, their tracks can be incorrectly matched together. For example, in Figure 3.6b, the person with ID 4 on camera 2 is occluded on camera 1 while the person with ID 10 on camera 1 is occluded on camera 2, resulting in an incorrect match. This situation is common due to the small overlapping area (about 4 m^2) and a high number of people simultaneously passing through it. These occlusions generate many FP and few FN, leading to a higher number of FP than FN.

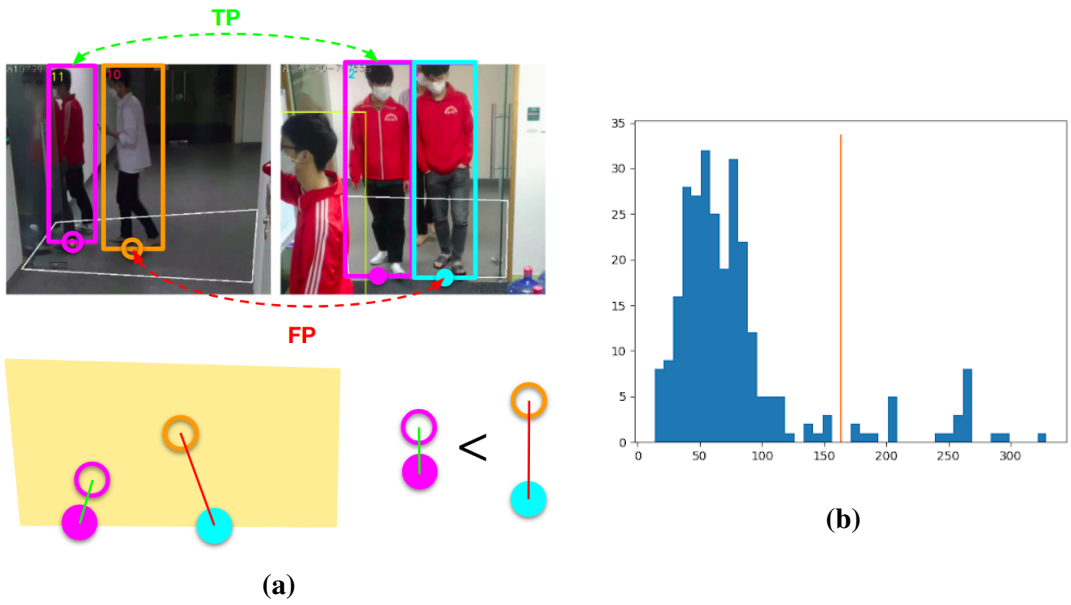
In the next section, a better solution will be presented to mitigate the problem of FP caused by missing detections.

3.3.3 Extension 1 to the baseline method: Address missing detections by FP filtering

3.3.3.1 Theory and Evaluation

As being pointed out in the previous step of the error analysis, one of the main causes of many FP is when bounding boxes are missed due to two different people being missed on two different cameras. To reduce this issue, a reasonable assumption is that: *False positives due to missing detections have larger spatial distance than true positives*. For example, in Figure 3.7a, there is one TP and one FP, where the FP is assumed to have a larger spatial distance than the TP. If this assumption is correct, then these FP can be treated as outliers and outlier detection methods can be used to remove them.

One commonly used outlier detection method is the Inter-quantile range (IQR).



(a) (b)

Figure 3.7: a) Demonstration of the assumption made for the FP filtering. **b)** Distance distribution of pairs matched by the baseline approach for one of the recorded videos. The *x*-axis represents the spatial distance of matched pairs. The *y*-axis represents the number of matched pairs. The vertical seam represents the upper bound by IQR(25, 75).

Video set	Baseline	IQR (15,85)	**IQR (20, 80)**	IQR (25,75)
Easy	0.986 (511,7,7)	0.983 (507,7,11)	0.983 (507,7,11)	0.983 (507,7,11)
Medium	0.951 (662,46,22)	0.954 (659,39,25)	0.958 (658,32,26)	0.949 (638,22,46)
Hard	0.913 (966,144,41)	0.926 (962,109,45)	0.927 (959,103,48)	0.928 (958,99,49)
Total	0.941 (2139,197,70)	0.941 (2139,197,70)	0.949 (2124,142,85)	0.947 (2103,128,106)

Table 3.3: The comparison between the baseline method and the version that uses IQR on different percentiles. Each cell value is in the format of F1(#TP, #FP, #FN).

Figure 3.7b shows the histogram of the distance distribution of pairs matched by the baseline method. It can be seen that this is a bell-shaped distribution with no significant skewness, making it appropriate to apply the IQR method.

Basically, everything remains the same as the baseline method. Additionally, the spatial distances of the matched pairs are collected during the iteration through frames. After finishing the iteration, the IQR of the distance set is used as a threshold to filter out the suspicious candidates. To determine the appropriate percentiles, experiments need to be done several times. Table 3.3 compares the baseline method with the version that uses IQR on different percentiles.

As shown in Table 3.3, when FP filtering is not applied, the number of FP is

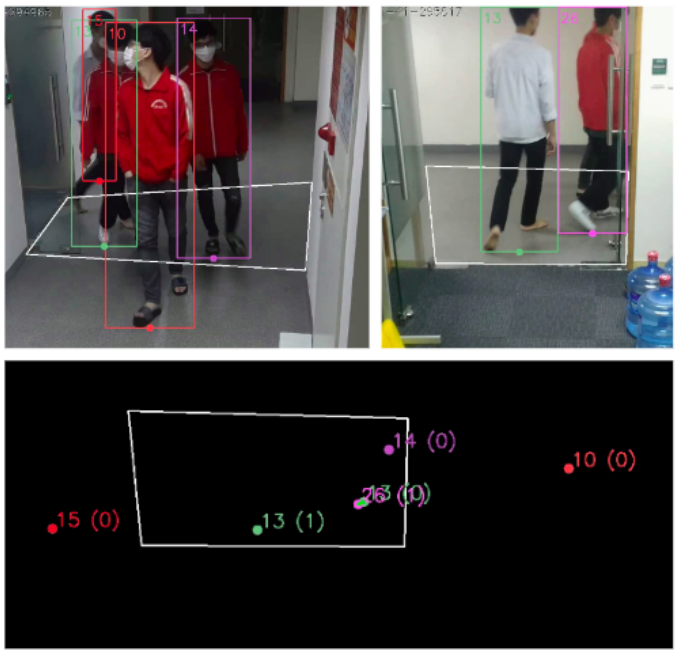


Figure 3.8: A typical example where FP filtering with IQR correctly eliminate a FP match.

significantly higher than the number of FN. After applying IQR:

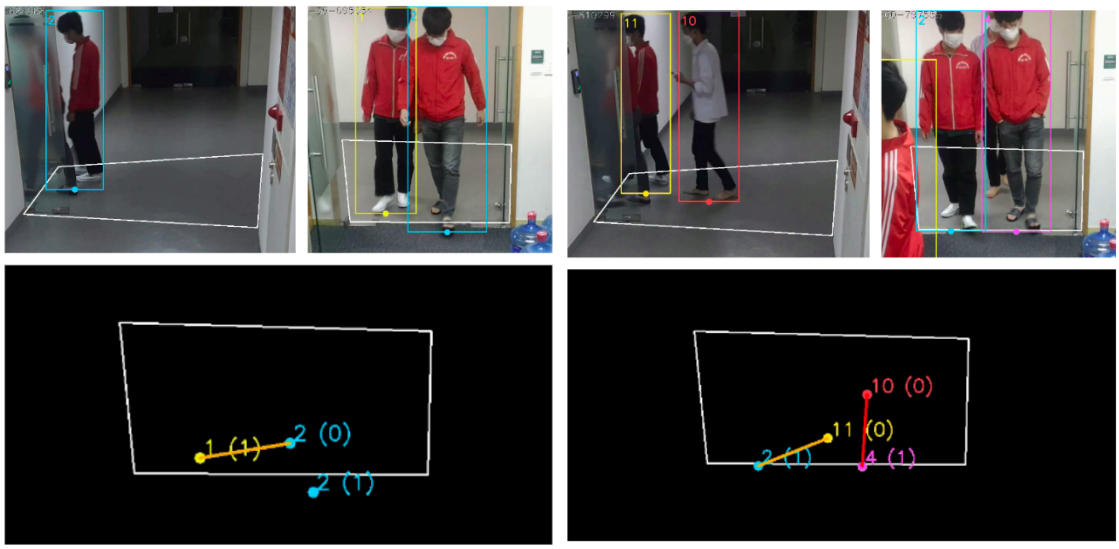
1. For the easy set: using IQR does not have a significant impact on the performance. The number of FP remains the same while the number of FN slightly increases.
2. For the medium set: using IQR improves the performance. The number of FP decreases significantly while the number of FN slightly increases.
3. For the hard set: using IQR improves the performance. The number of FP decreases more than the easy and medium sets, while the number of FN slightly increases.

To evaluate the impact of FP filtering more deeply, an error analysis was conducted, which is presented in the next section.

3.3.3.2 Error analysis

Figure 3.8 shows an example of successful FP filtering using IQR, which eliminates a FP match. The person with ID 13 on camera 2 (right) is missed on camera 1 and the person with ID 14 on camera 1 (left) is missed on camera 2. These two people are not falsely matched, unlike in the baseline method.

However, IQR can also create FN by removing correct matches, as shown in Figure 3.9a. The bounding box's inaccurate fit causes an incorrect distance measurement between the two footpoints, leading to the removal of a correct match



(a) (b)

Figure 3.9: Two typical examples where FP filtering with IQR produces worse effect. The green line indicates a true positive match, the red line indicates a FP match, the yellow line indicates a FN match.

between ID 2 on camera 1 and ID 1 on camera 2.

Figure 3.9b demonstrates another case where IQR fails to remove a false positive match between ID 10 on camera 1 and ID 4 on camera 2. The distance of this FP is within the interquartile range and the distance distribution is likely to be negatively affected by inaccurate foot points.

Overall, FP filtering using IQR works as expected, as follows:

- It significantly reduces the number of FP in complex cases, with a slight increase in FN that is not significant compared to the decrease in FP.
- In simple cases, it has a negative but negligible impact, with a slight decrease or no change in the number of FP and a slight increase in FN that is less than the decrease in FP.

3.3.4 Extension 2 to the baseline method: Address missing detections by Window-based mapping

3.3.4.1 Theory and Evaluation

During the error analysis of the baseline method, it's noticeable that sometimes part of a track can be matched almost perfectly with a corresponding part of a track on the other camera, except for one or a few frames where the match is incorrect. However, if we observe the matching in the neighboring frames, we can fix

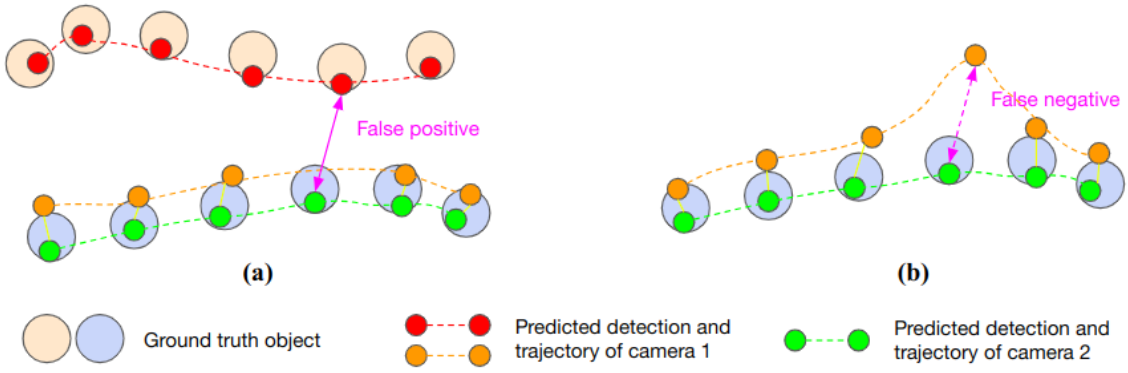


Figure 3.10: An example where window-based mapping may help reducing **a)** false positive and **b)** false negative. From the single camera tracking results, camera 1 has tracks for Red and Orange. Camera 2 has tracks for Lime.

those incorrect matches. This observation suggests using the average distance over neighboring frames of a pair of IDs as an input cost for the Hungarian algorithm.

For example, in Figure 3.10a, camera 1 has two tracks, Red and Orange, and camera 2 has one track, Lime. On most frames, Green is correctly matched with Orange. However, there is one frame where Orange is missed, causing Green to be incorrectly matched with Red and generating a FP. In the baseline method, only the boxes appearing at that time frame are taken into account. But the average distance between Green and Orange over a window of the neighboring frames is still smaller than the average distance between Green and Red. As a result, a FP can be eliminated.

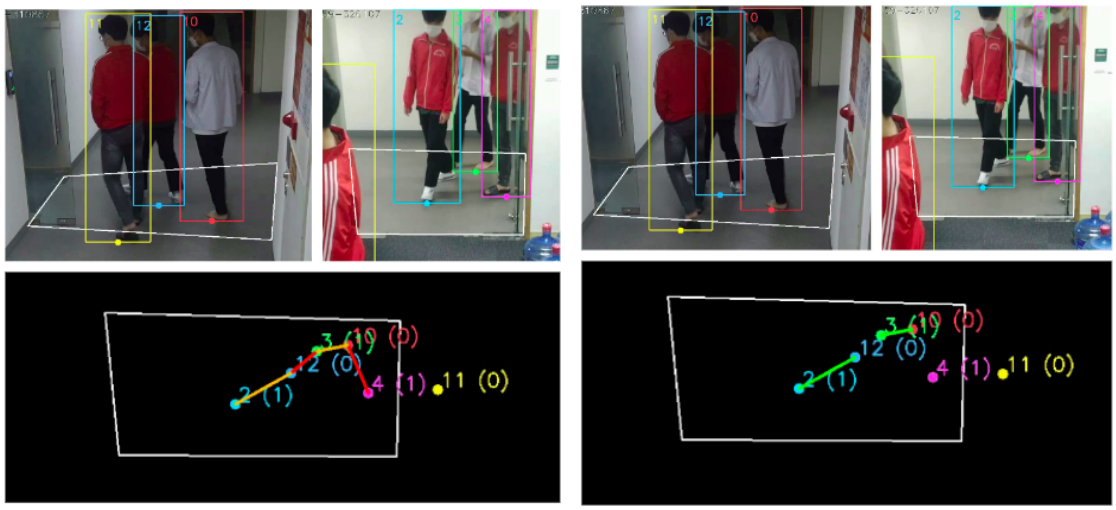
Another example is shown in Figure 3.10b, where each camera has one track assigned to the same person. However, at some frame, the footpoint of Orange is far away from Green, which may be due to partial occlusion. If this distance is large enough, it will generate a FN. But if the distances over neighboring frames are averaged, the distance will decrease and a FN can be avoided.

To improve the baseline method, a window-based strategy was applied independently from FP filtering. The basic approach remained the same as the baseline method, but instead of measuring the distance between two footpoints at one frame, the distance over the surrounding frames was averaged. To determine the appropriate window size, several experiments would need conducting.

Table 3.4 compares the baseline method with the window-based version using different window sizes. The results show that all three video sets - easy, medium, and hard - experienced a slight decrease in both FP and FN. The hard videos showed a greater improvement.

Video set	Baseline	size = 7	Size = 11	**Size = 15**
Easy	0.986 (511,7,7)	0.990 (513,5,5)	0.990 (513,5,5)	0.994 (515,3,3)
Medium	0.951 (662,46,22)	0.958 (667,41,17)	0.958 (667,41,17)	0.955 (665,43,19)
Hard	0.913 (966,144,41)	0.915 (968,142,39)	0.921 (975,135,32)	0.921 (975,135,32)
Total	0.941 (2139,197,70)	0.945 (2148,188,61)	0.948 (2155,181,54)	0.948 (2155,181,54)

Table 3.4: The comparison between the baseline method and the version that uses window-based mapping on different window size. Each cell value is in the format of F1(#TP,#FP,#FN).



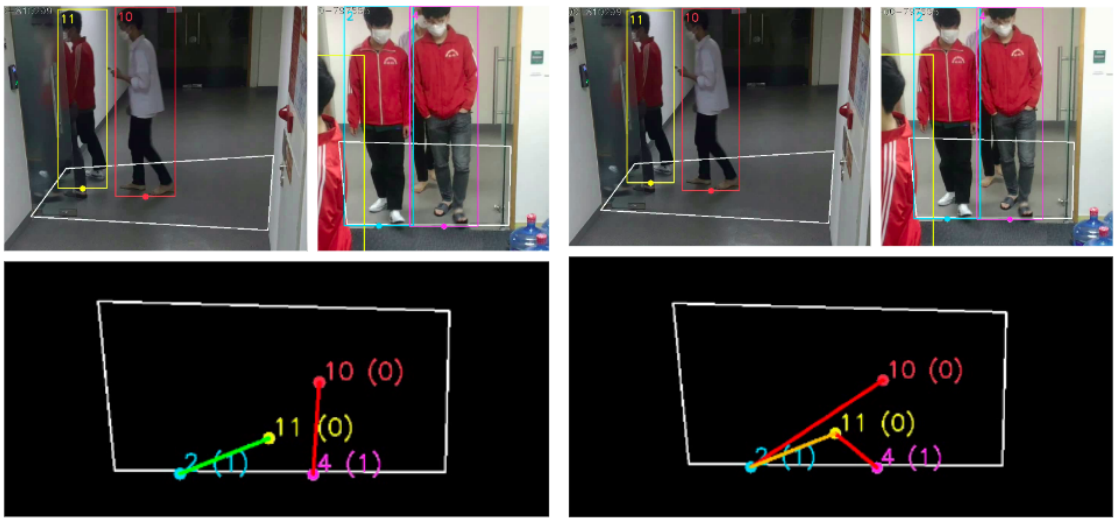
(a) **(b)**

Figure 3.11: An example where the window-based mapping performs well. The green line indicates a true positive match, the red line indicates a FP match, the yellow line indicates a FN match.

3.3.4.2 Error analysis

Figure 3.11 shows a successful frame where window-based mapping is effective. Figure 3.11a shows the baseline method's matching result, while Figure 3.11b shows the result after applying window-based mapping. Prior to this frame, ID 11 on camera 1 and ID 4 on camera 2 were not in the overlapping area on both cameras. During that preceding interval, only four tracks covered two people, accumulating enough correct matches to rectify the incorrect match in the current frame.

On the other hand, Figure 3.12 shows a frame where the window-based mapping results in a worse outcome. ID 11 on camera 1 and ID 4 on camera 2 were incor-



(a) **(b)**

Figure 3.12: An example where the window-based mapping fails. The green line indicates a true positive match, the red line indicates a FP match, the yellow line indicates a FN match.

rectly matched on several previous frames due to the missing person corresponding to ID 10 on camera 1, leading to an incorrect match in the current frame.

Overall, the window-based mapping works as expected:

- It positively impacts most cases, particularly complex ones, although the improvement is not significant. It decreases the number of FP and FN.
- Compared to FP filtering:
  - FP filtering significantly reduces FP but slightly increases FN.
  - The window-based mapping reduces FP, but is weaker than FP filtering in this regard. It also reduces FN instead of increasing it, unlike FP filtering does.

3.3.5 Extension 3 to the baseline method: Combining FP Filtering and Window-based mapping

By combining the FP filtering and window-based mapping solutions, greater benefits can be achieved. Table 3.5 summarizes the experiments of combining these two solutions with different parameter sets. However, due to limitations in page layout, only the parameter sets with the best results are listed in this table.

From the results in Table 3.5, it can be seen that combining the FP filter and window-based mapping produces more impressive results than using only one of the two extensions or not using them at all. Specifically:

Video set	Baseline	**IQR(20,80) size = 11**	IQR(20,80) size = 15	IQR(25,75) size = 15
Easy	0.986 (511,7,7)	0.986 (509,5,9)	0.991 (512,3,6)	0.987 (508,3,10)
Medium	0.951 (662,46,22)	0.969 (662,20,22)	0.965 (658,22,26)	0.965 (655,19,29)
Hard	0.913 (966,144,41)	0.938 (963,83,44)	0.939 (962,79,45)	0.941 (950,62,57)
Total	0.941 (2139,197,70)	0.959 (2134,108,75)	0.959 (2132,104,77)	0.959 (2113,84,96)

Table 3.5: The comparison between the baseline method and the version that uses IQR and window-based mapping. Each cell value is in the format of F1(#TP,#FP,#FN).

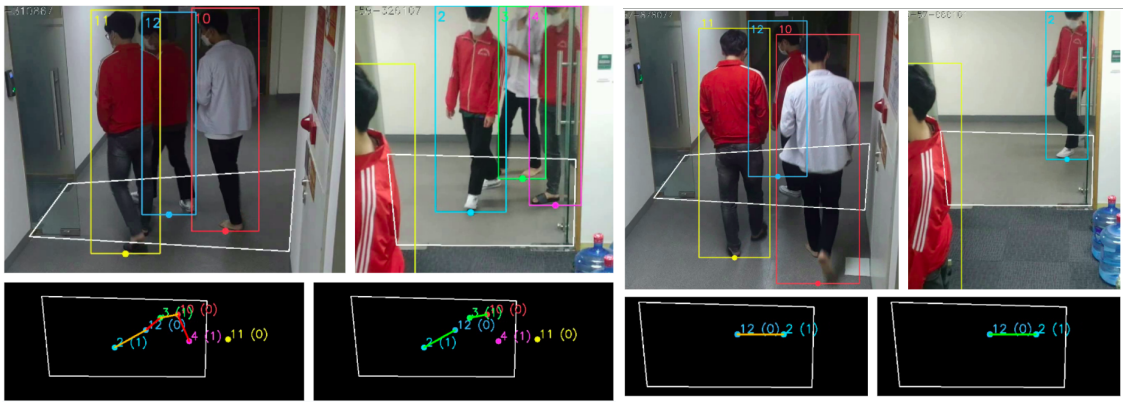
- For the easy set: there is no significant change overall.
- For the medium set: the number of FP decreases significantly, while the number of FN remains the same or slightly increases, but not significantly.
- For the hard set: the number of FP decreases the most significantly, while the number of FN increases slightly. However, the amount of increase in FN is much smaller than the amount of decrease in FP.

3.3.5.1 Error Analysis

The error analysis in this section will be slightly different from the previous sections. The combination of FP filtering and window-based mapping will be compared with using only one of the two extensions rather than the baseline method. My purpose is to see if FP filtering and window-based mapping can complement each other.

Figure 3.13 illustrates two cases where the combination of FP filtering and window-based mapping produces better results than using only FP filtering. In the first case, shown in Figure 3.13a, FP filtering fails to remove FP with temporarily smaller distances than that of FN. Window-based mapping can remove these temporary FP pairs, as it provides the necessary support. The second case, shown in Figure 3.13b, occurs when the distance between the detections of the same person on two cameras is temporarily larger than expected due to wrong foot points. In this case, FP filtering removes this match, whereas window-based mapping can keep this match by using matching information from the past and future.

On the other hand, Figure 3.14 provides an example where the combination of FP filtering and window-based mapping works better than using only window-



(a) **(b)**

Figure 3.13: Two examples where the combination of FP filtering and window-based mapping produces better results than using only FP filtering. **a)** Spatial distance of FP is smaller than that of FN. **b)** Spatial distance of the FN is larger than the filtering threshold. The green line indicates a true positive match, the red line indicates a FP match, the yellow line indicates a FN match. The bottom left of each subfigure is the matching result of FP filtering alone. The bottom right of each subfigure is the matching result of the combination.

based mapping. It shows a moment from a continuous sequence of frames where only ID 10 of camera 1 and ID 26 of camera 2 simultaneously appear in the overlap area. If only window-based mapping is used, these two IDs will be matched incorrectly. However, thanks to FP filtering, many of these FP can be eliminated.

Combining FP filtering with window-based mapping produces better results than using either method alone. The following effects are observed when combined:

- It significantly reduces FP in difficult cases more than using only FP filtering or only window-based mapping, with a decrease or slight increase in FN.
- It has a negative but negligible impact on easy cases, with a slight decrease or no change in FP and a slight increase in FN.
- Loosening the parameter values for each extension is necessary when using both, as they have similar effects. For example, IQR(20, 80) and a window size of 11 can be selected instead of IQR(25, 75) or a window size of 15 when used separately.

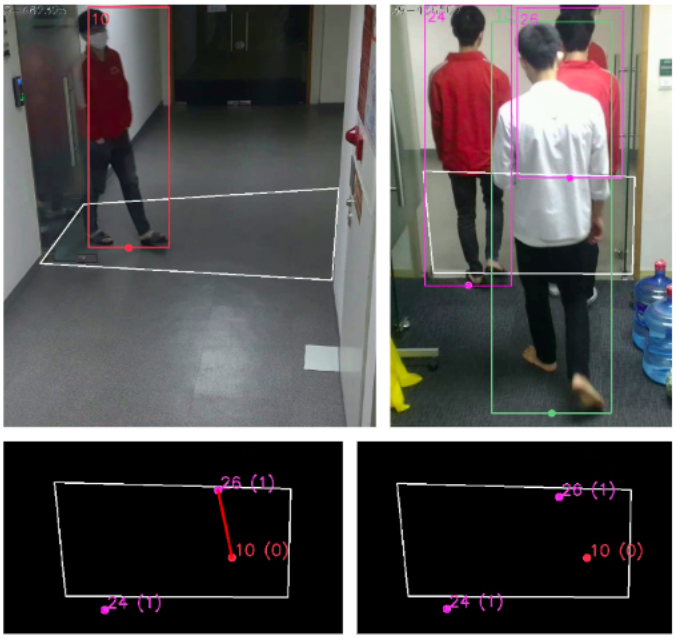


Figure 3.14: An example where the combination of FP filtering and window-based mapping produces better results than using only window-based mapping. The red line indicates a FP match. The bottom left of each subfigure is the matching result of FP filtering alone. The bottom right of each subfigure is the matching result of the combination.

3.3.6 Extension 4 to the baseline method: Address inaccurate foot point by Pose estimation

3.3.6.1 Theory and Evaluation

One of the main issues that cause matching errors is because the bounding box does not fit the person tight or the interpolated foot point is inaccurate.

As discussed in the section 3.3.2.2, one of the main issues causing incorrect or missing matches is that the box doesn't fit the person or the footpoints are interpolated inaccurately. Even if detection models provide rectangular bounding boxes that fit the body well, the footpoints may not be accurate. Since matching cameras using STA relies heavily on the footpoints, an alternative to the rectangular bounding box is needed. One solution is pose estimation.

Pose estimation is a computer vision task that aims to detect the positions of keypoints in the human body, including the feet. The authors of YOLOv7 [27] have also provided a pre-trained version of YOLOv7-Pose, which is very convenient to compare the effectiveness of the familiar rectangular bounding box with pose estimation with no neural network architecture bias.

Before conducting the evaluation experiments, our expectations of using pose are:

Video set	Baseline with box	**Baseline with pose**
Easy	0.986 (511,7,7)	0.985 (665,11,9)
Medium	0.951 (662,46,22)	0.950 (883,81,11)
Hard	0.913 (966,144,41)	0.928 (1200,145,42)
Total	0.941 (2139,197,70)	0.948 (2748,237,62)

Table 3.6: The comparison between the baseline method using bounding box and using pose estimation. Each cell value is in the format of F1(#TP,#FP,#FN).

1. First, pose estimation will provide more accurate foot points than interpolation from the bounding box.
2. Second, using pose estimation will have a greater and positive impact on hard cases than on easy ones. This is because hard videos involve complex situations such as crowded scenes and occlusion that require accurate foot points.

Table 3.6 summarizes the comparison results between using pose estimation and bounding box with the baseline method. Overall, using pose estimation leads to better matching results on the hard video set. However, when examining the evaluation results on each individual video (see Table 3.7), there is a concerning observation phenomenon:

1. 5 videos showed better results: 2, 5, 9, 7, 8. Among them, the significant changes occurred in videos 7 and 8 - two videos belonging to the hard set.
2. 5 videos showed worse results: 4, 6, 10, 11, 12. Among them, the significant changes occurred in videos 10, 11, and 12 - two videos belonging to the hard set and one video belonging to the medium set.

The two hard videos with worse results showed a sharp increase in the number of FP when using pose estimation, which is contrary to the initial expectation. Therefore, an error analysis needs to be done to understand the cause.

3.3.6.2 Error Analysis

Firstly, cases where using pose estimation produced better results than using bounding box are investigated. Figure 3.15 demonstrates an example where using pose estimation resulted in more accurate foot points than using the bounding box, which helped to avoid errors in matching. To further support this finding, the spatial distance distribution between matched foot points in a video when using bounding box and pose estimation were plotted in Figure 3.16. The distribution generated by pose estimation is skewed to the left compared to the distribution generated by the

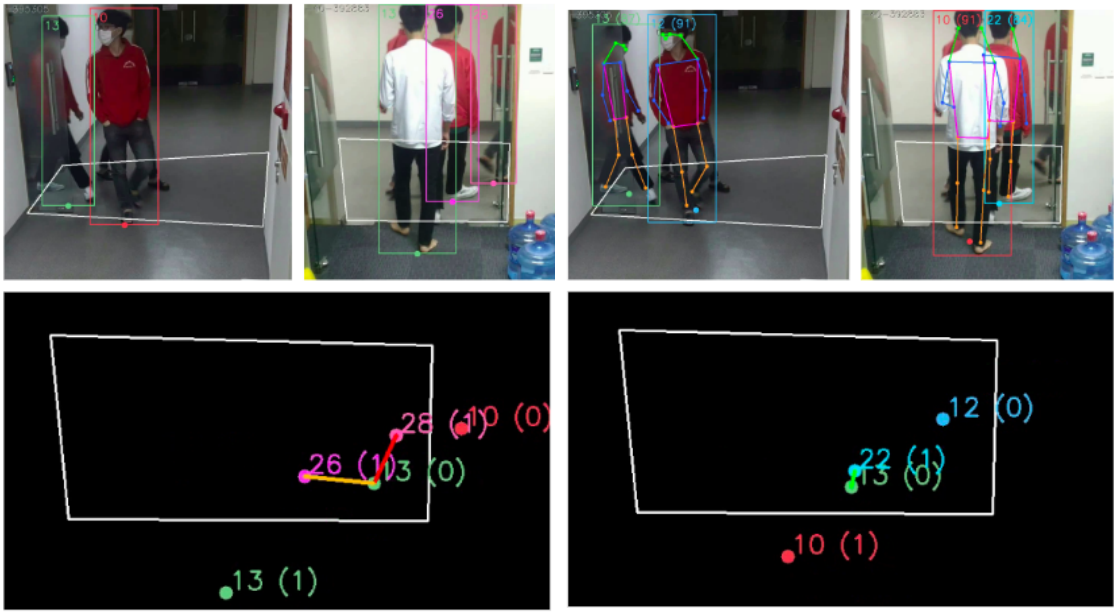
Video set	Video ID	Baseline with box	Baseline with pose
Easy	1	1.0 (117,0,0)	1.0 (164,0,0)
	2	0.959 (139,6,6)	0.971 (186,6,5)
	3	1.0 (97,0,0)	1.0 (135,0,0)
	4	0.994 (158,1,1)	0.976 (180,5,4)
Medium	5	0.954 (145,8,6)	0.981 (207,7,1)
	6	1.0 (173,0,0)	0.988 (217,5,0)
	9	0.938 (190,18,7)	0.946 (263,29,1)
	10	0.914 (154,20,9)	0.889 (196,40,9)
Hard	7	0.953 (333,20,13)	0.987 (421,7,4)
	8	0.903 (250,46,8)	0.990 (316,6,0)
	11	0.875 (186,36,11)	0.859 (211,48,21)
	12	0.895 (218,42,9)	0.833 (252,84,17)

Table 3.7: The comparison between the baseline method using bounding box and using pose estimation on individual video. Each cell value is in the format of F1(#TP,#FP,#FN).

bounding box, indicating that the initial expectation that pose estimation produces more accurate foot points is correct.

Another advantage of using pose estimation over the bounding box is that it can interpolate foot points more accurately in many cases. Figure 3.17 illustrates a scenario where a person is partially occluded while moving. Although the person is not within the overlapping region, the foot point interpolated by the bounding box falls within the overlapping region, resulting in a FP (Figure 3.17a). In contrast, using only hip and shoulder points, the foot point can still be interpolated accurately (Figure 3.17b).

Next, the cases where using pose estimation produces worse results than using the bounding box were investigated. In contrast to the initial expectations, 2 out of the 4 videos in the hard set produced significantly worse results. During the analysis of the results obtained using pose estimation, a phenomenon was observed when a person enters the overlapping region to move (and disappear) from one camera's FOV to (and appears in) another camera's FOV (see Figure 3.18). When a person entered the overlapping region, pose estimation would localize the foot point inside the overlapping region earlier than when using the bounding box. However, when a person exited the overlapping region, pose estimation would localize the foot point outside the overlapping region later than when using the bounding box. This resulted in a longer duration of a person's movement within the overlapping region when using pose estimation. Consequently, if two different people were missed on two different cameras in succession, using pose estimation resulted in more FP.



(a) **(b)**

Figure 3.15: An example where using pose estimation produced better results than using the bounding box. **a)** using box. **b)** using pose. The green line indicates a true positive match, the red line indicates a FP match, the yellow line indicates a FN match.

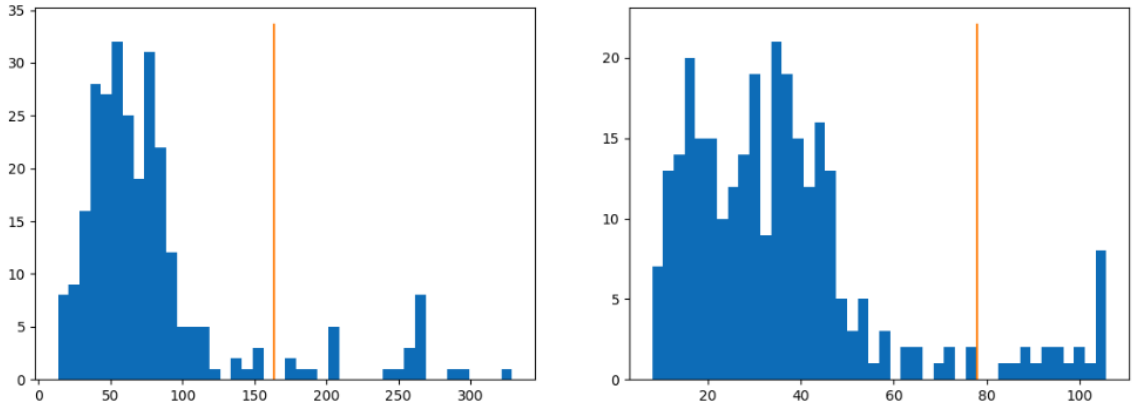
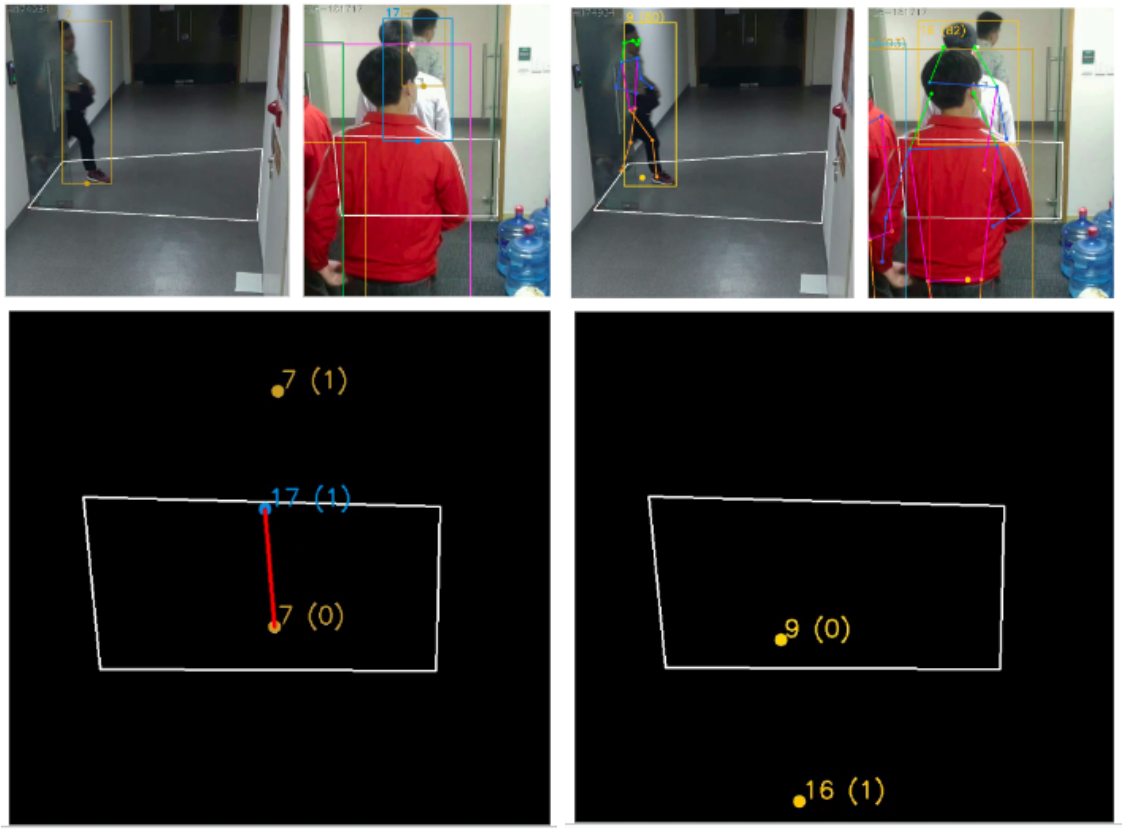
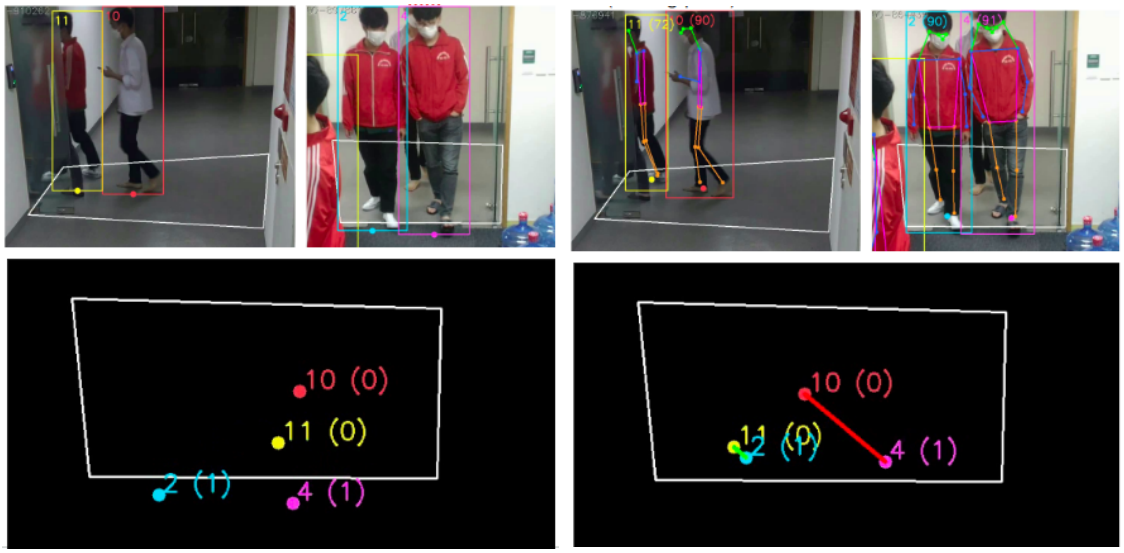


Figure 3.16: Spatial distance distribution between matched foot points in a video. **a)** using box. **b)** using pose. The x-axis represents the spatial distance of matched pairs. The y-axis represents the number of matched pairs. The vertical seam represents the upper bound by IQR(25, 75).



(a) **(b)**

Figure 3.17: An example where pose estimation interpolates foot points more accurately than the bounding box, even when the person is partially occluded. **a)** interpolation using box. **b)** interpolation using pose. The green line indicates a true positive match, the red line indicates a FP match, the yellow line indicates a FN match.



(a) **(b)**

Figure 3.18: A common example in which using pose estimation produces worse results than using the bounding box. **a)** using box. **b)** using pose. The green line indicates a true positive match, the red line indicates a FP match, the yellow line indicates a FN match.

This issue has already been discussed in section 3.3.2.2. This issue, though, can be addressed by applying FP filtering and window-based mapping.

Indeed, after applying FP filtering and window-based mapping, the performance on medium and hard videos improved significantly and outperformed using the bounding box (see Table 3.8).

3.4 Comparative evaluation of MCT by STA vs. MCT by Re-ID

This section provides a comparison of the matching results between multiple cameras using two methods: Re-ID and the proposed method based on STA. Eventually, track level matching is still needed to do a comparison. However, as mentioned in Section 3.3.1, it is very difficult to match tracks between cameras due to the ID switch problem with tracks obtained from the output of SCT. Therefore, in order to obtain comparison results at the track level, the IDs of the tracks have to be manually corrected before performing the matching.

As mentioned in Section 2.1.1, in practical Re-ID applications, a verification task is performed to determine whether a query image and a gallery image belong to the same person. If the distance between them is less than a certain threshold, the two individuals are considered the same person. The results obtained from the entire video dataset show that all individuals wearing the same uniform are

Video set	ID	Baseline with box	Baseline with pose	IQR(20, 80) size = 11 with box	**IQR(25, 75) size = 7 with pose**
Easy		0.986 (511,7,7)	0.985 (665,11,9)	0.986 (509,5,9)	0.985 (657,3,17)
Medium		0.951 (662,46,22)	0.950 (883,81,11)	0.969 (662,20,22)	0.991 (886,8,8)
Hard	11	0.875 (186,36,11)	0.859 (211,48,21)	0.898 (158,18,18)	0.894 (200,15,32)
	12	0.895 (218,42,9)	0.833 (252,84,17)	0.948 (219,16,8)	0.938 (253,17,16)
	all	0.913 (966,144,41)	0.928 (1200,145,42)	0.938 (963,83,44)	0.960 (1179,36,63)
Total		0.941 (2139,197,70)	0.948 (2748,237,62)	0.959 (2134,108,75)	0.976 (2722,47,88)

Table 3.8: The comparison between using the bounding box and using pose estimation after applying FP filtering and window-based mapping. Each cell value is in the format of F1(#TP,#FP,#FN).

Video set	Re-ID	**STA**
Easy	0.5 (32 - 64 - 0)	1.0 (32 - 0 - 0)
Medium	0.348 (57 - 211 - 2)	0.982 (57 - 0 - 2)
Hard	0.380 (54 - 176 - 0)	0.991 (53 - 0 - 1)

Table 3.9: The comparison between Re-ID and the proposed STA method. Each cell value is in the format of F1(#TP,#FP,#FN). Note that this evaluation was done at track-level after fixing ID switch cases.

considered the same person, as illustrated in Figure 3.19.

On the other hand, when using the proposed STA method, all tracks are matched correctly across all recorded videos (Figure 3.20). To generate matching results at the track level from matching results at the frame level when two IDs competing for one ID on another camera, the ID with more matched frames will be chosen as a rule of thumb. Table 3.9 summarizes the quantitative comparison between Re-ID and STA. The results demonstrate that the proposed method based on STA is more effective in matching tracks between cameras.

3.5 Summary on the findings

Previous sections presented the proposed theory and experimental results for the inter-camera association solution using STA. To evaluate the solution's practical

Figure 3.19: The matching results between 3 cameras using Re-ID. The 3 frames were captured at the same time. Tracks with the same ID are considered to belong to the same person.

Figure 3.20: The matching results between two cameras using the proposed STA method. Tracks with the same ID are considered to belong to the same person.

applicability, the following questions from Section 3.1 need answering:

1. What are the underlying issues with the proposed method at the frame-level?
2. In which scenarios does the method work/fail?
3. When should the method (not) be used?
4. If the method is used, how should it be used?

Briefly, mapping at the frame level is recommended as it is difficult to find a suitable matching method at the track level if there are many ID switches. Moreover, the evaluation results at the frame level are more reliable.

The frame-level mapping can be inaccurate due to missing detection and inaccurate foot point estimation, which originate from the limitations of Single Camera Tracking. When individuals are occluded, each on different cameras, this can cause the mapping method to produce false positive matches. The method also requires accurate foot point interpolation, while bounding boxes are not suitable for this task. These two issues become more prevalent as complexity increases, for example, when the overlap area is small or when the number of people moving simultaneously through the overlap area increases.

FP filtering is necessary to reduce the number of false positives due to missing detection. It has a significant and positive impact on complex cases, although it may have a negative but negligible impact on simple cases. Window-based mapping can mitigate some limitations of FP filtering and support it in reducing both

false positives and false negatives. It has a significant and positive impact on most cases, particularly complex ones. When combining them, a better performance can be achieved, and it may be necessary to loosen the parameters of each technique compared to using only one of them.

Pose estimation can result in a more reliable foot point, but requires FP filtering to be applied. Because the distance distribution is more precise, it may be necessary to reduce the window size and increase the FP filtering threshold compared to using the bounding box.

For the the application system of employee behavior monitoring in Chapter 4, the proposed STA solution will be used for the inter-camera association step and will apply a combination of FP filtering with IQR(25, 75), window-based mapping with window size = 7, and pose estimation.

CHAPTER 4

APPLICATION SYSTEM DEVELOPMENT

4.1 Overview

One of the main interests in AI research is its practical application. While some techniques achieve high levels of accuracy in research, they may not always be feasible in real-world applications due to some limitations such as processing speed or model size. I want to evaluate the practical applicability of my thesis research work by building a system that has real industrial demand.

Therefore, as pointed out in Section 1.3, my thesis work includes implementing a software application to monitor employee productivity, using the proposed tracking method to record employee staying intervals in their designated working areas.

Since my goal is not to build a software system that fully meets all business process requirements, some features are either simplified or omitted (such as updating personal information, security, or searching). Instead, the software development process focuses on two key tasks:

1. Developing only the most basic features enough for the system to operate. The major effort is put into few important usecases which illustrate my research results and are critical to monitor employee productivity.
2. Optimizing the system for real-time tracking operation. This enables the system to perform functions such as sending immediate notifications to specified managers when an employee is absent from a monitored area for a too long time.

In particular, the system needs to support the following two important functions:

1. Estimating productivity of employees.
2. Notifying managers instantly when employees are late or leave their work position for too long.

4.2 Software Requirements Analysis

This section provides a detailed description of the functional and non-functional requirements for the Employee Behavior Monitoring System.

4.2.1 Functional requirements

4.2.1.1 Actors

The system has 4 actors, namely Guest, Staff, Manager, and System Admins.

No	Actor	Description
1	Guess	The user who has not signed in the system
2	Staff	The staff in a company whose productivity (i.e., staying in specific areas) needs to be tracked and monitored
3	Manager	The manager who tracks and monitors their employees
4	System Admin	The user who manages the system accounts

Table 4.1: List of actors interacting with the system.

4.2.1.2 Modelling business processes

The following activity diagrams describe some key business processes, including:

- user logging into the system (Figure 4.1)
- staff registering for a work shift (Figure 4.2)
- manager viewing real-time cameras (Figure 4.3)
- manager viewing staff productivity reports (Figure 4.4)

The *user logging into the system* process is described as in Figure 4.1. In addition to authenticating login information, the system will use the result of the tracking service to observe if anyone is standing at the check-in region at the time it receives the login signal. If someone appears at the check-in region at the time

the system receives a login signal from a staff's account, the system will assign that track ID to that account.

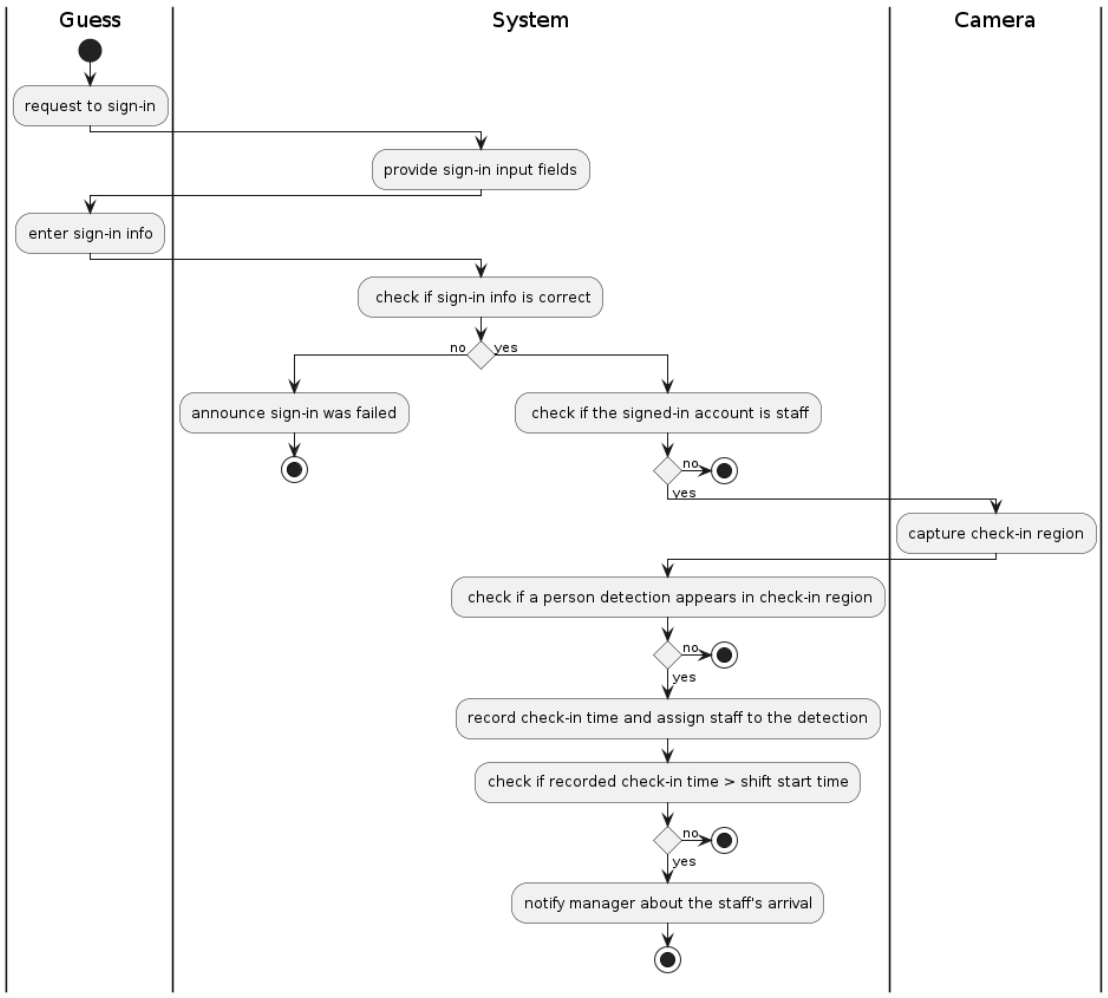


Figure 4.1: Activity diagram for the *user logging into the system* process.

The *staff registering for a work shift* process is described as in Figure 4.2. Each shift that an employee registers for will include a time slot. The day shift will consist of the start and end times, which will be used to determine if an employee is late and to calculate their on-the-job-region staying time.

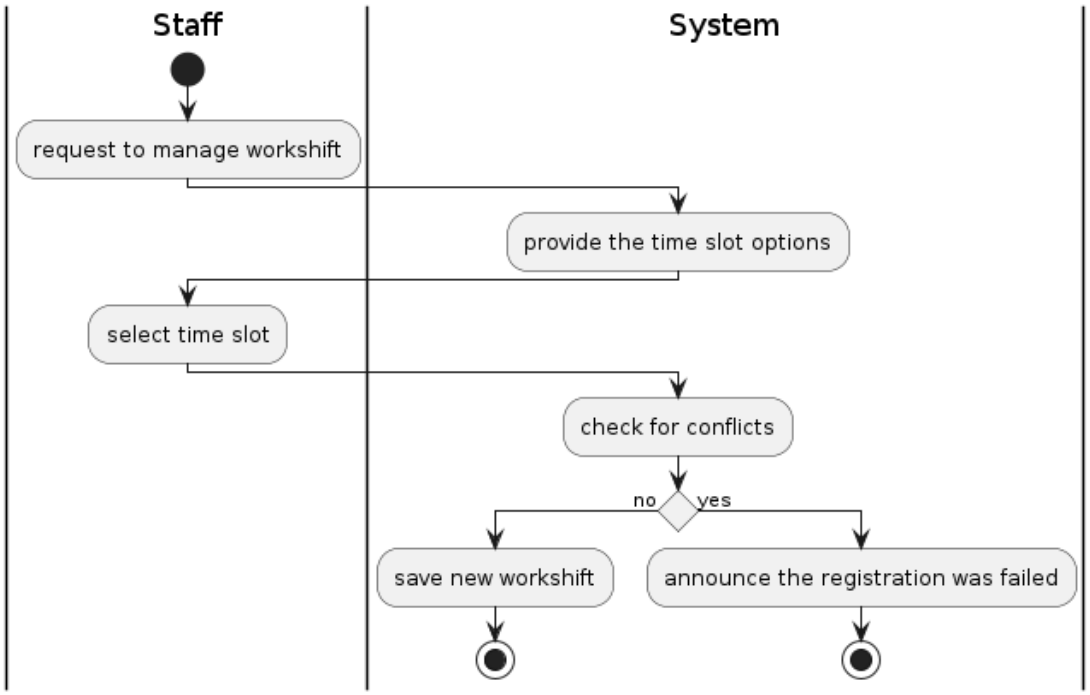


Figure 4.2: Activity diagram for the *staff registering for a work shift* process.

The *manager viewing real-time cameras* process is described as in Figure 4.3. When a manager requests to view the cameras, the system will first ensure that the tracking service is running. Before displaying the images on the screen, the system will draw the detection box and username onto the frames for visual reference.

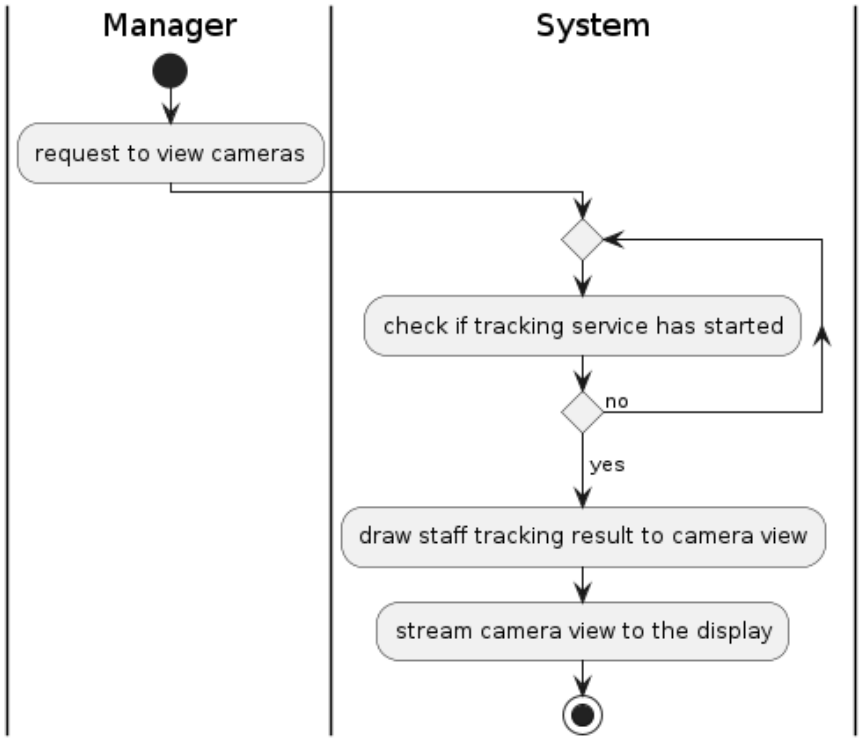


Figure 4.3: Activity diagram for the *manager viewing real-time cameras* process.

The *manager viewing staff productivity reports* process is described as in Figure 4.4. The productivity report includes statistics on the number of times a staff arrives late and the percentage of time he stays in the designated region.

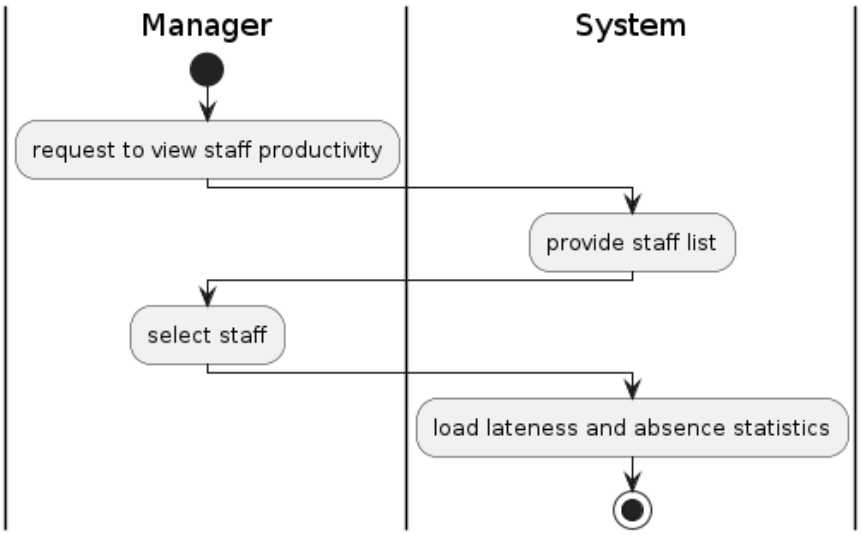


Figure 4.4: Activity diagram for the *manager viewing staff productivity reports* process.

4.2.1.3 Usecases analysis

Below is a list of the usecase diagrams, including:

- General usecase diagram (Figure 4.5)
- Decomposed usecase diagram for Manager (Figure 4.6)

There are 2 important usecases extended for actor Manager, which are *View staff productivity* and *Get notified about staff's irregular behaviors*. The proposed MCT system in the research section is necessary to function these two usecases.

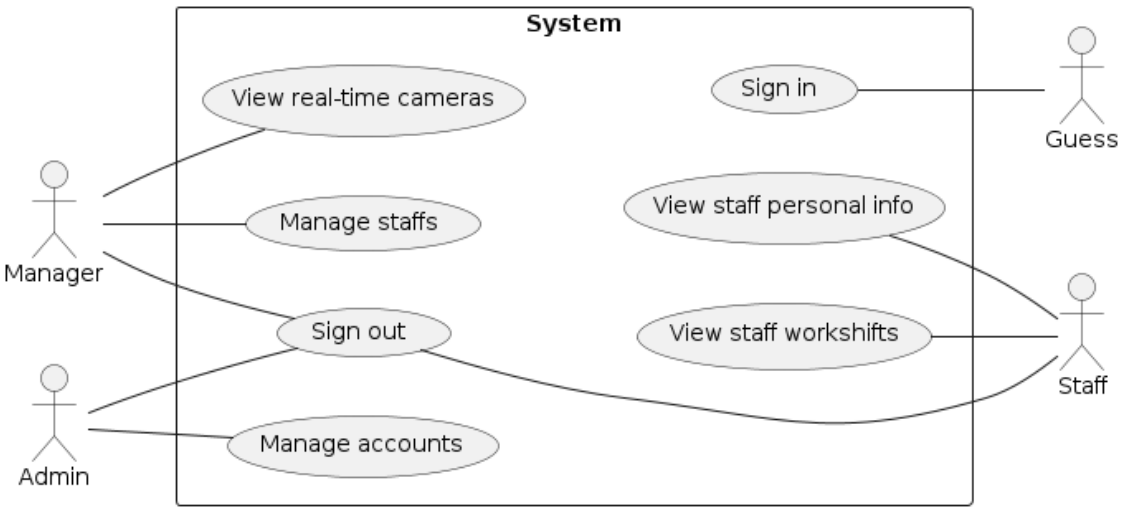


Figure 4.5: General usecase diagram.

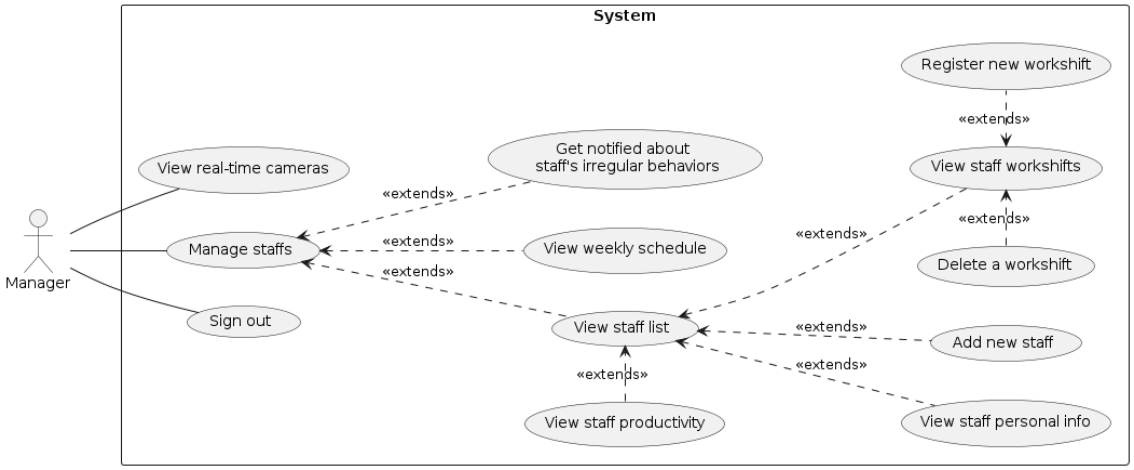


Figure 4.6: Decomposed usecase diagram for Manager.

The specifications of the usecases are provided in the below tables, from Table 4.2 to Table 4.13.

Usecase code	UC01
Usecase name	sign in
Description	Allow Guess to log in to the system in order to use its functions.
Actors	Guess
Trigger event	Guess selects the login command from the user interface.
Pre-conditions	No pre-condition
Post-condition	The actor is directed to an interface screen that corresponds to their account role.
Main flow	1. Guess selects the login function. 2. System displays a login interface including a username and a password field. 3. Guess enters their username and password. 4. System verifies the login information. 5. System records sign-in time and sends the signal to its tracking service. 6. System directs the user to the interface corresponding to their role.
Alternative flow	1. Guess enters incorrect username and/or password. 2. System informs Guess of the incorrect input. 3. Proceed to step 3 of the main event flow.

Table 4.2: Specification of the usecase *sign in*.

Usecase code	UC02
Usecase name	sign out
Description	Allow the user to sign out of the system.
Actors	Staff, Manager, System Admin
Trigger event	Guess selects the sign out option from the interface.
Pre-conditions	The actor has successfully signed in.
Post-condition	The actor successfully signs out of the system and becomes guess
Main flow	1. The actor selects the sign out function. 2. System terminates the sign out session and redirects the user to the login page.
Alternative flow	No alternative flow

Table 4.3: Specification of the usecase *sign out*.

Usecase code	UC03
Usecase name	view real-time cameras
Description	Allow the manager to view the synchronized cameras in real-time.
Actors	Manager
Trigger event	Manager chooses to view cameras from the interface.
Pre-conditions	Tracking service has started
Post-condition	The camera streams are fed on to the screen.
Main flow	1. Manager selects the view cameras function. 2. System waits until tracking service has started. 3. System draw tracking result on to frame images. 4. System streams the camera view to the display.
Alternative flow	1. Tracking service has not started for 5 seconds. 2. System announces that the request was failed.

Table 4.4: Specification of the usecase *view real-time cameras*.

Usecase code	UC05
Usecase name	get notified about staff's irregular behaviors
Description	Allow Manager to view instant alerts about staff lateness and absence during work shifts.
Actors	Manager
Trigger event	Manager chooses to view messages from the interface.
Pre-conditions	No pre-condition
Post-condition	The list of messages is displayed on the screen.
Main flow	1. Manager selects the view messages function. 2. System loads the messages associated with the user account and displays them on the screen.
Alternative flow	No alternative flow

Table 4.5: Specification of the usecase *get notified about staff's irregular behaviors*.

Usecase code	UC06
Usecase name	view weekly schedule
Description	Allow Manager to view weekly schedule.
Actors	Manager
Trigger event	Manager chooses to view weekly schedule from the interface.
Pre-conditions	No pre-condition
Post-condition	A timetable with work shifts and staff registration is displayed on the screen.
Main flow	1. Manager selects the view weekly schedule function. 2. System loads the registered work shifts of all staffs and displays on the screen.
Alternative flow	No alternative flow

Table 4.6: Specification of the usecase *view weekly schedule*.

Usecase code	UC07
Usecase name	view staff personal info
Description	Allow manager to view staff personal info, or allow staff to view their own info.
Actors	Manager, Staff
Trigger event	The actor chooses to view staff info from the interface.
Pre-conditions	Staff info exists in database.
Post-condition	Staff personal info is displayed on the screen.
Main flow	1. The actor selects the view staff personal info function. 2. If user is manager, then Manager selects a staff. Else, move to step 3. 3. System loads the staff personal info and displays on the screen.
Alternative flow	No alternative flow

Table 4.7: Specification of the usecase *view staff personal info*.

Usecase code	UC10
Usecase name	view staff list
Description	Allow manager to view staff list.
Actors	Manager
Trigger event	Manager chooses to view staff list from the interface
Pre-conditions	Staff exists in database.
Post-condition	The list of staff is displayed on the screen.
Main flow	1. Manager selects the view staff list function. 2. System loads the list of staffs and displays on the screen.
Alternative flow	No alternative flow

Table 4.8: Specification of the usecase *view staff list*.

Usecase code	UC11
Usecase name	add new staff
Description	Allow manager to add new staff.
Actors	Manager
Trigger event	Manager chooses to add new staff from the interface
Pre-conditions	No pre-condition
Post-condition	A new user account is created.
Main flow	1. Manager selects the create account function. 2. System displays an interface to enter staff information. 3. Manager enters staff information. 4. System creates a new account.
Alternative flow	1. Manager enters an existing username. 2. System informs Manager of the incorrect input. 3. Proceed to step 3 of the main event flow

Table 4.9: Specification of the usecase *add new staff*.

Usecase code	UC12
Usecase name	view staff productivity
Description	Allow manager to view staff productivity reports.
Actors	Manager
Trigger event	Manager chooses to view staff productivity from the interface
Pre-conditions	Staff registered workshifts and actual behaviors exist in database.
Post-condition	The statistics on staff lateness and absence is displayed on the screen.
Main flow	1. Manager selects the view staff productivity function. 2. Manager selects a staff. 3. System loads the staff productivity and displays on the screen.
Alternative flow	No alternative flow

Table 4.10: Specification of the usecase *view staff productivity*.

Usecase code	UC08
Usecase name	view staff workshifts
Description	Allow manager to view staff registered work shift, or allow staff to view their own registered work shift.
Actors	Manager, Staff
Trigger event	The actor chooses to view staff info from the interface.
Pre-conditions	No pre-condition
Post-condition	The staff registered work shifts are displayed on the screen
Main flow	1. The actor selects the view staff registered work shifts function. 2. If user is manager, then Manager selects a staff. Else, move to step 3. 3. System loads the staff registered work shifts and displays on the screen.
Alternative flow	No alternative flow

Table 4.11: Specification of the usecase *view staff workshifts*.

Usecase code	UC14
Usecase name	register new workshift
Description	Allow Staff to register new work shift in a week.
Actors	Staff
Trigger event	Staff chooses to register new work shift from the interface.
Pre-conditions	No pre-condition
Post-condition	A new work shift is created and associated to the staff
Main flow	1. Staff selects the register work shifts function. 2. System displays an interface including a week day and a day shift selection. 3. Staff selects a week day and a day shift. 4. System creates a new work shift for the Staff.
Alternative flow	No alternative flow

Table 4.12: Specification of the usecase *register new workshift*.

Usecase code	UC15
Usecase name	delete a workshift
Description	Allow Staff to delete a work shift in a week.
Actors	Staff
Trigger event	Staff chooses to delete work shift from the interface.
Pre-conditions	No pre-condition
Post-condition	A work shift is deleted from the staff's work shift list.
Main flow	1. Staff selects the delete work shifts function. 2. System displays the list of registered work shifts. 3. Staff select a work shift. 4. System delete the work shift.
Alternative flow	No alternative flow

Table 4.13: Specification of the usecase *delete a workshift*.

4.2.2 Non-functional requirements

1. The time needed for processing and displaying frames in the usecase *UC03: view real-time cameras* must not be greater than the cameras' capturing time. This is to ensure the system can deliver real-time monitoring.
2. The cameras' capturing speed is at least 5 FPS to ensure a smooth viewing experience.
3. The system's response latency must be less than 5 seconds.

4.3 Software System Design

This section presents the software design for the Employee Behavior Monitoring System, which includes the following sections: Structure analysis, Interaction analysis, Design of the system's overall architecture, Class detailed design, User Interface Design, and Data Design.

4.3.1 Structure analysis

4.3.1.1 UC01 (sign in)

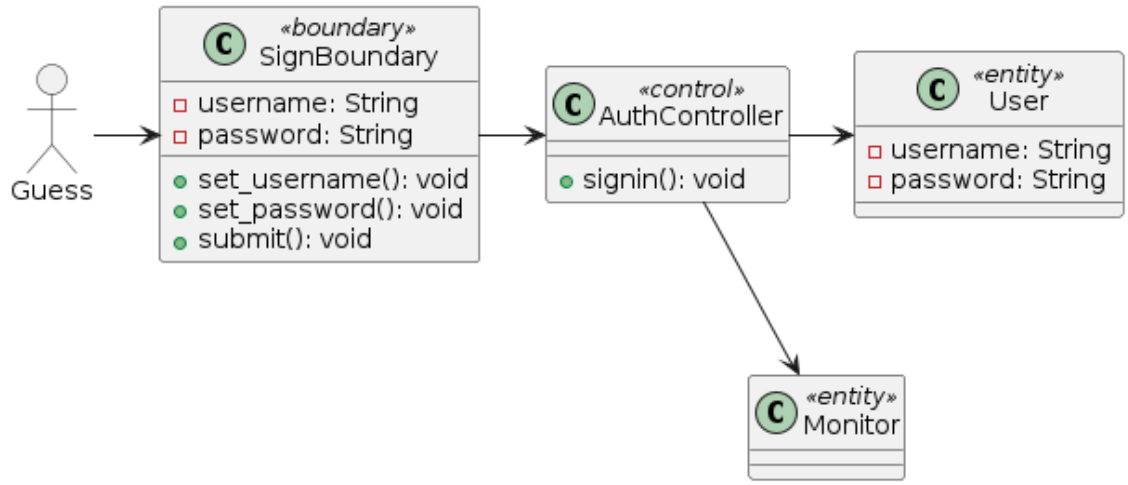


Figure 4.7: Class diagram for the use case UC01 (*sign in*).

4.3.1.2 UC06 (view weekly schedule)

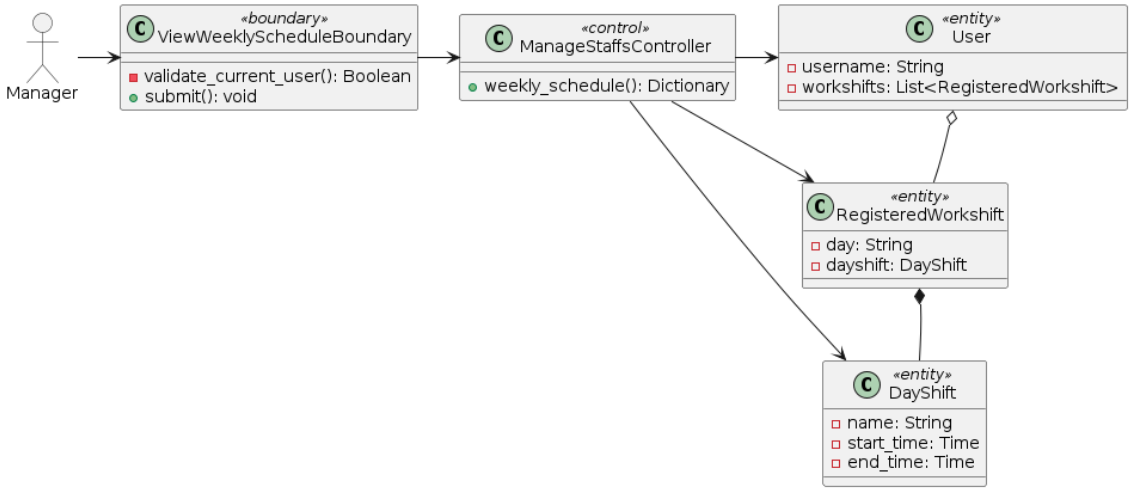


Figure 4.8: Class diagram for the use case UC06 (*view weekly schedule*).

4.3.1.3 UC10 (view staff list)

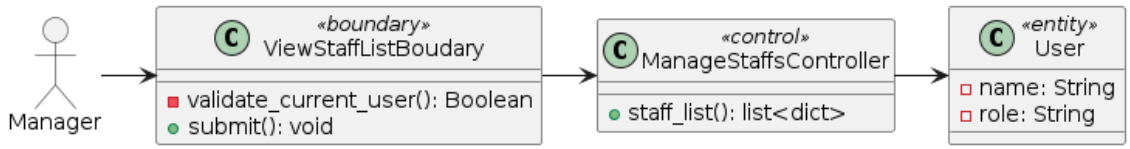


Figure 4.9: Class diagram for the use case UC10 (*view staff list*).

4.3.1.4 UC11 (add new staff)

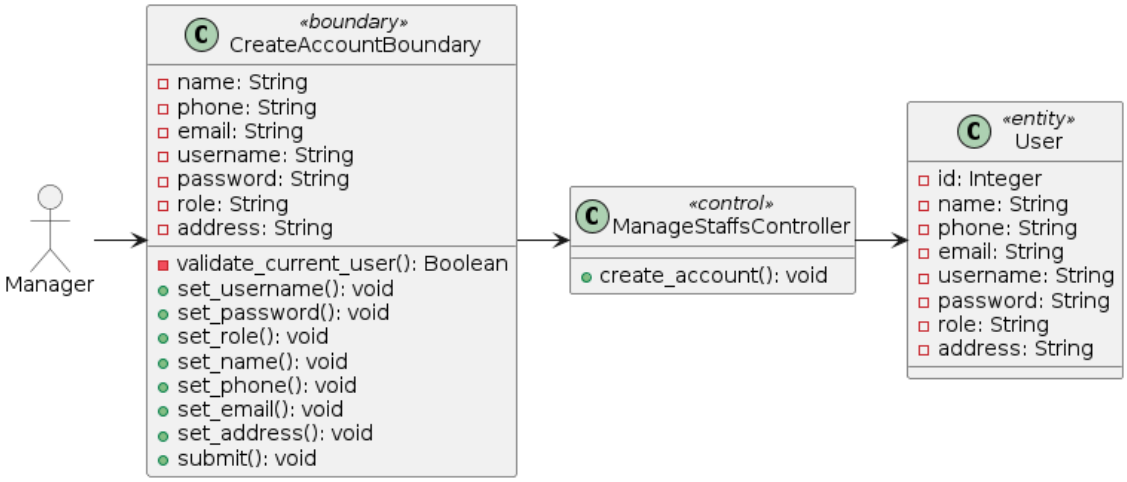


Figure 4.10: Class diagram for the use case UC11 (*add new staff*).

4.3.1.5 UC07 (view staff personal info)

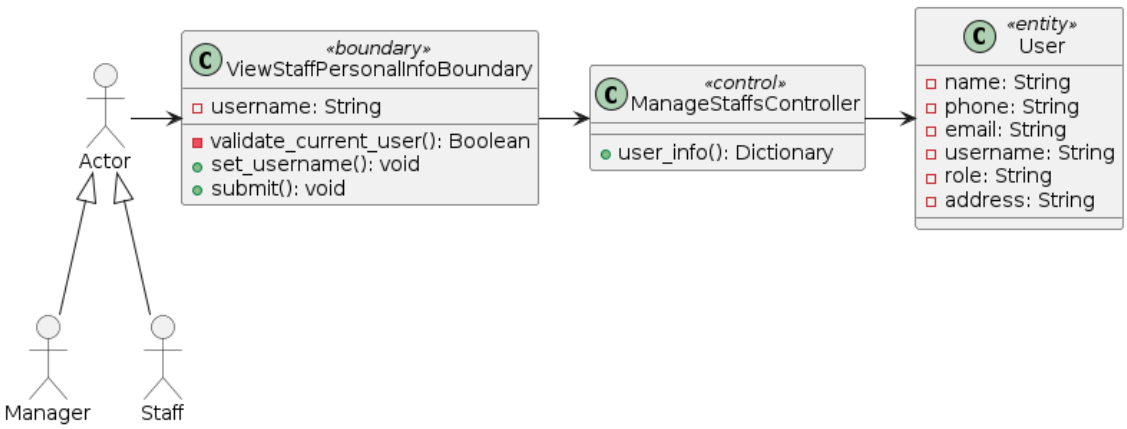


Figure 4.11: Class diagram for the use case UC07 (*view staff personal info*).

4.3.1.6 UC08 (view staff workshifts)

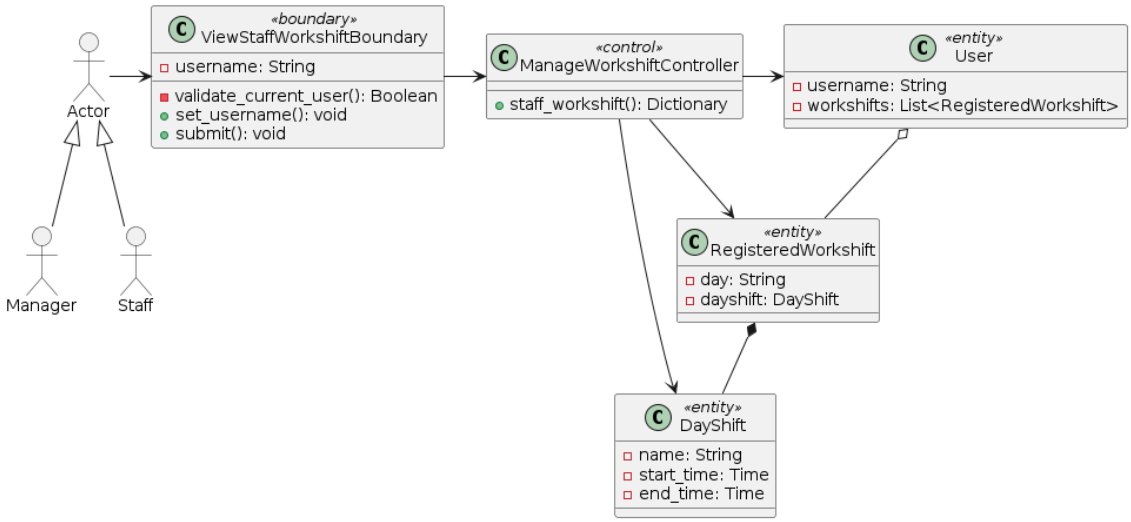


Figure 4.12: Class diagram for the use case UC08 (*view staff workshifts*).

4.3.1.7 UC12 (view staff productivity)

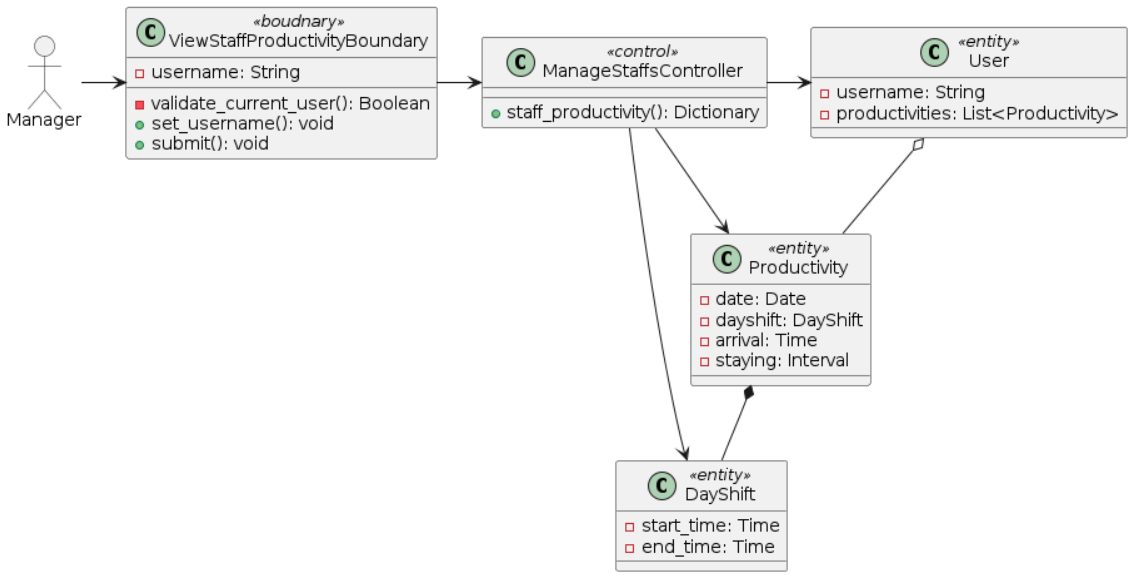


Figure 4.13: Class diagram for the use case UC12 (*view staff productivity*).

4.3.1.8 UC05 (get notified about staff's irregular behaviors)

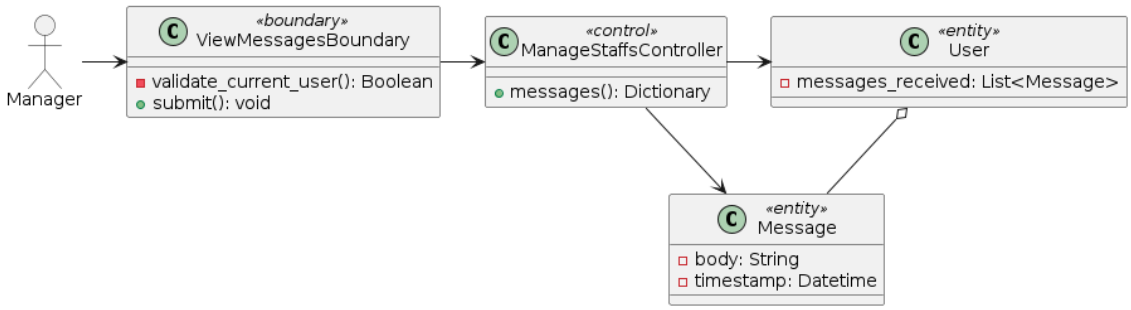


Figure 4.14: Class diagram for the use case UC05 (*get notified about staff's irregular behaviors*).

4.3.1.9 UC03 (view real-time cameras)

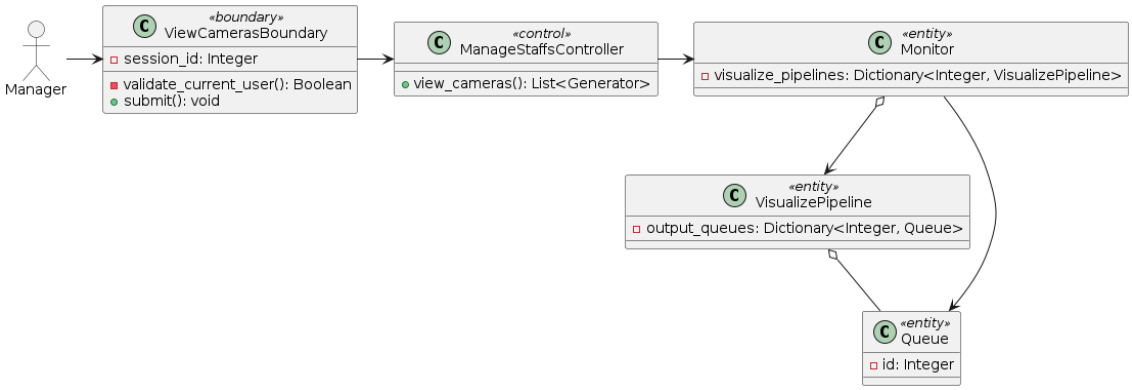


Figure 4.15: Class diagram for the use case UC03 (*view real-time cameras*).

4.3.1.10 UC14 (register new workshift)

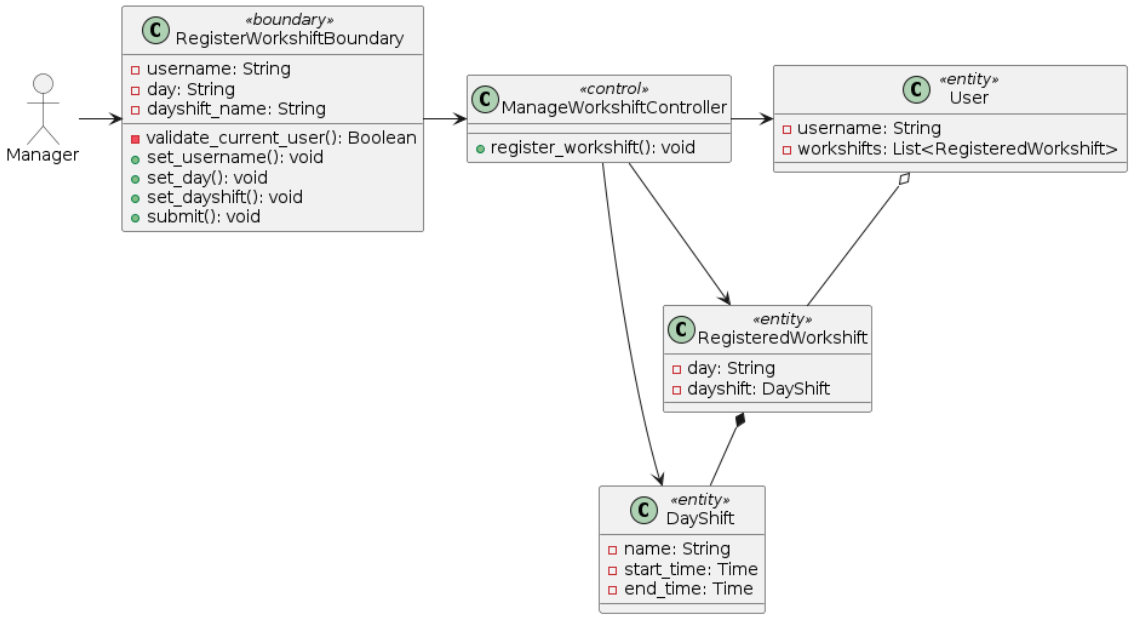


Figure 4.16: Class diagram for the use case UC14 (*register new workshift*).

4.3.1.11 UC15 (delete a workshift)

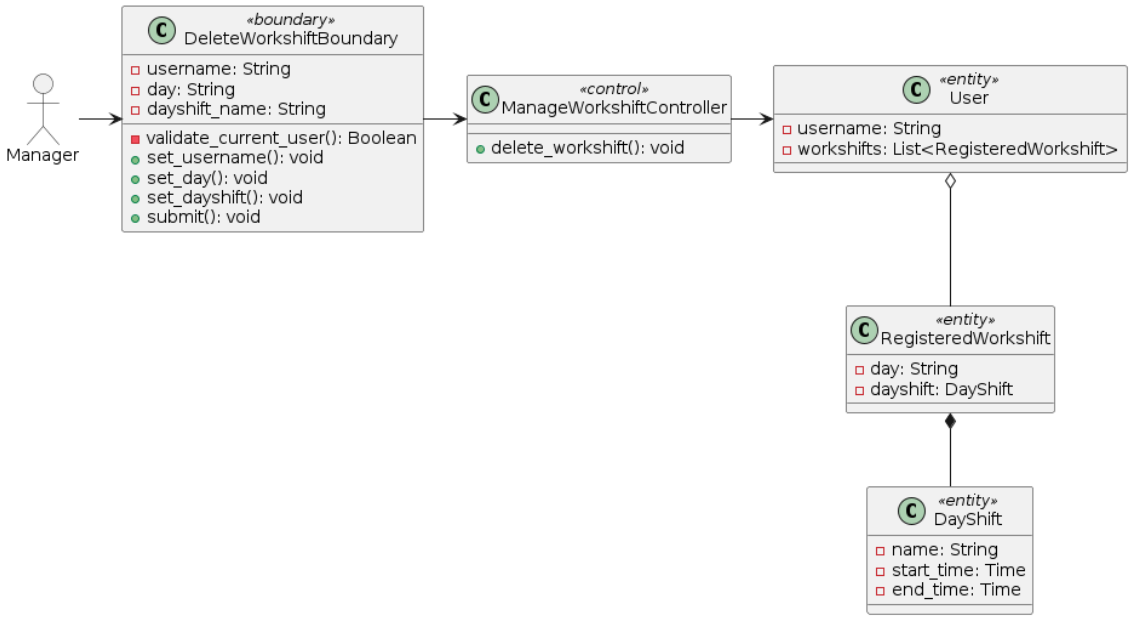


Figure 4.17: Class diagram for the use case UC15 (*delete a workshift*).

4.3.2 Interaction analysis

4.3.2.1 UC01 (sign in)

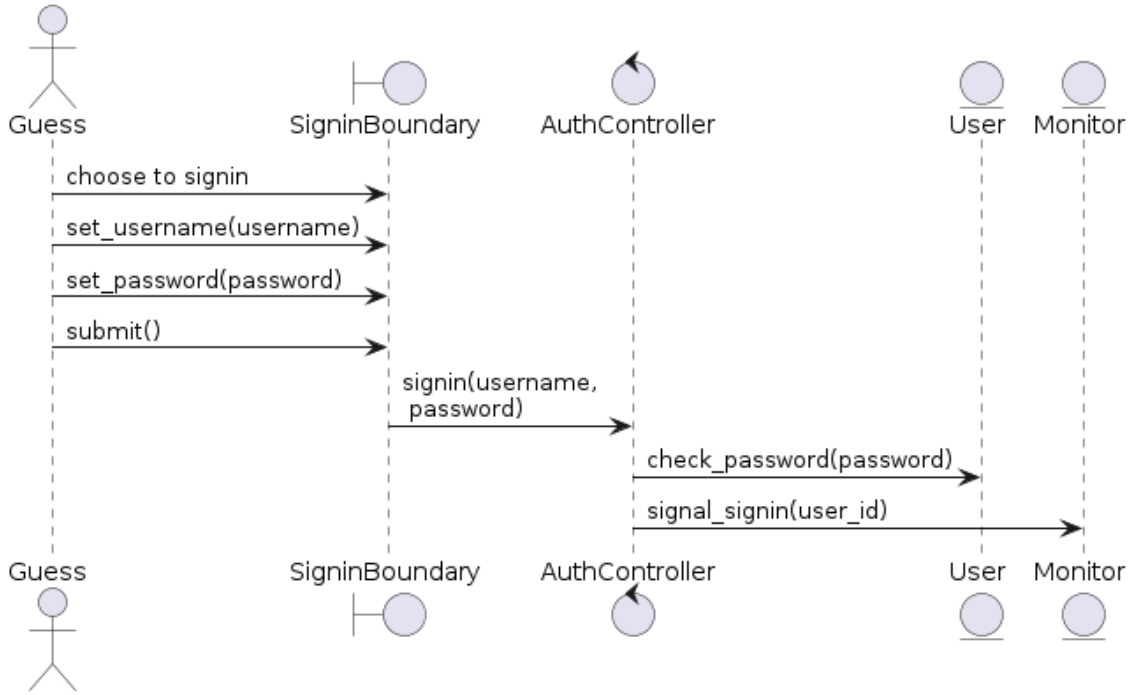


Figure 4.18: Sequence diagram for the use case UC01 (*sign in*).

4.3.2.2 UC06 (view weekly schedule)

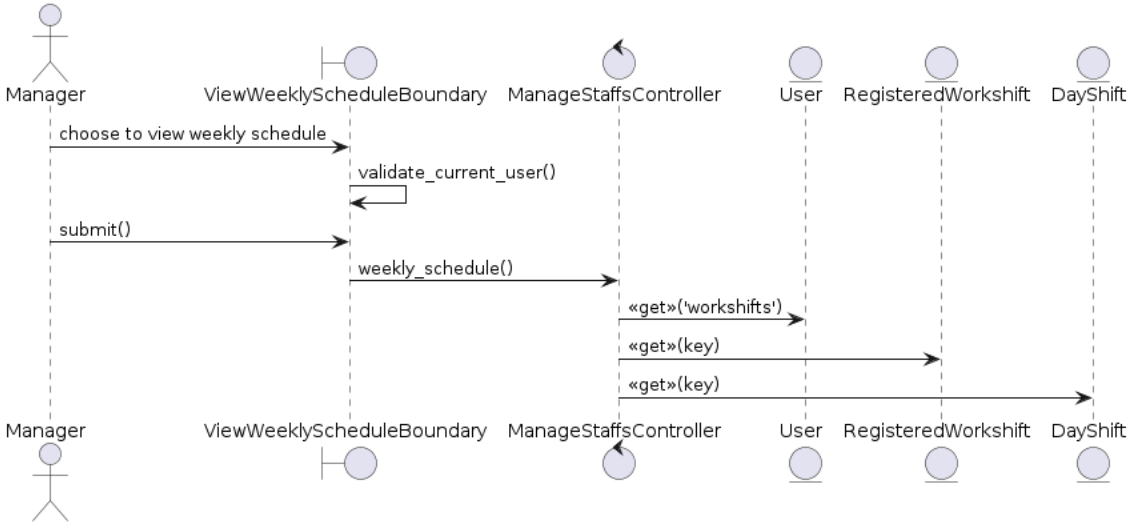


Figure 4.19: Sequence diagram for the use case UC06 (*view weekly schedule*).

4.3.2.3 UC10 (view staff list)

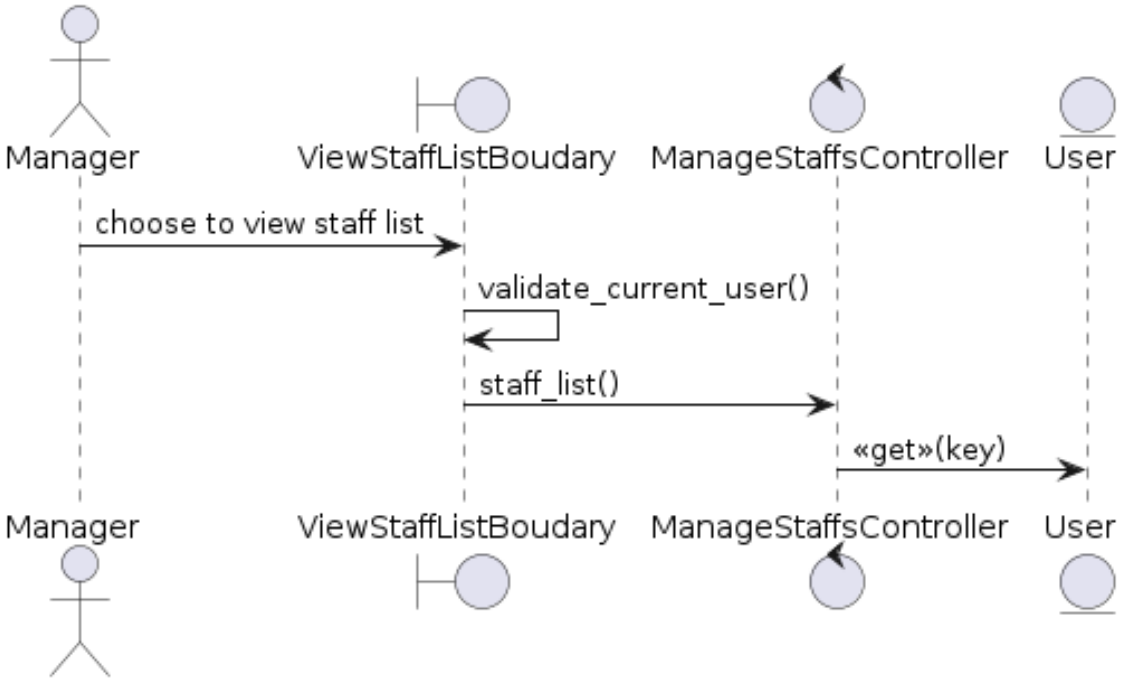


Figure 4.20: Sequence diagram for the use case UC10 (*view staff list*).

4.3.2.4 UC11 (add new staff)

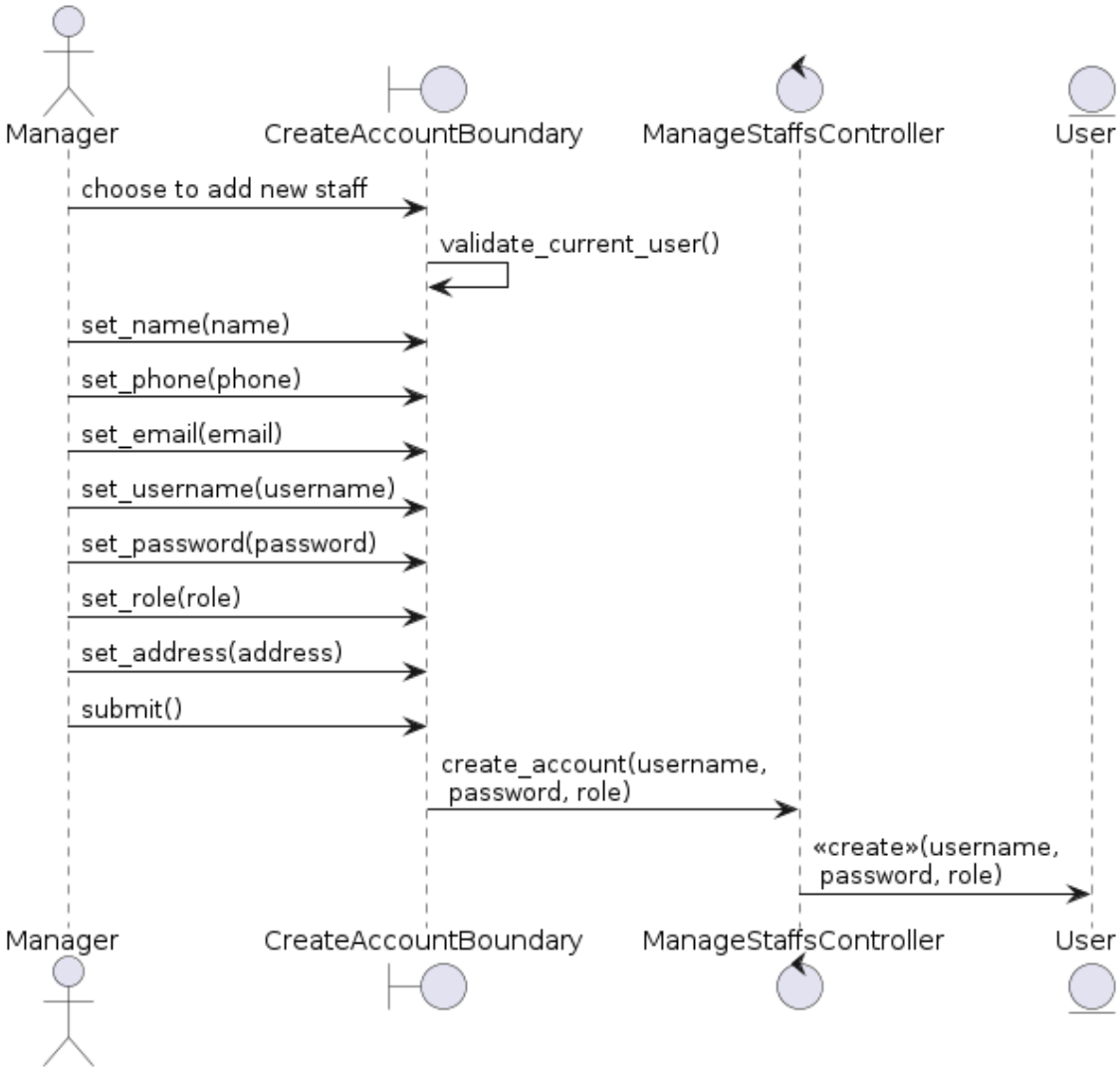


Figure 4.21: Sequence diagram for the use case UC11 (*add new staff*).

4.3.2.5 UC07 (view staff personal info)

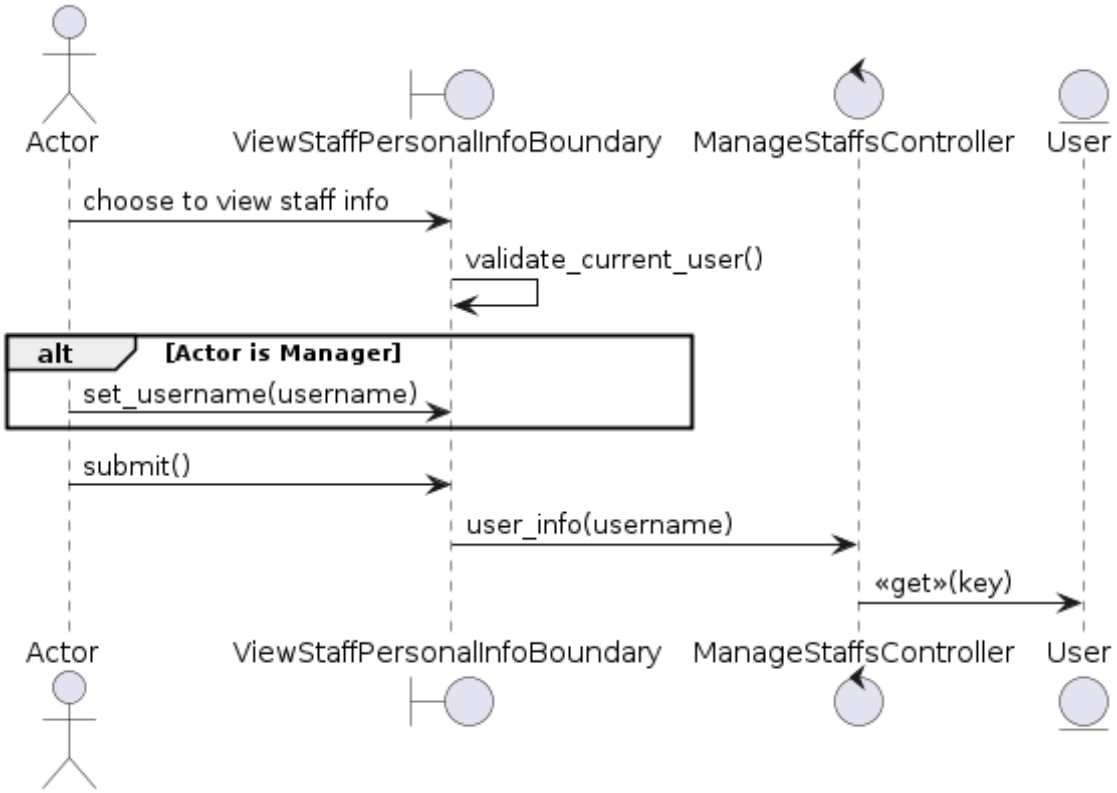


Figure 4.22: Sequence diagram for the use case UC07 (*view staff personal info*).

4.3.2.6 UC08 (view staff workshifts)

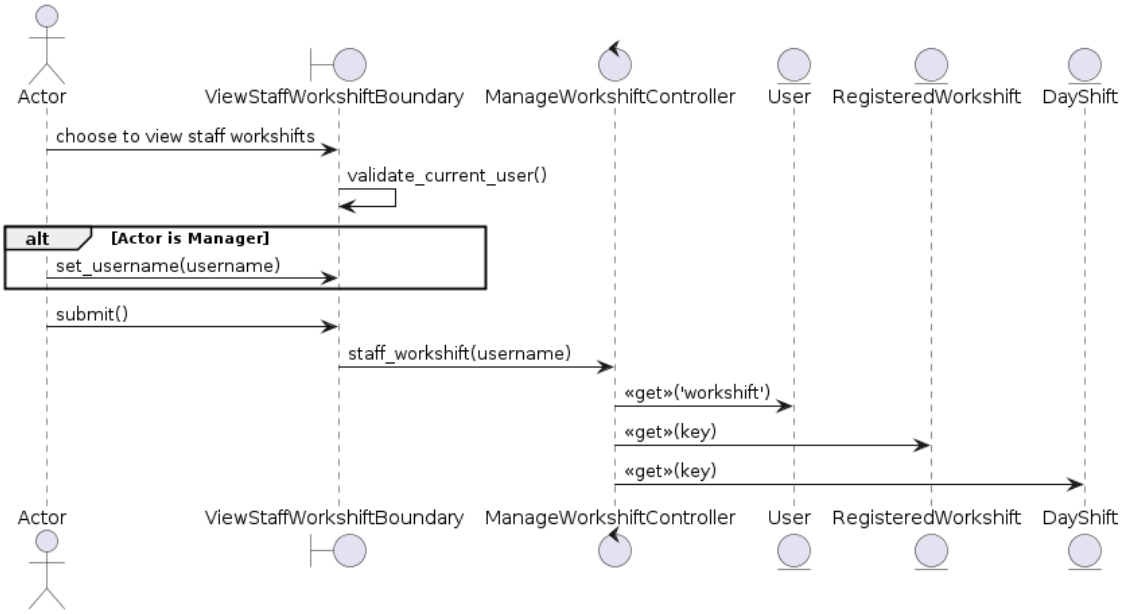


Figure 4.23: Sequence diagram for the use case UC08 (*view staff workshifts*).

4.3.2.7 UC12 (view staff productivity)

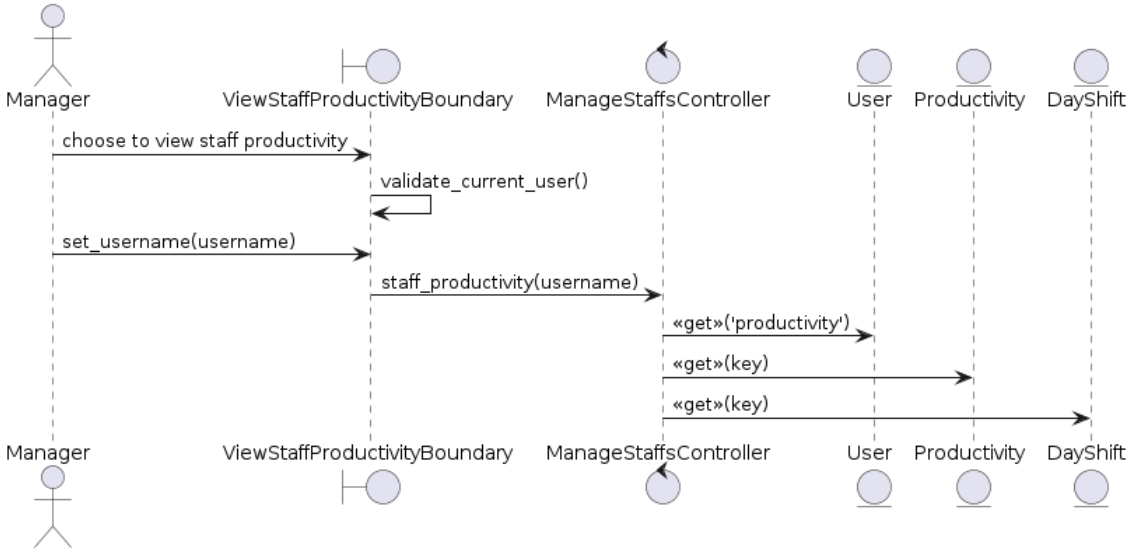


Figure 4.24: Sequence diagram for the use case UC12 (*view staff productivity*).

4.3.2.8 UC05 (get notified about staff's irregular behaviors)

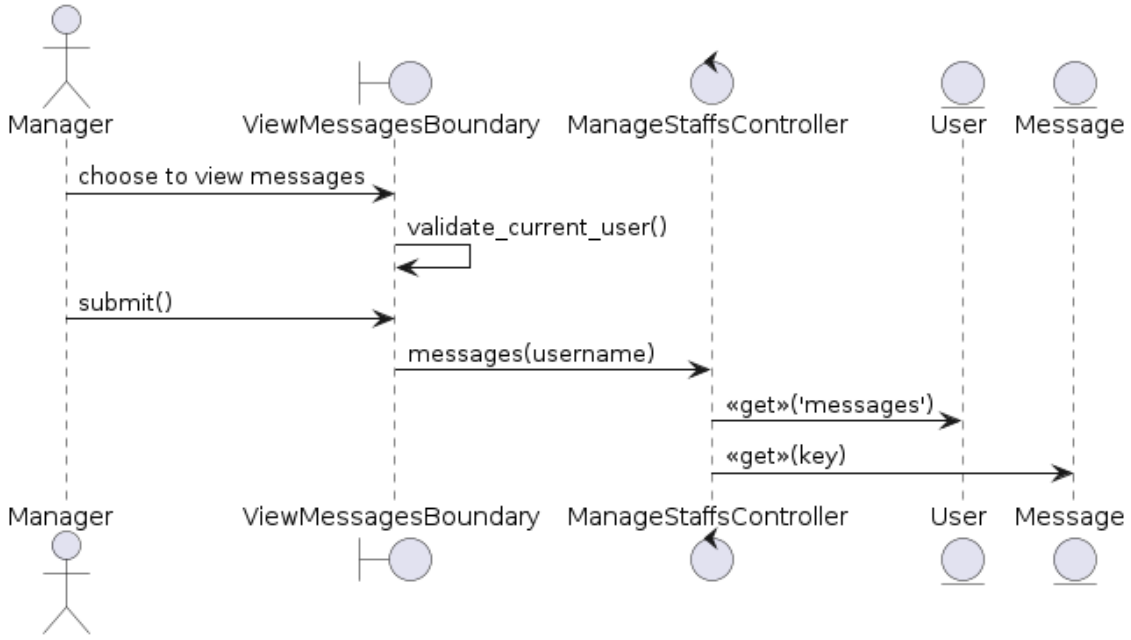


Figure 4.25: Sequence diagram for the use case UC05 (*get notified about staff's irregular behaviors*).

4.3.2.9 UC03 (view real-time cameras)

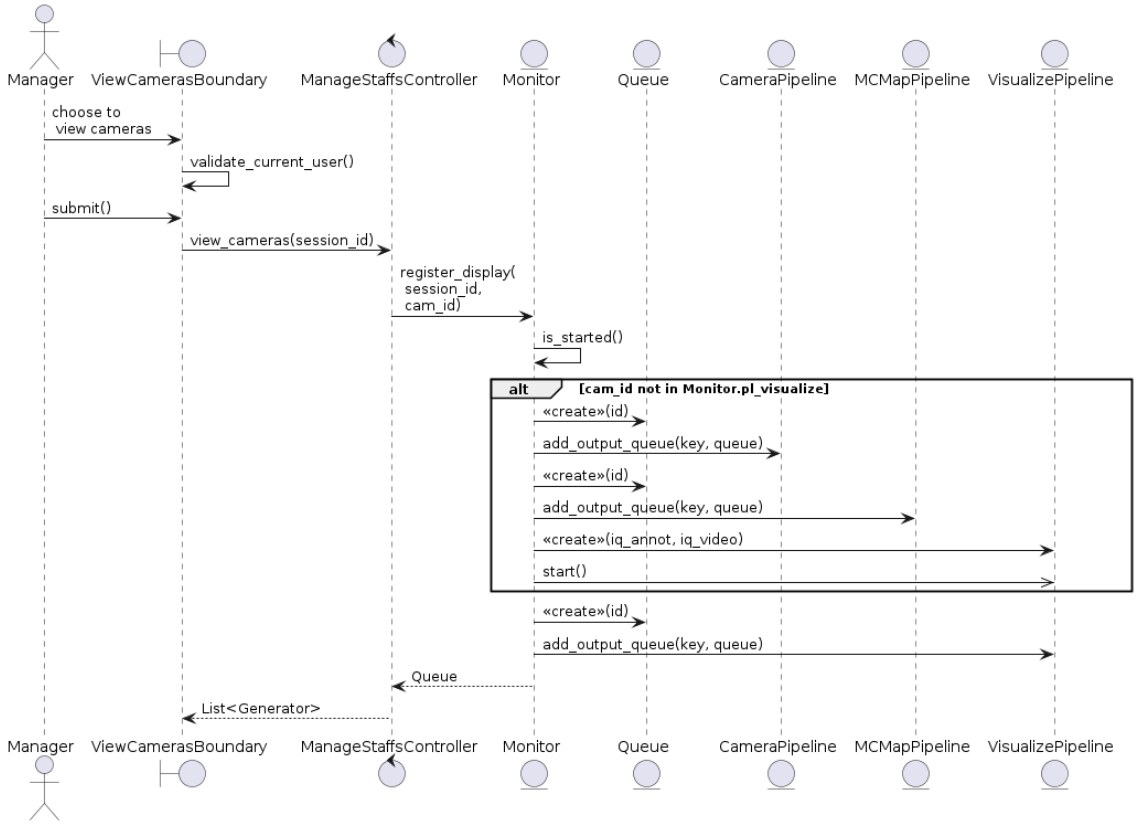


Figure 4.26: Sequence diagram for the use case UC03 (*view real-time cameras*).

4.3.2.10 UC14 (register new workshift)

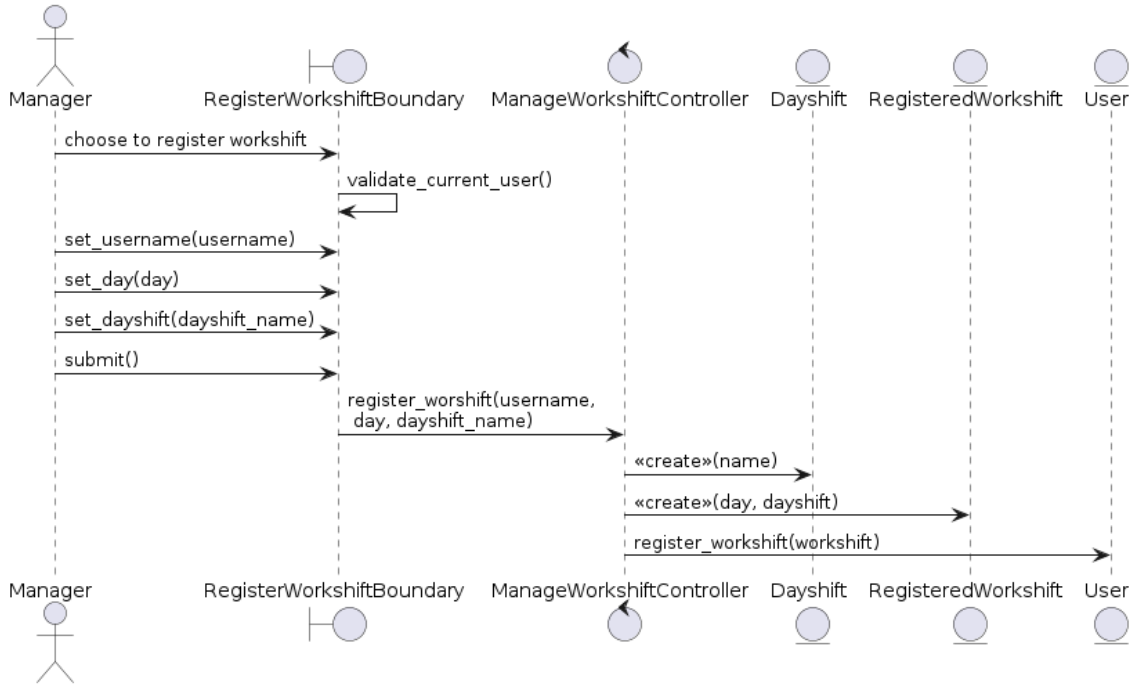


Figure 4.27: Sequence diagram for the use case UC14 (*register new workshift*).

4.3.2.11 UC15 (delete a workshift)

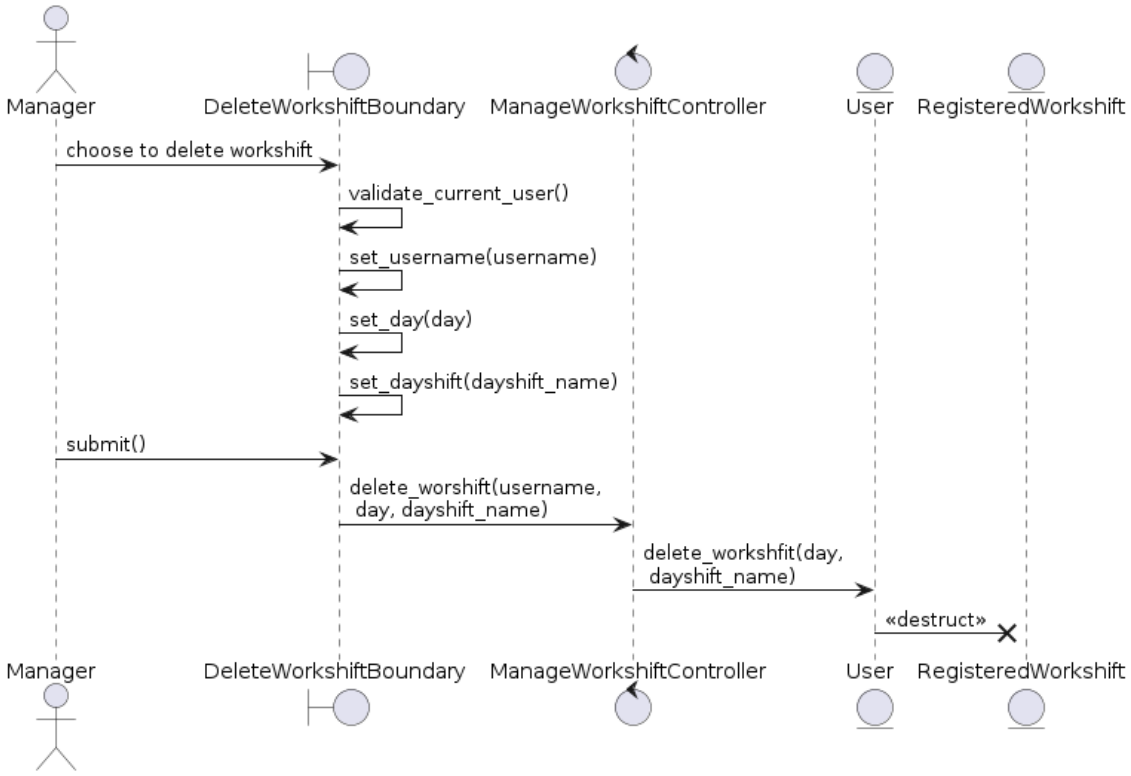


Figure 4.28: Sequence diagram for the use case UC15 (*delete a workshift*).

4.3.3 Design of the system's overall architecture

4.3.3.1 Package diagram

The system is designed as architectural building layers that follows the MVC model as in Figure 4.29. The class diagrams for the Model - View - Control packages are described as in Figure 4.30 through Figure 4.32 respectively.

- The View contains user interfaces used for interacting with the system.
- The Controller contains controllers that handle requests.
- The Model contains data entities.

Figure 4.29: Architectural building layers of the system that follows the MVC model.

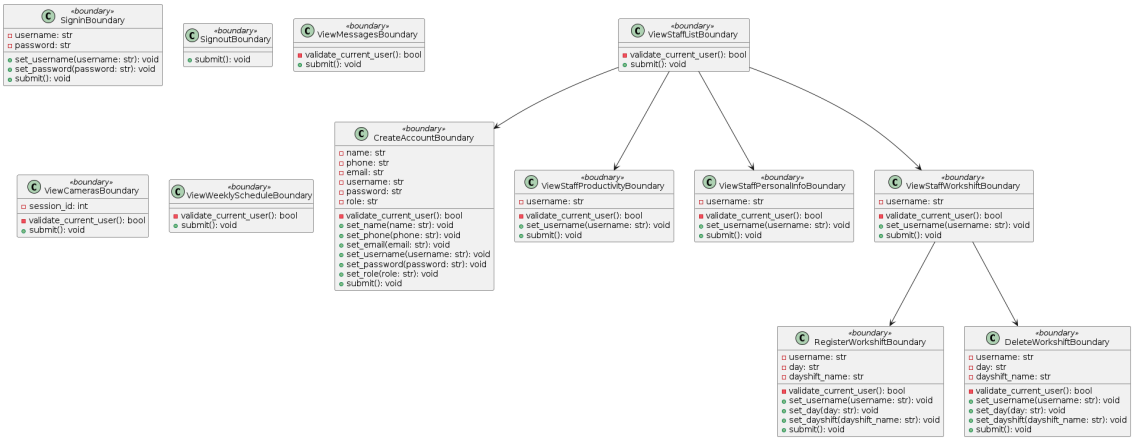


Figure 4.30: Class diagram for the View package.

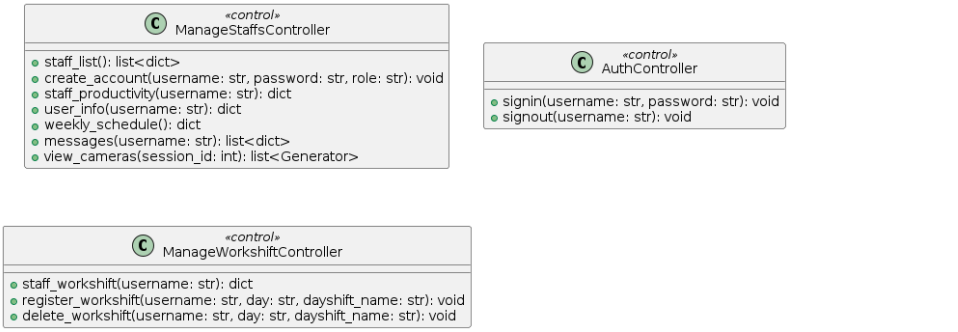


Figure 4.31: Class diagram for the Control package.

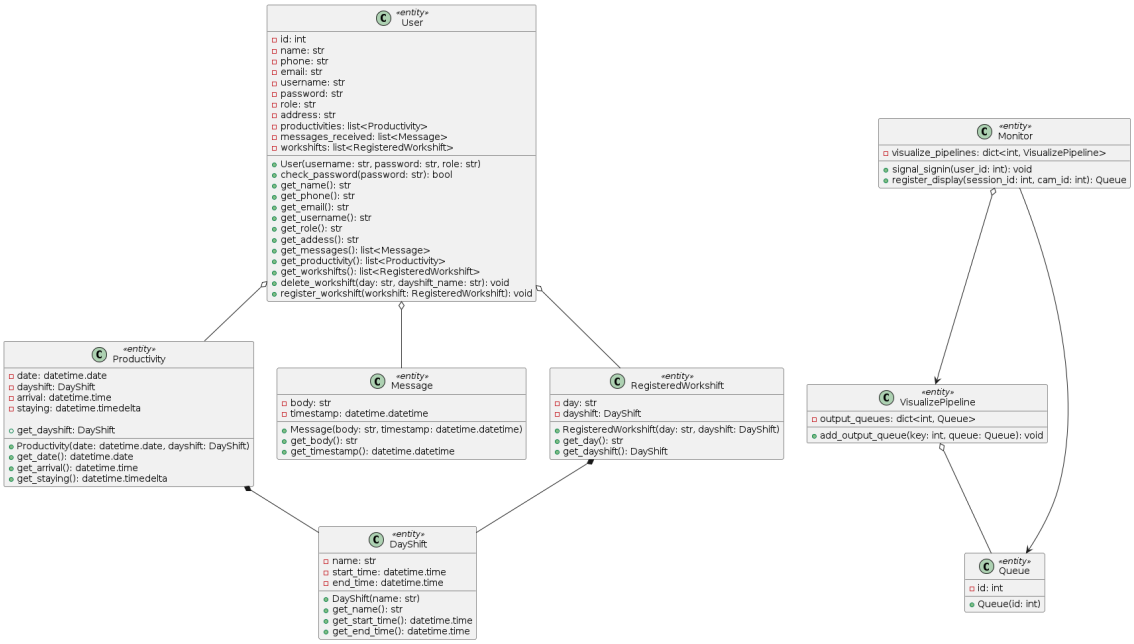


Figure 4.32: Class diagram for the Model package.

4.3.3.2 Deployment diagram

As being described in Figure 4.33, the deployment diagram includes:

- Web browser from client devices that displays the system's output to users.
- Web server that contains applications built upon Python Flask.
- The Database serving as the storage location for the system's data.

The client Web browser and the Web server communicate with each other through the HTTP/HTTPS protocol.

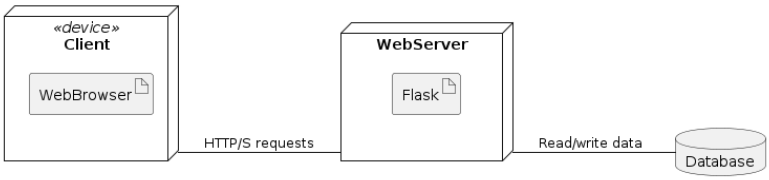


Figure 4.33: Deployment diagram for the system design.

4.3.4 Class detailed design

4.3.4.1 SigninBoundary

Purpose of use: Allow user to perform sign in to the system.

Attribute	Visibility	Data type	Description
username	private	string	username that is entered from input field
password	private	string	password that is entered from input field

Table 4.14: Description for the attributes of class *SigninBoundary*.

Operations	Visibility	Parameters	Return type	Description
set_username	public	username: string	void	set username value for the input field
set_password	public	password: string	void	set password value for the input field
submit	public		void	send sign-in request to controller

Table 4.15: Description for the operations of class *SigninBoundary*.

4.3.4.2 SignoutBoundary

Purpose of use: Allow user to perform sign out of the system.

Operations	Visibility	Parameters	Return type	Description
submit	public		void	send sign-out request to controller

Table 4.16: Description for the operations of class *SignoutBoundary*.

4.3.4.3 CreateAccountBoundary

Purpose of use: Allow user to create new account

Attribute	Visibility	Data type	Description
name	private	string	username of the new account
phone	private	string	phone number of the new account
email	private	string	email of the new account
username	private	string	username of the new account
password	private	string	password of the new account
role	private	string	role of the new account
address	private	string	address of the new account

Table 4.17: Description for the attributes of class *CreateAccountBoundary*.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
set_name	public	name: string	void	set name for the new account
set_phone	public	phone: string	void	set phone number for the new account
set_email	public	email: string	void	set email for the new account
set_username	public	username: string	void	set username for the new account
set_password	public	password: string	void	set password for the new account
set_role	public	role: string	void	set role for the new account
set_address	public	address: string	void	set address for the new account
submit	public		void	send creation request to controller

Table 4.18: Description for the operations of class *CreateAccountBoundary*.

4.3.4.4 ViewStaffListBoundary

Purpose of use: Allow user to view staff list.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
submit	public		void	send creation request to controller

Table 4.19: Description for the operations of class *ViewStaffListBoundary*.

4.3.4.5 ViewStaffProductivityBoundary

Purpose of use: Allow user to view productivity of a staff.

Attribute	Visibility	Data type	Description
name	private	string	username of the staff

Table 4.20: Description for the attributes of class *ViewStaffProductivityBoundary*.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
set_username	public	username: string	void	set username of the staff
submit	public		void	send the request to controller

Table 4.21: Description for the operations of class *ViewStaffProductivityBoundary*.

4.3.4.6 ViewStaffPersonalInfoBoundary

Purpose of use: Allow user to view personal information of a staff.

Attribute	Visibility	Data type	Description
name	private	string	username of the staff

Table 4.22: Description for the attributes of class *ViewStaffPersonalInfoBoundary*.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
set_username	public	username: string	void	set username of the staff
submit	public		void	send the request to controller

Table 4.23: Description for the operations of class *ViewStaffPersonalInfoBoundary*.

4.3.4.7 ViewStaffWorkshiftBoundary

Purpose of use: Allow user to view registered workshifts of a staff.

Attribute	Visibility	Data type	Description
name	private	string	username of the staff

Table 4.24: Description for the attributes of class *ViewStaffWorkshiftBoundary*.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
set_username	public	username: string	void	set username of the staff
submit	public		void	send the request to controller

Table 4.25: Description for the operations of class *ViewStaffWorkshiftBoundary*.

4.3.4.8 RegisterWorkshiftBoundary

Purpose of use: Allow user to view registered workshifts of a staff.

Attribute	Visibility	Data type	Description
day	private	string	a day in the week
dayshift_name	private	string	a shift in the day

Table 4.26: Description for the attributes of class *RegisterWorkshiftBoundary*.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
set_day	public	day: string	void	set day of the week
set_dayshift	public	dayshift_name: string	void	set shift of the day
submit	public		void	send the request to controller

Table 4.27: Description for the operations of class *RegisterWorkshiftBoundary*.

4.3.4.9 DeleteWorkshiftBoundary

Purpose of use: Allow user to view registered workshifts of a staff.

Attribute	Visibility	Data type	Description
day	private	string	a day in the week
dayshift_name	private	string	a shift in the day

Table 4.28: Description for the attributes of class *DeleteWorkshiftBoundary*.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
set_day	public	day: string	void	set day of the week
set_dayshift	public	dayshift_name: string	void	set shift of the day
submit	public		void	send the request to controller

Table 4.29: Description for the operations of class *DeleteWorkshiftBoundary*.

4.3.4.10 ViewMessagesBoundary

Purpose of use: Allow user to view messages sent to them.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
submit	public		void	send request to controller.

Table 4.30: Description for the operations of class *ViewMessagesBoundary*.

4.3.4.11 ViewCamerasBoundary

Purpose of use: Allow user to view real-time cameras.

Attribute	Visibility	Data type	Description
session_id	private	int	current user's session

Table 4.31: Description for the attributes of class *ViewCamerasBoundary*.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
submit	public		void	send request to controller.

Table 4.32: Description for the operations of class *ViewCamerasBoundary*.

4.3.4.12 ViewWeeklyScheduleBoundary

Purpose of use: Allow user to view the working schedule in a week.

Operations	Visibility	Parameters	Return type	Description
validate_current_user	private		bool	check if the current user is accessible to this function
submit	public		void	send request to controller.

Table 4.33: Description for the operations of class *ViewWeeklyScheduleBoundary*.

4.3.4.13 AuthController

Purpose of use: Handle requests related to authentication.

Operations	Visibility	Parameters	Return type	Description
signin	public	username: string password: string	void	log user into the system
signout	public		void	log user out of the system

Table 4.34: Description for the operations of class *AuthController*.

4.3.4.14 ManageStaffsController

Purpose of use: Handle requests related to staff management.

Operations	Visibility	Parameters	Return type	Description
staff_list	public		list<dict>	retrieve the staff list
create_account	public	name: string phone: string email: string username: string password: string role: string	void	create account
staff_productivity	public	username: string	dict	retrieve the productivity report of a staff
user_info	public	username: string	dict	retrieve user's personal information
weekly_schedule	public		dict	get the working schedule of the week
messages	public	username: string	list<dict>	get the messages sent to a user
view_cameras	public	session_id: int	list<Generator>	get the list of frame generator for camera streaming

Table 4.35: Description for the operations of class *ManageStaffsController*.

4.3.4.15 ManageWorkshiftController

Purpose of use: Handle request related to workshift management.

Operations	Visibility	Parameters	Return type	Description
staff_workshift	public	username: string	dict	retrieve the registered workshifts of a staff
register_workshift	public	username: string day: string dayshift_name: string	void	create a new workshift for user
delete_workshift	public	username: string day: string dayshift_name: string	void	delete a workshift of user

Table 4.36: Description for the operations of class *ManageWorkshiftController*.

4.3.4.16 User

Purpose of use: Model behavior and encapsulate data of user entity.

Attribute	Visibility	Data type	Description
id	private	int	unique ID of user
name	private	string	user's name
phone	private	string	user's phone number
email	private	string	user's email
username	private	string	user's username
password	private	string	user's password
role	private	string	user's role
address	private	string	user's address
productivities	private	list<Productivity>	user's productivity records
messages_received	private	list<Message>	user's received messages
workshifts	private	list<RegisteredWorkshift>	user's registed workshifts

Table 4.37: Description for the attributes of class *User*.

Operations	Visibility	Parameters	Return type	Description
User	public	username: string password: string role: string	User	user constructor
check_password	public	password: string	bool	check if the password is correct
get_name	public		string	get user's name
get_phone	public		string	get user's phone
get_email	public		string	get user's email
get_username	public		string	get user's username
get_role	public		string	get user's role
get_address	public		string	get user's addresss
get_messages	public		list	get user's received messages
get_productivity	public		list	get user's productivity records
get_workshifts	public		list	get user's workshifts
delete_workshift	public	day: string dayshift_name: string	void	delete a user's workshift
register_workshift	public	workshift: RegisteredWorkshift	void	add new workshift

Table 4.38: Description for the operations of class *User*.

4.3.4.17 Productivity

Purpose of use: Model behavior and encapsulate data of productivity entity.

Attribute	Visibility	Data type	Description
date	private	datetime.date	date of a working day
dayshift	private	DayShift	working shift of a day
arrival	private	datetime.time	arrival time of user
staying	private	datetime.timedelta	the interval a worker stays in working area

Table 4.39: Description for the attributes of class *Productivity*.

Operations	Visibility	Parameters	Return type	Description
Productivity	public	date: datetime.date dayshift: DayShift	Productivity	constructor
get_date	public		datetime.date	get date of the working day
get_dayshift	public		DayShift	get shift of the working day
get_arrival	public		datetime.time	get staff's arrival time
get_staying	public		datetime.timedelta	get the interval the staff staying in working area

Table 4.40: Description for the operations of class *Productivity*.

4.3.4.18 DayShift

Purpose of use: Model behavior and encapsulate data of day shift entity.

Attribute	Visibility	Data type	Description
name	private	string	name of the day shift
start_time	private	datetime.time	start time of the say shift
end_time	private	datetime.time	end time of the say shift

Table 4.41: Description for the attributes of class *DayShift*.

Operations	Visibility	Parameters	Return type	Description
DayShift	public	name: string	DayShift	constructor
get_name	public		string	get the name of the day shift
get_start_time	public		datetime.time	get the start time of the day shift
get_end_time	public		datetime.time	get the end time of the day shift

Table 4.42: Description for the operations of class *DayShift*.

4.3.4.19 Message

Purpose of use: Model behavior and encapsulate data of message entity.

Attribute	Visibility	Data type	Description
body	private	string	body of the message
timestamp	private	datetime.datetime	the time that the message was sent

Table 4.43: Description for the attributes of class *Message*.

Operations	Visibility	Parameters	Return type	Description
get_body	public		string	get the body of the message
get_timestamp	public		datetime.datetime	get the sending time of the message

Table 4.44: Description for the operations of class *Message*.

4.3.4.20 RegisteredWorkshift

Purpose of use: Model behavior and encapsulate data of a user's workshift.

Attribute	Visibility	Data type	Description
day	private	string	the week day
dayshift	private	DayShift	the shift of a day

Table 4.45: Description for the attributes of class *RegisteredWorkshift*.

Operations	Visibility	Parameters	Return type	Description
RegisteredWorkshift	public	day: string dayshift: DayShift	Registered-Workshift	constructor
get_day	public		string	get the day of the workshift
get_dayshift()	public		DayShift	get the shift of the day

Table 4.46: Description for the operations of class *RegisteredWorkshift*.

4.3.5 User interface design

4.3.5.1 Screen flow diagram

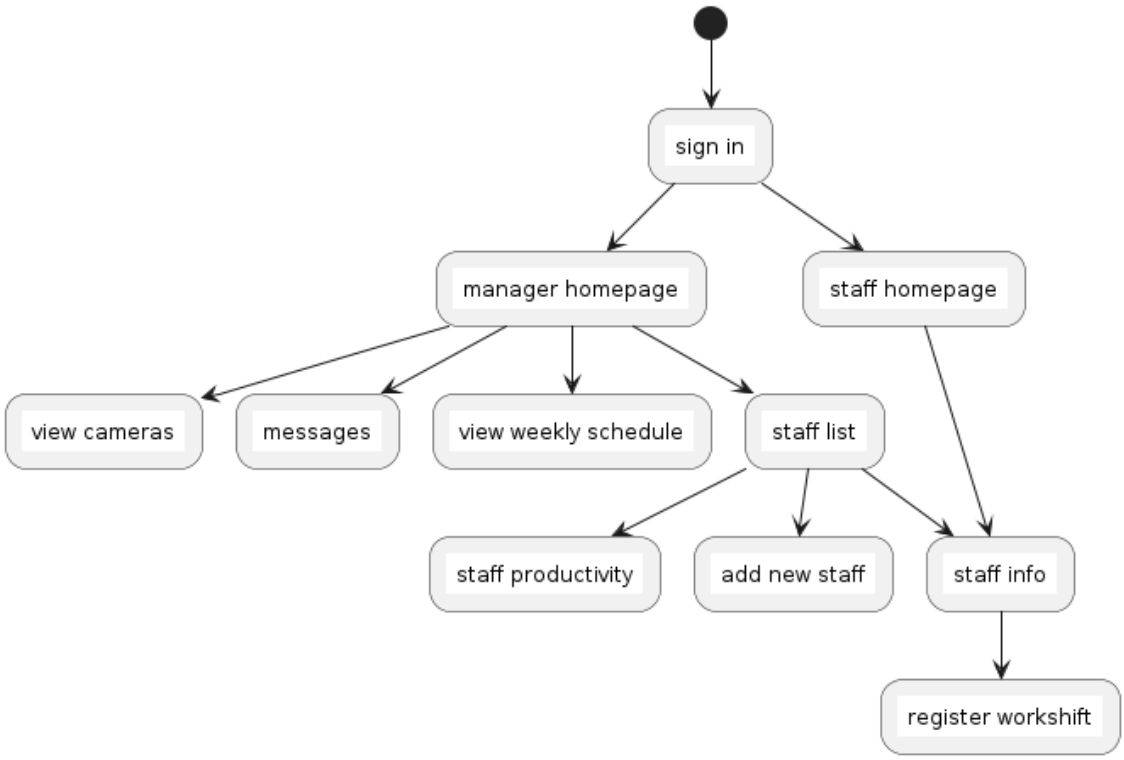


Figure 4.34: Screen flow diagram for the system's user interfaces navigation.

4.3.5.2 sign in

Figure 4.35: GUI design for the screen *sign in*.

4.3.5.3 manager homepage

Figure 4.36: GUI design for the screen *manager homepage*.

4.3.5.4 staff homepage

Figure 4.37: GUI design for the screen *staff homepage*.

4.3.5.5 staff list

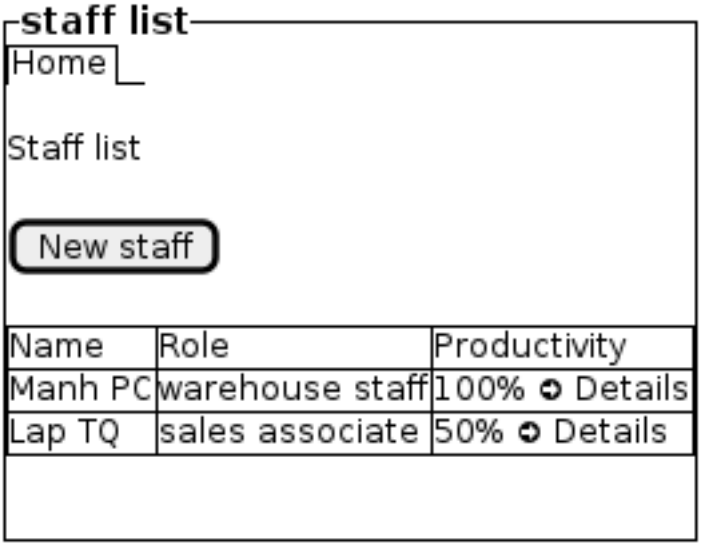


Figure 4.38: GUI design for the screen *staff list*.

4.3.5.6 view cameras

Figure 4.39: GUI design for the screen *view cameras*.

4.3.5.7 messages

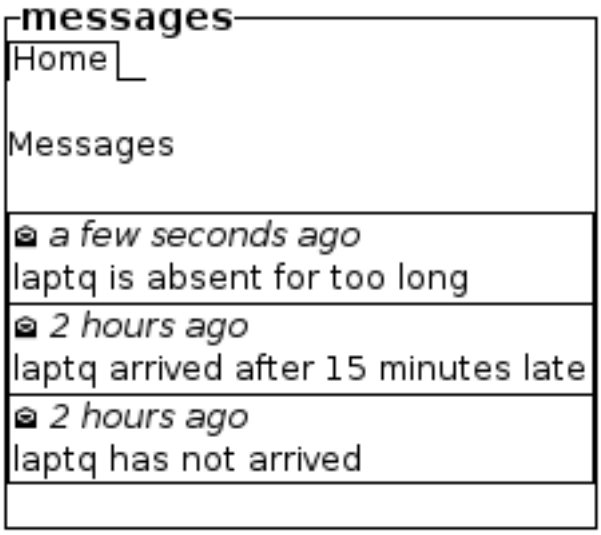


Figure 4.40: GUI design for the screen *messages*.

4.3.5.8 register workshift

Figure 4.41: GUI design for the screen *register workshift*.

4.3.5.9 staff productivity

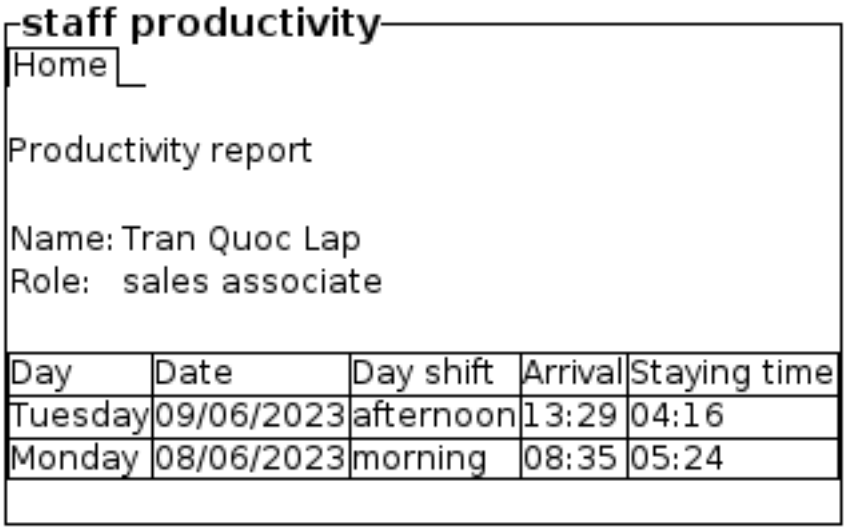


Figure 4.42: GUI design for the screen *staff productivity*.

4.3.5.10 add new staff

Figure 4.43: GUI design for the screen *add new staff*.

4.3.5.11 staff info

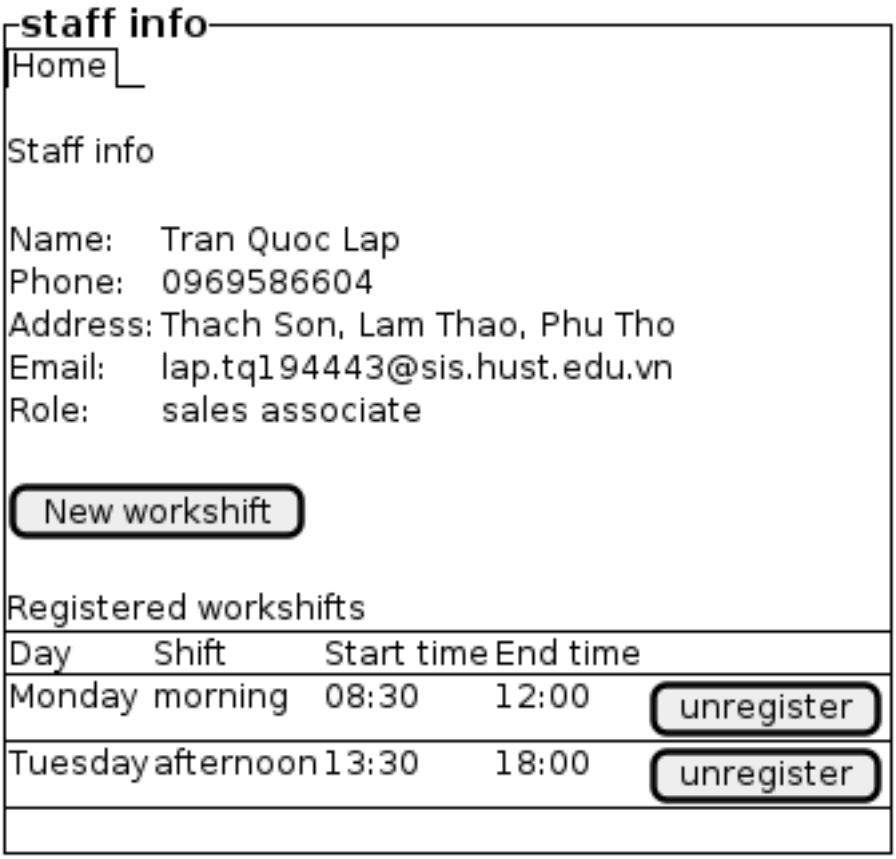


Figure 4.44: GUI design for the screen *staff info*.

4.3.6 Data design

4.3.6.1 Data tables relationship diagram

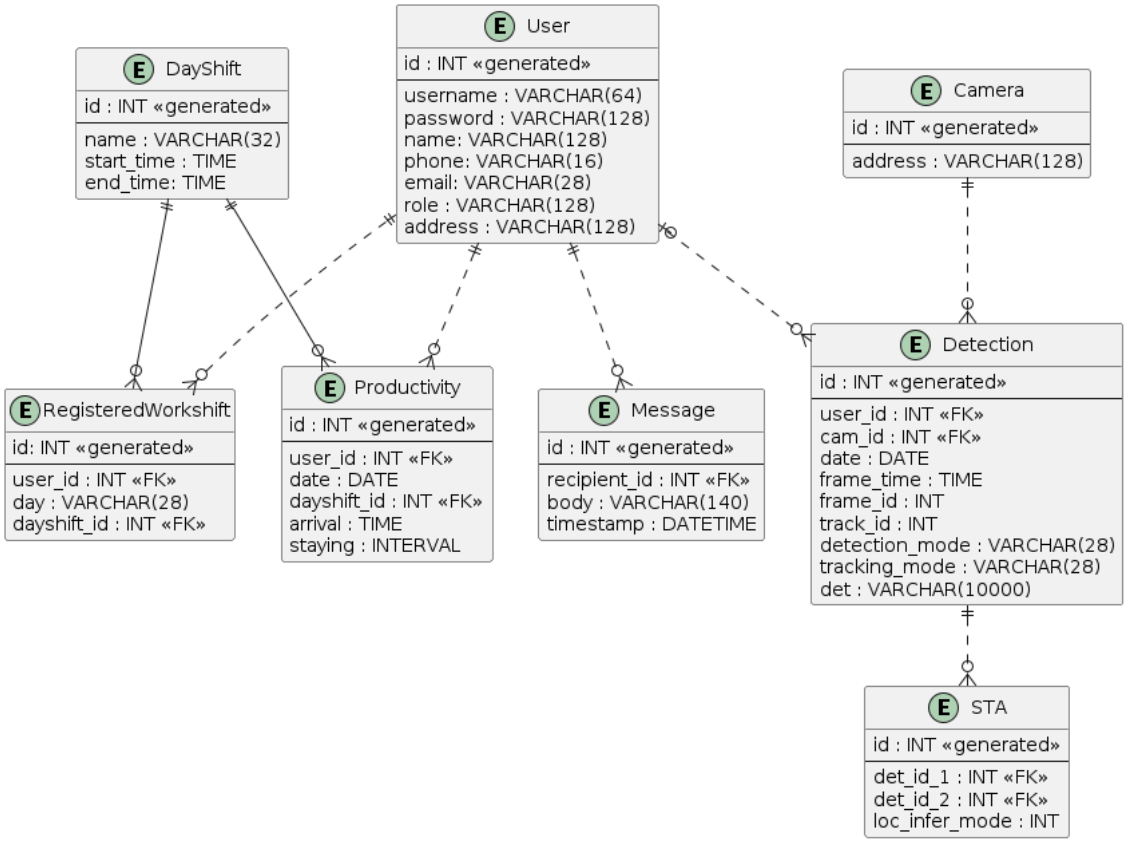


Figure 4.45: Data tables relationship diagram.

4.3.6.2 *User*

Purpose of use: Store user's information.

Field	Data type	Mandatory	Key	Description
id	INT	yes	primary	id of the user
username	VARCHAR(64)	yes		username of the user
password	VARCHAR(128)	yes		password of the user
name	VARCHAR(128)	yes		name of the user
phone	VARCHAR(16)	yes		phone number of the user
email	VARCHAR(28)	yes		email of the user
role	VARCHAR(128)	yes		role of the user

Table 4.47: Detailed design of the table *User*.

4.3.6.3 *DayShift*

Purpose of use: Store information of a day shift.

Field	Data type	Mandatory	Key	Description
id	INT	yes	primary	id of the day shift
name	VARCHAR(32)	yes		name of the day shift
start_time	TIME	yes		start time of the shift
end_time	TIME	yes		end time of the shift

Table 4.48: Detailed design of the table *DayShift*.

4.3.6.4 *RegisteredWorkshift*

Purpose of use: Store the working shift that a staff registered.

Field	Data type	Mandatory	Key	Description
id	INT	yes	primary	id of the workshift
user_id	INT	yes	foreign	id of the user
day	VARCHAR(28)	yes		week day of the workshift
dayshift_id	INT	yes	foreign	id of the day shift

Table 4.49: Detailed design of the table *RegisteredWorkshift*.

4.3.6.5 *Message*

Purpose of use: Store the message sent to a user.

Field	Data type	Mandatory	Key	Description
id	INT	yes	primary	id of the message
recipient_id	INT	yes	foreign	id of the recipient
body	VARCHAR(140)	yes		body of the message
timestamp	DATETIME	yes		the time the message was sent

Table 4.50: Detailed design of the table *Message*.

4.3.6.6 *Productivity*

Purpose of use: Store the productivity records of a user.

Field	Data type	Mandatory	Key	Description
id	INT	yes	primary	id of the productivity record
user_id	INT	yes	foreign	id of the user
date	DATE	yes		date of the working day
dayshift_id	INT	yes	foreign	id of the day shift
arrival	TIME	no		arrival time of user
staying	INTERVAL	no		staying interval of user in working area

Table 4.51: Detailed design of the table *Productivity*.

4.3.6.7 *Detection*

Purpose of use: Store the person detection result from the camera system.

Field	Data type	Mandatory	Key	Description
id	INT	yes	primary	id of the detection record
user_id	INT	no	foreign	id of the user assigned to the detection
cam_id	INT	yes	foreign	id of the camera
date	DATE	yes		date that the person is detected
frame_time	TIME	yes		time that the person is detected
frame_id	INT	yes		id of the frame
track_id	INT	yes		if of the track
detection_mode	VARCHAR(28)	yes		detection mode used
tracking_mode	VARCHAR(28)	yes		tracking mode used
det	VARCHAR(10000)	yes		coordinate of the detection

Table 4.52: Detailed design of the table *Detection*.

4.3.6.8 *STA*

Purpose of use: Store the matching of people among cameras.

Field	Data type	Mandatory	Key	Description
id	INT	yes	primary	id of the match
det_id_1	INT	yes	foreign	id of the first detection
det_id_2	INT	yes	foreign	id of the second detection
loc_infer_mode	INT	yes		method used to infer location from detection

Table 4.53: Detailed design of the table *STA*.

4.3.6.9 *Camera*

Purpose of use: Store camera's information.

Field	Data type	Mandatory	Key	Description
id	INT	yes	primary	id of the camera
address	VARCHAR(128)	yes		address of the camera

Table 4.54: Detailed design of the table *Camera*.

4.4 Implementation and Deployment

4.4.1 Software technologies used for the application system development

The system was deployed on a server with Intel Xeon E5-2620 v4 [6] and two NVIDIA GeForce GTX 1080 Ti [18]. It was implemented using:

- Web app framework: Flask [19]
- Database: SQLite [14]

Flask [19] is a micro-framework web framework built using the Python programming language. It supports creating a wide range of web applications, from simple APIs to complex web applications such as websites, blogs, wikis, and even e-commerce sites. To build high-quality applications, Flask [19] is equipped with a variety of tools, libraries, and technologies:

- Flask-WTF [11] extension that handles the web forms in this employee management system.

- Flask-SQLAlchemy [10] extension, which is an Object Relational Mapper that enables the management of a database through the use of high-level entities like classes and methods, rather than tables and SQL queries.
- Flask-Migrate [9] extension, which is a database migration that modifies an existing database to accommodate changes in the application.
- Flask-Login [8] extension, which manages the user logged-in state.

4.4.2 Software technologies used for Deep learning and Computer vision

To perform experiments and operate the key functions of the application system, some AI and image processing frameworks were used, including:

- Machine Learning framework: PyTorch [20]
- Image processing: OpenCV [4]
- Optimization: SciPy [26]

PyTorch [20] is the most popular framework for training deep learning models in computer vision. Building deep learning models requires a large amount of training data, but research works using PyTorch [20] also include pre-trained models with high accuracy to facilitate the tasks. In this thesis, the person detection model used is YOLOv7 [27], which has been pre-trained by the authors.

OpenCV [4] is an open-source library that is widely used for image and video processing. It includes algorithms for feature detection, object recognition, etc. This thesis utilized OpenCV [4] for fundamental image processing tasks like reading video streams from disk or cameras, calculating homography, drawing visualization, and displaying the final output.

SciPy [26] is an open-source Python library for scientific computing, which contains efficient implementations of common algorithms like optimization and linear algebra. In this thesis, SciPy [26] implementation of the Hungarian algorithm was used to solve the assignment problem in matching object detections between different cameras. By finding the optimal one-to-one assignment, SciPy [26] allowed the linking of tracks with the lowest cost based on the spatio-temporal distance metric.

4.4.3 Screenshots of the implemented application system

The system was mainly designed to support managers, so signing in with a manager account was needed to review the functionalities of the implemented system.

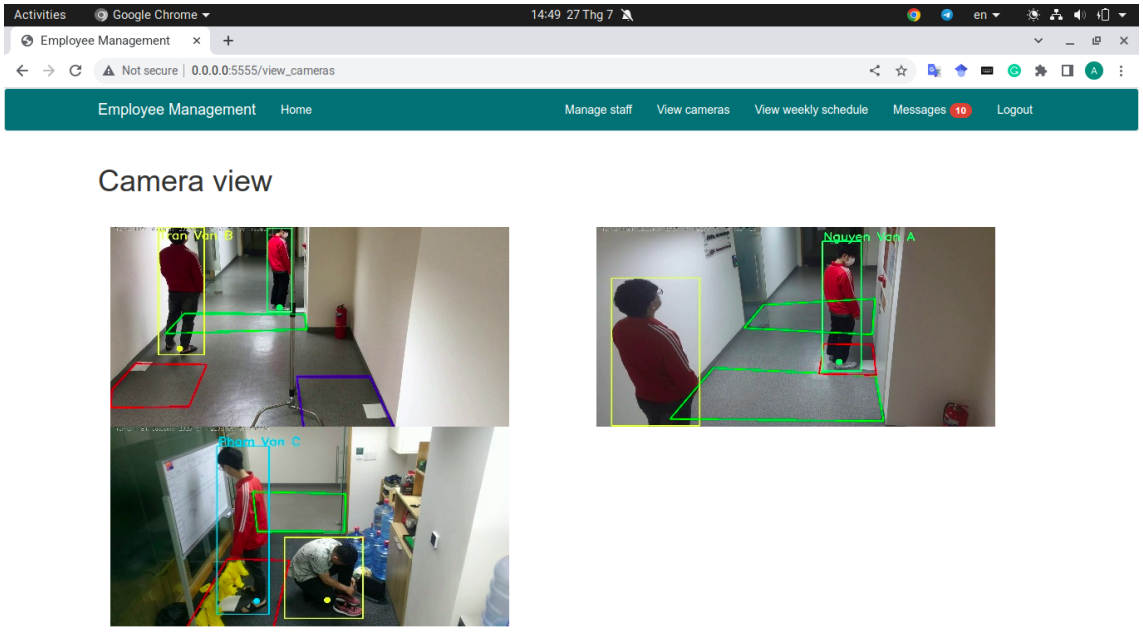


Figure 4.46: View real-time camera screen. The *blue* region represents the check-in area. The *green* region represents the overlapping area. The *red* region represents the work area.

Figure 4.46 through 4.49 are the screenshots of the most important features of the system.

Particularly, Figure 4.46 is the screenshot of the *View camera* screen, which corresponds to the usecase *UC03: view real-time cameras*. The figure describes a moment when the system visualizes the employee tracking result.

Figure 4.47 and Figure 4.48 capture the employee productivity reported by the system, which fulfills the usecase *UC12: view staff productivity*. Especially, Figure 4.47 summarizes the number of days each worker went to work late and the percentage of their staying time in the designated work area over the entire employment. Figure 4.48 is the detailed productivity report for each day of an individual.

It's noticeable that there is a big red badge in the top navigation bar, right next to the *Message* tab. This badge emerges whenever an irregular behavior of employees is detected, for example when they do not arrive in time after a certain interval, or when they are absent from their work area for too long. This feature corresponds to the usecase *UC05: get notified about staff's irregular behaviors*. By clicking the *Message* tab, managers can get details of the message, as shown in Figure 4.49.

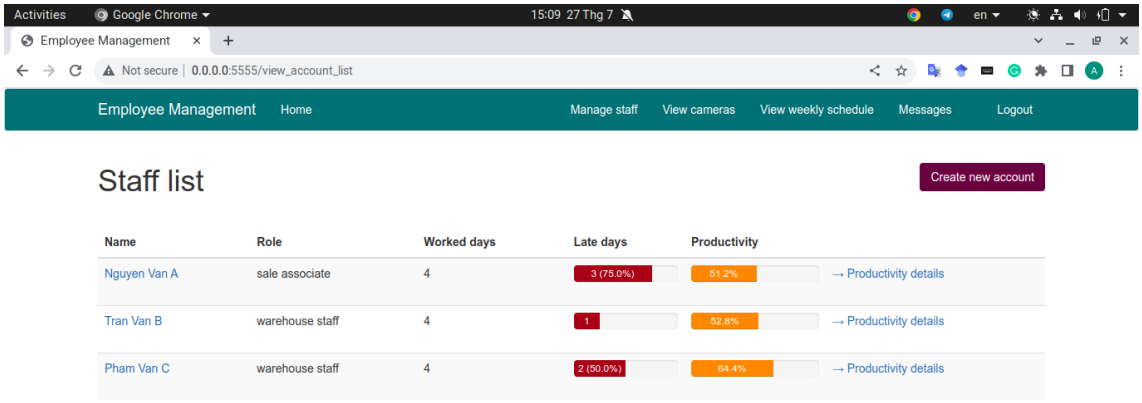


Figure 4.47: View general staff productivity screen. Notice that the badge in the tab *Message* of the navigation shows the number of unseen messages sent to the manager.

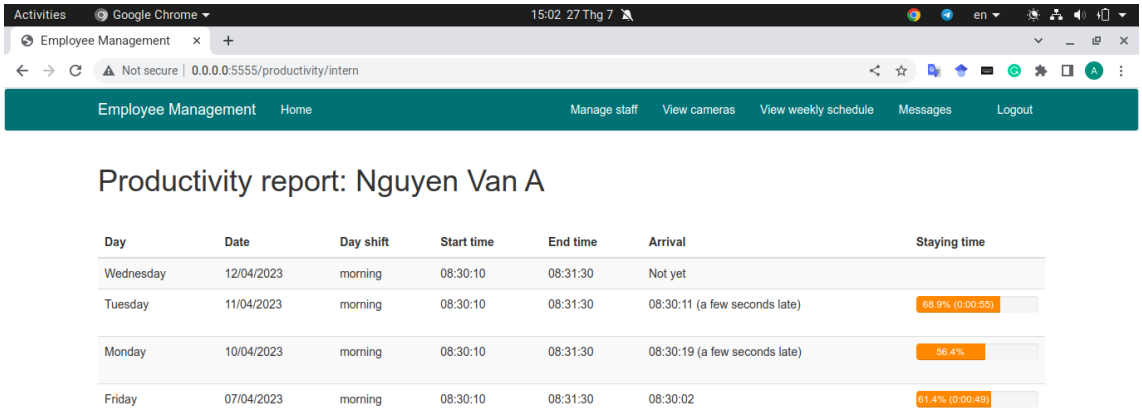


Figure 4.48: View detailed productivity of each staff screen.

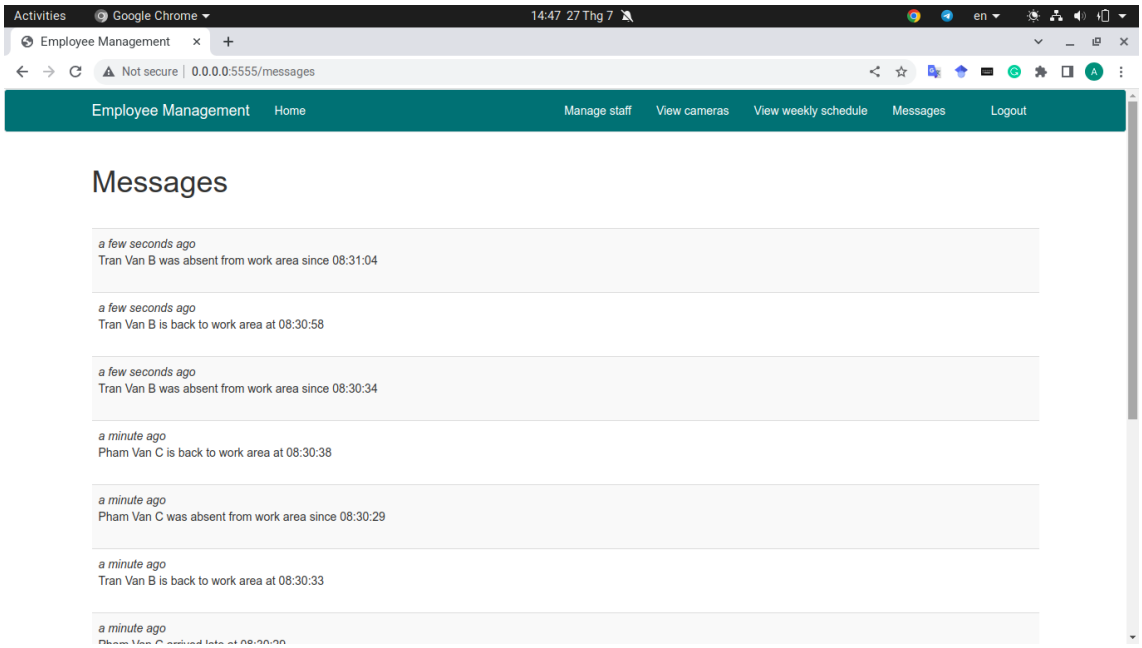


Figure 4.49: View alerted messages screen.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The thesis work has completed all three defined objectives: research, technology, and application. In terms of research, the proposed method using spatio-temporal association is a good solution to MCT. It can work well in case of current visual appearance-based methods performing worse such as people wearing uniforms or having similar appearance. In addition, its strength and weakness were also deeply investigated for further improvement. In terms of technology, frameworks like PyTorch [20], OpenCV [4] and Scipy [26] were effectively used to perform experiments. In terms of application, a software system was developed to prove the applicability of the research work in practice, by tracking employee behavior and calculating their productivity in the workplace. Implementing this system has provided me with invaluable insight to challenges encountered when implementing AI in real software systems.

However, the current MCT system still has several limitations that need to be addressed. In terms of research, the mapping method between cameras relies on the results from previous steps, which is single camera tracking. Two main issues are missing detection and ID switching, which have a direct and negative effect on the results and the process of building a solution for MCT. Observations on the particular videos show that missing detection occurs frequently when multiple people are close together optically. Additionally, Table 3.1 demonstrates that ID switching is quite common, especially in challenging videos. Therefore, a more robust person tracking algorithm needs to be developed that can both interpolate bounding boxes when individuals are temporarily occluded and limit track swaps. Solving this issue would significantly improve the accuracy of many MCT research

projects and increase their practical applicability. Moreover, the proposed method only uses spatial and temporal information. although the purpose of the thesis is to build a STA solution to replace visual-based Re-ID methods. Combining these two methods in general problems might produce even more impressive results.

Regarding the application system, the current system design is basic and only provides the most essential functions for demonstration purposes. Therefore, in the future, additional features will need to be developed to create a more feature-rich tracking system. Furthermore, the system is not yet optimized in terms of speed and memory management. For example, the current system uses a Pose estimation model that returns 17 keypoints to interpolate a single foot point, which is inefficient and wasteful. A specialized model that detects only foot points may be more suitable. Therefore, the plan is to optimize and scale the system in the future to handle, for example, a larger number of cameras more efficiently. The system will be upgraded it with more advanced technologies to further suited for professional use.

REFERENCE

[1] Nadeem Anjum and Andrea Cavallaro. “Trajectory Association and Fusion across Partially Overlapping Cameras”. In: *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance* (2009), pp. 201–206.

[2] Keni Bernardin and Rainer Stiefelhagen. “Evaluating multiple object tracking performance: The CLEAR MOT metrics”. In: *EURASIP Journal on Image and Video Processing* 2008 (Jan. 2008). DOI: 10.1155/2008/246309.

[3] Alex Bewley et al. “Simple online and realtime tracking”. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003.

[4] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000).

[5] Andrew Tzer-Yeu Chen, Morteza Biglari-Abhari, and Kevin I-Kai Wang. “Fusing Appearance and Spatio-Temporal Models for Person Re-Identification and Tracking”. In: *Journal of Imaging* 6 (2020).

[6] Intel Corporation. *Intel® Xeon® Processor E5-2620 v4 Product Specifications*. https://www.intel.com/content/www/us/en/products/sku/92986/intel-xeon-processor-e52620-v4-20m-cache-2-10-ghz/specifications.html. Accessed: July 13th, 2023. 2016.

[7] Yunhao Du et al. “GIAOTracker: A Comprehensive Framework for MCMOT With Global Information and Optimizing Strategies in VisDrone 2021”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2021, pp. 2809–2819.

[8] Flask-Login Contributors. *Flask-Login Documentation*. GitHub. 2021. URL: https://flask-login.readthedocs.io/.

[9] Flask-Migrate Contributors. *Flask-Migrate Documentation*. GitHub. 2021. URL: https://flask-migrate.readthedocs.io/.

[10] Flask-SQLAlchemy Contributors. *Flask-SQLAlchemy Documentation*. GitHub. 2021. URL: https://flask-sqlalchemy.palletsprojects.com/en/3.0.x/.

[11] Flask-WTF Contributors. *Flask-WTF Documentation*. GitHub. 2021. URL: https://flask-wtf.readthedocs.io/.

[12] Ross Girshick. "Fast R-CNN". In: *International Conference on Computer Vision (ICCV)*. 2015.

[13] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Computer Vision and Pattern Recognition*. 2014.

[14] D. Richard Hipp. "SQLite". In: *ACM Transactions on Database Systems (TODS)* 32.4 (2007), pp. 1–39.

[15] Jungik Jang, Min-Ju Seon, and Jaehyuk Choi. "Lightweight Indoor Multi-Object Tracking in Overlapping FOV Multi-Camera Environments". In: *Sensors (Basel, Switzerland)* 22 (2022).

[16] Jonathon Luiten et al. "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking". In: *International Journal of Computer Vision* 129.2 (Oct. 2020), pp. 548–578. ISSN: 1573-1405. DOI: 10.1007/s11263-020-01375-2. URL: http://dx.doi.org/10.1007/s11263-020-01375-2.

[17] Debapriya Maji et al. "YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2022, pp. 2637–2646.

[18] NVIDIA Corporation. *GeForce GTX 1080 Ti Graphics Card*. https://www.nvidia.com/en-gb/geforce/graphics-cards/geforce-gtx-1080-ti/specifications/. Accessed: July 13th, 2023. 2017.

[19] Pallets Projects. *Flask*. Version 2.0.1. 2021. URL: https://github.com/pallets/flask.

[20] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[21] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[22] Joseph Redmon and Ali Farhadi. *YOLO9000: Better, Faster, Stronger*. 2016. arXiv: 1612.08242 [cs.CV].

[23] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. 2018. arXiv: 1804.02767 [cs.CV].

[24] Joseph Redmon et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2015. arXiv: 1506.02640 [cs.CV].

[25] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2015.

[26] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2. URL: https://doi.org/10.1038/s41592-019-0686-2.

[27] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. In: *arXiv preprint arXiv:2207.02696* (2022).

[28] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. “Simple Online and Realtime Tracking with a Deep Association Metric”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962.

[29] Xindi Zhang and Ebroul Izquierdo. “Real-Time Multi-Target Multi-Camera Tracking with Spatial-Temporal Information”. In: *2019 IEEE Visual Communications and Image Processing (VCIP)*. 2019, pp. 1–4. DOI: 10.1109/VCIP47243.2019.8965845.

[30] Yifu Zhang et al. “ByteTrack: Multi-Object Tracking by Associating Every Detection Box”. In: (2022).

[31] Kaiyang Zhou et al. *Omni-Scale Feature Learning for Person Re-Identification*. 2019. arXiv: 1905.00953 [cs.CV].