People Tracking in a Smart Campus context using Multiple Cameras

Henrique Matos[1], Henrique Santos[1]

[1]ALGORITMI RD Centre, University of Minho, Guimarães, Portugal

Abstract

Object multi-tracking has been a relevant topic for different applications, such as surveillance, mobility, and ambient intelligence. It is particularly challenging when considering open spaces, like Smart Cities, which demand multi-camera solutions with issues like re-identification. In this paper, we describe a framework aiming to provide multi-tracking of people throughout a university campus as part of a larger project (Lab4USpaces) to develop a Smart Campus initiative. Several object detection models and real-time tracking open-source algorithms were compared. The project contemplates a set of low-cost video cameras covering most of the campus, with or without overlapping. After researching different alternatives, the proposed framework uses the YOLOv7 tiny model for object detection, BoT-Sort for multiple object tracking, and Deep Person Reid for re-identification. We also faced challenges concerning the privacy and security of campus users. The multi-tracking system complies with current regulations since no personal identification is ever performed, and no images are stored for longer than necessary for object detection and re-identification. Besides describing the first prototype, this paper discusses some validation tests and describes some potential uses.

Keywords

Smart Campus, Object Detection, Multiple Object Tracking, Re-Identification, People Tracking

1. Introduction

In the context of an undergoing research project named Lab4U&Spaces[1] – aiming at exploring innovative technologies to raise the quality of life at the university campus – the work described here is focused on the campus' users management and mobility dimension. Using this platform, students can, for example, avoid a place with a more extensive flow of users when scheduling a joint activity. On the other hand, campus managers can quickly locate areas of more significant influx, properly understand this dynamic, and prepare more appropriate responses to avoid it, if recommended. The need to prevent excessive exposure to UV rays due to carelessness by users or even reduce contact to prevent viral dissemination (as happened in the recent pandemic situation caused by COVID-19) are other examples of important campus management objectives that would benefit from this platform. Using video-based techniques for this purpose, indoor

RCIS 2023: The 17th International Conference on Research Challenges in Information Science, May 23–26, 2023, Corfu, Greece

matoshenrique1999@gmail.com (H. Matos); hsantos@dsi.uminho.pt (H. Santos)

https://www.linkedin.com/in/henriquematos99/ (H. Matos); https://hsantos.dsi.uminho.pt (H. Santos)

0000-0002-3850-2159 (H. Matos); 0000-0001-7116-9338 (H. Santos)

[1]https://transparencia.gov.pt/pt/fundos-europeus/beneficiarios-projetos/projeto/NORTE-01-0145-FEDER-000072

and outdoor, is not usually identified as a possible solution for economic reasons [1]. Even so, the rise of processing power and widespread availability of low-cost video-capable devices makes it possible. Security and privacy is the main concern in this type of environment, and regulatory documents like the GDPR, in particular, its items demanding privacy-by-design and by default must be attended to, imposing specific requirements that limit the solution space concerning detection and identification. The paper is organised as follows: Section 2 compares related projects and methods, and Section 3 presents the proposed solution and a comparison of object detection, tracking, and re-identification techniques. Section 4 describes the testing and validation methods, and the final section presents conclusions and possible project evolution.

2. Related projects

Most of the techniques used in this paper are more frequently detailed in video surveillance or Computer Vision applications. Concerning university campuses and the project's context, those techniques should be used and framed by specific requirements. When researching related projects, we searched within both domains but emphasised application rather than the algorithms development themselves. In [2], the authors present an IoT-based system designed to track vehicles and pedestrians on a Smart Campus. The system combines various sensors, including GPS, RFID, and LiDAR, with cameras to collect tracking data. This data is then processed to create real-time location information, which is communicated and stored in a central database. The system has potential applications in traffic management, safety monitoring, and environmental monitoring, and the authors argue that it is both reliable and cost-effective. The work described in [3] presents a real-time tracking algorithm that can track multiple targets using multiple cameras. The algorithm employs a Kalman filter and a spatial-temporal model. The authors demonstrate the applicability in several surveillance applications, including security, transportation, and sports.

In [4] the authors proposed a smart city and traffic analysis system similar to the one planned for our project. They used Cascade R-CNN with ResNet-101 for vehicle detection, TPM for multiple object tracking in a single camera, and HRNet and Res2Net for vehicle re-identification. The system was effective but has performance limitations, indicating the need for improvements. Moreover, those limitations impact negatively people tracking. Another similar solution is proposed in [5]. However, there is one significant distinction, since it was developed for a wide range of applications. The authors introduce two techniques: DeepCC for Multi-Target Multi-Camera Tracker (MTMCT); and Adaptive Weighted Triplet Loss (AWTL) for re-identification. The results are auspicious, but since the publication of this work, some new technologies have emerged, allowing for optimised techniques within this research context. They will be referred to in the next section along with the description of the proposed solution.

3. Proposed solution

The general system architecture for the Lab4USpaces platform is divided into four layers, as shown in Figure 1. The physical layer includes the tracking component, which is located where all sensors and actuators are placed. IP cameras capture video and send it to an edge server for

processing, including configuration management and object geolocation. The network layer enables wireless communication between the sensor subsystems and the middleware. The integration layer includes an Identity and Access Manager module for device authentication, a Message Broker for communication organisation, a Temporal Database for data storage, and the Home Assistant platform as the Hub. The application layer uses the collected data for analysis, visualisation, and decision-support applications. The data tracking subsystem will be explained in detail later.

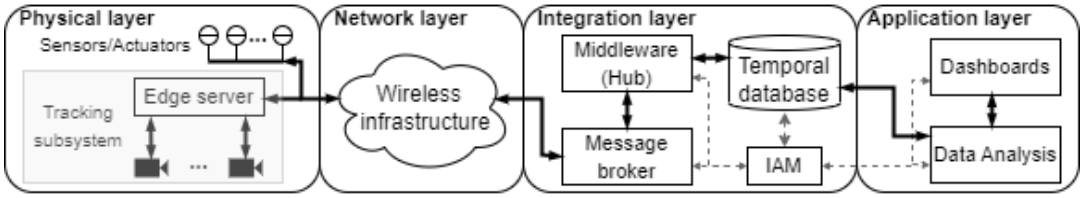


Figure 1: General system architecture.

Figure 2 depicts the tracking subsystem flow graph. It consists of several identical parallel branches, each one running in a dedicated camera subsystem. The generic operation model for this program involves a global initialisation process, followed by a loop that includes acquiring the next video frame and performing object detection, which predicts object boxes and performs non-maximum suppression (NMS) and intersection over union (IOU) to remove duplicate detection boxes of the same object. The loop also involves performing object tracking, which results in an ID for each detected object. In two specific cases, alternative paths are triggered: if any object enters the scene, a query is executed to the re-identification module to check for a previously attributed ID, and if any object leaves the scene, its ID and necessary shape information are sent to the re-identification module. Finally, tracking information is sent to the database.

3.1. Object Detection

The object detection module must be able to identify correctly all people in crowded scenarios using low-cost video cameras. Such scenarios pose challenges like occlusion and clustering, hindering precision and recognition [6]. Open-source YOLO-based techniques were compared for this purpose using a machine with an Intel Core i7-8550U @1.80GHz CPU and 8GB RAM. All models were trained using the COCO dataset with 91 object types and 2.5 million labelled instances in 328k images. Table 1 shows the mean average precision (mAP) and average process time with and without GPU – the results obtained with YOLOv3 and some YOLOVv5 variants were suppressed since they are not influential. YOLOv7 was recently introduced and reportedly outperforms other detectors in both speed and accuracy [7], which aligns with our results. However, for higher precision with low processing time, YOLOR is also an alternative. When using a GPU YOLOR or YOLOv7 would be good choices. Overall, YOLOv7 is the best choice.

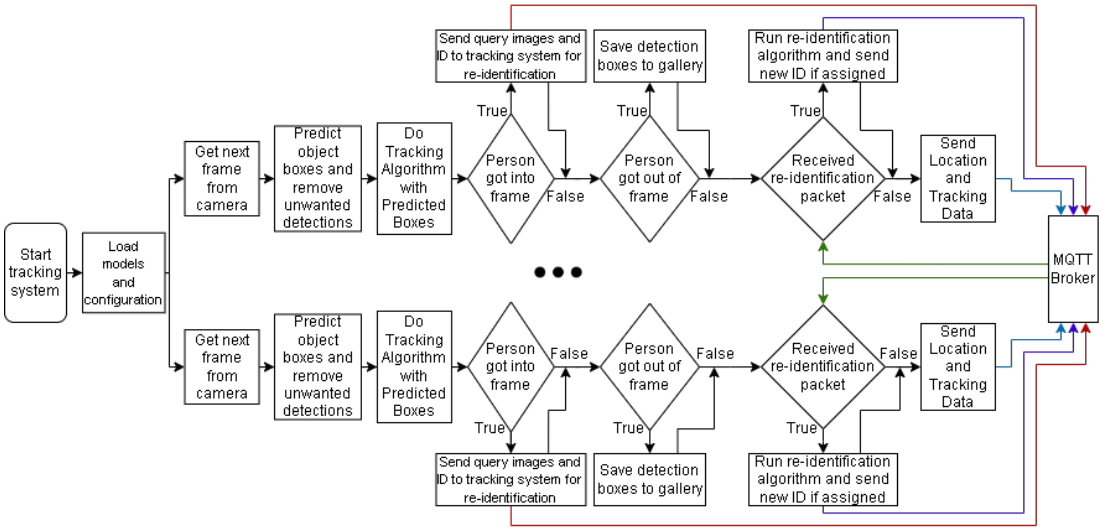


Figure 2: Basic system flowchart

Table 1
Models Comparison (APT – Average Processing Time)

Model Name	COCO mAP (%)	GPU APT (ms)	CPU APT (ms)
YOLOv5n 640×640	28.0	6.3	**91**
YOLOR-CSP 640×640	52.8	9.4	577
YOLOR-CSP-X 640×640	**54.8**	11.5	967
YOLOv7-tiny 640×640	38.7	**3.5**	97.1
YOLOv7 640×640	51.4	6.2	463
YOLOv7X 640×640	53.1	8.7	1227

3.2. Single Camera Multiple Object Tracking

There are two types of trackers: offline (previous and subsequent frames are available to create more accurate predictions), and online (work on the fly). In this project, we need a real-time online tracking system. Multiple problems can occur, like occlusions, initialisation and termination of tracks, people with similar appearances, and interaction between multiple objects. Occlusion can cause identity switches and fragmentation of trajectories, which should be avoided in our project. A common way to benchmark object tracking algorithms is to use the MOTChallenge, a standardised evaluation framework for multiple object tracking (MOT). It contains two datasets with indoor and outdoor videos.

We evaluated four MOT algorithms: SORT [8], DeepSORT [9], StrongSORT [10], and BOT-SORT [11]. Table 2 shows the results obtained for the two datasets, using the metrics relevant to our case: HOTA (Higher Order Tracking Accuracy), IDF1 (ratio of correctly identified detections over the average number of ground-truth and computed detections, MOTA (Multiple Object Tracking Accuracy), and processing time per frame. SORT has the lowest processing time but its accuracy is too low. Both BOT-SORT and BOT-SORT-ReID have better accuracy, but the ReID version has a higher processing time, making BOT-SORT the best option.

Table 2
MOT17 and MOT20 Algorithms Comparison [11]

MOT Challenge	Algorithm	HOTA	IDF1	MOTA	Frame Process Time (ms)
MOT17	SORT	34	39.8	43.1	**7**
	DeepSORT	61.2	74.5	78	72.5
	StrongSORT++	64.4	79.5	79.6	140.8
	BoT-SORT	64.6	79.5	**80.6**	151.5
	BoT-SORT-ReID	**65.0**	**80.2**	80.5	222.2
MOT20	SORT	36.1	45.1	42.7	**17.5**
	DeepSORT	57.1	69.6	71.8	312.5
	StrongSORT++	62.6	77	73.8	714.3
	BoT-SORT	62.6	76.3	77.7	151.5
	BoT-SORT-ReID	**63.3**	**77.5**	**77.8**	416.7

3.3. Re-Identification

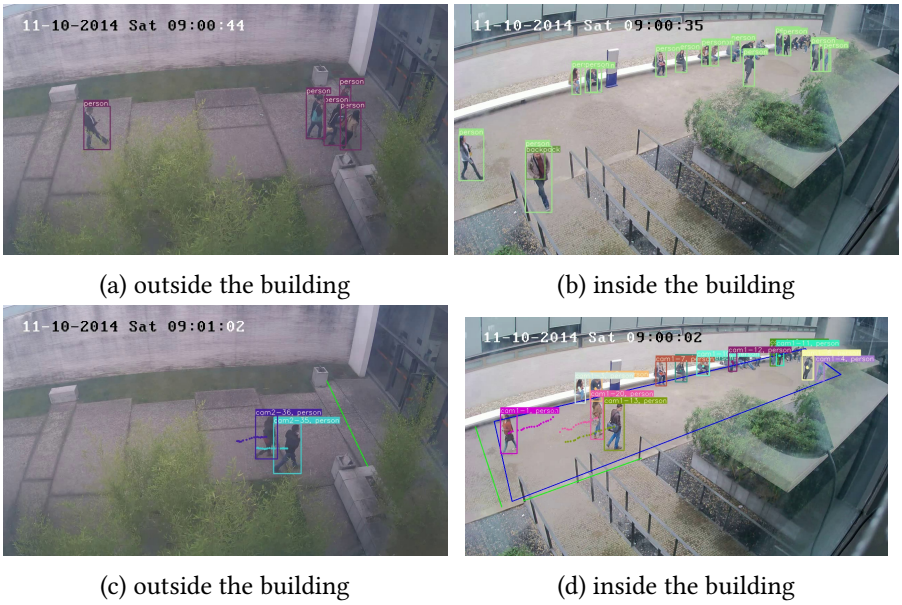
This operation involves using query images of the person to be labelled and gallery images (e dedicated common storage) that contain previously detected IDs from neighbour cameras with a common path. There are well-known re-identification algorithms such as Centroids-Reid [12] and LUPerson [13]. However, they were developed for specific datasets with highly predictive flows and shapes that do not match our project's needs. The Deep Person Reid [14] algorithm is similar and was chosen since it was trained and used in cross-domain with datasets similar to what we expect to have. Re-identification is performed using line intersection zones that delineate boundaries between camera views. When a subject crosses these lines, the re-identification operation is triggered, either for querying a neighbour's camera or storing a group of images with the corresponding ID. The querying operation returns a similarity value between the gallery images and the input one, along with associated IDs. If the value obtained is acceptable, the ID is assumed.

4. Results and Configurations

This section presents the results of object detection, tracking, and re-identification when applied to images captured in the Lab4U&Spaces project, using two video cameras, one inside and the other outside a building, in connected spaces without scene overlapping. We also describe the configuration settings of the global system and demonstrate the final results on campus.

Concerning object detection, Figures 3a and 3b display the outcomes obtained from both cameras. The inside camera covers a wider area with extreme resolution variation due to near and far objects, with no constraints imposed on the minimum object size. It is noticeable that some people in the most distant zone, on the right side of the figure, are not detected. Even though, the selected technique performs better than all other solutions in terms of response time and computing resources required. The rate of false negatives in figure 3b is approximately 64% – despite not being optimal, it is acceptable.

Concerning the Single Camera Multiple Object Tracking adopted technique, Figures 3c and 3d illustrate the result of the BoT-SORT algorithm in sequential scenes captured from both cameras. The average time required for detecting and tracking people was between 142.8ms (7



(a) outside the building (b) inside the building

(c) outside the building (d) inside the building

Figure 3: People detection and multiple object tracking

FPS) and 263ms (3.8 FPS). These values are consistent with the recommended frame rate for this application class, as suggested in [15] – the authors propose a working point of 6 FPS with 0% accuracy loss, and it is acceptable to reduce the FPS by 80% (1.2 FPS) while maintaining precision rates above 60%.

Concerning the re-identification operation, figure 4a shows a person crossing the delimiter zone line in the outside scene, using the label "cam2_1" (meaning camera 2 and the person ID 1); Figure 4b shows the same person entering the inside camera's view area a few seconds later and being labelled "cam1_22", meaning it was not yet re-identified; finally, in figure 4c, it is visible that the ID was redefined, to the one previously assigned by camera 2 (about two seconds after being initially detected). In the small dataset used, out of four possible re-identifications, three of them were correctly performed suggesting an efficacy of 75% – but particularly in this case, more experiments are required to validate this result.

(a) subject ID at outside (b) subject ID at inside (c) subject ID re-assigned

Figure 4: Re-identification in action

After completing the main loop shown in Figure 2, the detection and tracking data is stored

in the Hub. The collected data can be utilised in several applications, such as the one depicted in Figure 5, which shows in real-time or using recorded data the density of people in different campus spaces through colour and bubble size codes – this example uses only one camera, for illustration purposes. The left image displays the complete campus map, whereas the right image focuses on a particular corridor where the indoor camera was installed.

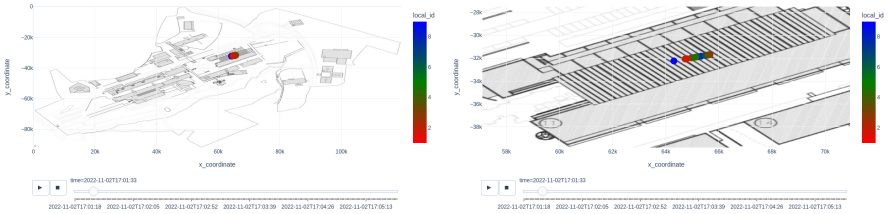


Figure 5: Campus tracking results.

Configuration

The tracking module needs additional configuration details to function properly throughout the entire campus, allowing to characterise the space and optimise computational and storage resources. A web application was created to manage configuration data. The main items include *Zone*, defined by multiple polygons in each camera's field of view, allowing for the definition of zones of interest where specific views or details should be highlighted; *Line Intersection Zone*, which delimits the boundary between zones in a camera's scene where particular operations like re-identification or people counting should be applied; *Black Area*, used to remove unwanted areas from a camera's view where no person can be found, or when two cameras overlap to prevent resource wastage; and *Global Coordinates*, necessary to track and re-identify individuals throughout the campus (it involves mapping each camera's field of view to the global campus map and define a scale, angle, and offset to transform the tracking data into campus' coordinates.

5. Conclusion

This paper describes a framework for tracking people on a university campus as part of the Lab4U&Spaces project, which aims to develop a platform for exploring smart campus technologies. We evaluated various technologies and selected the ones that best suits the project requirements, which included low computational resources, energy constraints, and open-source solutions. Privacy is another fundamental requirement we guarantee by not storing any image for consultation or beyond the time strictly required in the re-identification function. We conducted experiments at the prototype level to validate all operations and found that the framework is viable. We also discussed the potential use of this technology at the campus level. However, to determine the framework's actual usefulness, it needs testing with more than two cameras and evaluation of the behaviour of thousands of daily campus visitors. As future work, we start designing applications that exploit all available data to transform campus management and life into more intelligent activities.

Acknowledgments

Lab4U&Spaces – Living Lab of Interactive Urban Space Solution, Ref. NORTE-01-0145-FEDER-000072, financed by community funds (FEDER), through Norte 2020.

References

[1] M. Musa, M. N. Ismail, M. F. M. Fudzee, A survey on smart campus implementation in malaysia, JOIV : International Journal on Informatics Visualization 5 (2021) 51–56.

[2] J. Toutouh, E. Alba, A low cost iot cyber-physical system for vehicle and pedestrian tracking in a smart campus, Sensors 22 (2022).

[3] X. Zhang, E. 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.

[4] J. Ye, X. Yang, S. Kang, Y. He, W. Zhang, L. Huang, M. Jiang, W. Zhang, Y. Shi, M. Xia, X. Tan, A robust mtmc tracking system for ai-city challenge 2021, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 4039–4048.

[5] E. Ristani, C. Tomasi, Features for multi-target multi-camera tracking and re-identification, 2018.

[6] D. Prasad, Survey of the problem of object detection in real images, International Journal of Image Processing (IJIP) 6 (2012) 441.

[7] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464–3468.

[9] N. Wojke, A. Bewley, Deep cosine metric learning for person re-identification, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 748–756.

[10] Y. Du, Y. Song, B. Yang, Y. Zhao, Strongsort: Make deepsort great again, 2022.

[11] N. Aharon, R. Orfaig, B.-Z. Bobrovsky, Bot-sort: Robust associations multi-pedestrian tracking, 2022.

[12] M. Wieczorek, B. Rychalska, J. Dabrowski, On the unreasonable effectiveness of centroids in image retrieval, 2021.

[13] D. Fu, D. Chen, J. Bao, H. Yang, L. Yuan, L. Zhang, H. Li, D. Chen, Unsupervised pre-training for person re-identification, 2020.

[14] K. Zhou, Y. Yang, A. Cavallaro, T. Xiang, Omni-scale feature learning for person re-identification, in: ICCV, 2019.

[15] A. Mohan, A. S. Kaseb, K. W. Gauen, Y.-H. Lu, A. R. Reibman, T. J. Hacker, Determining the necessary frame rate of video data for object tracking under accuracy constraints, in: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2018, pp. 368–371.