Multi-Camera People Tracking for AI City Challenge

Dominik Kroupa*
Supervised by: prof. Ing. Adam Herout Ph.D.†

Faculty of Information Technology
Brno University of Technology
Brno / Czech Republic

Abstract

Multi-target multi-camera pedestrian tracking (MTMCT) plays a crucial role in urban surveillance, public safety, and crowd behavior analysis, contributing to the advancement of smart city infrastructure by providing analysis of pedestrian movement through given areas. However, due to challenges such as the presence of severe occlusion and changes in appearance throughout the scene, robust and accurate MTMCT remains a significant challenge in complex environments. To address these challenges, this study proposes an offline pipeline for pedestrian tracking, consisting of three main stages: (1) generation of single-camera tracklets through pedestrian detection (with keypoint estimation) and appearance feature extraction, (2) refinement and completion of tracklets using appearance features and strategies to reduce identity switches, and (3) inter-camera association (ICA) via global ID assignment leveraging appearance features. Additionally, a model trained on detected body keypoints is employed for ground position estimation. The solution was evaluated in the AI City Challenge MTMCT Track in the previous year, achieving a Higher Order Tracking Accuracy (HOTA) score of 31.52%.

Keywords: Multi-Camera Tracking, Object Detection, Re-Identification, Single-Camera Tracking, AI City Challenge

1 Introduction

Multi-Target Multi-Camera tracking (MTMC) is a challenging task in computer vision, where the aim is to track multiple objects across multiple cameras. This can be used for crowd analysis. In this paper, we propose an offline framework consisting of three stages to deal with the MTMC tracking problem. Firstly, we generate single-camera tracklets and extract appearance features. Secondly, we perform refinement and completion for extracted tracklets by using time constraint conditions, appearance features, and information about potential identity switches when tracklets move close to each other. Finally, we perform the Inter-Camera Association (ICA) by using appearance features. For the estimation of the ground position for each person, we trained a regression model with detected body keypoints as input.

Our contributions can be summarized as follows:

- We explored person re-identification performance on the challenge dataset, investigating feature separability by training models with different loss functions (softmax and circle loss).
- We integrated a keypoint-based ground position estimation method, utilizing YOLOv8x-pose model's output.
- We implemented and evaluated an offline multi-stage MTMCT pipeline, comparing different method combinations.

2 Background and Previous Work

This section review key components and prior work related to our multi-camera tracking approach. We cover essential areas of object detection, person re-identification and single-camera tracking algorithms.

2.1 Object Detection

Convolutional neural networks (CNNs) have become the primary method for object detection tasks. Typically, these methods employ a backbone architecture to extract image features and a specialized structure to enable the network to identify object classes and generate bounding boxes. Two major types of object detectors have emerged from these approaches: single-stage and two-stage detectors.

The object detection approach of two-stage detectors involves using a network to generate object proposals initially, which are then verified in a second stage to determine if they correspond to an actual object. Additionally, the bounding boxes produced in the first stage are refined to fit objects more precisely in the second stage. *Faster R-CNN* [13] is the most notable example of a two-stage object detector, and its extension *Mask R-CNN* [5] enables the detection of object masks in addition to bounding boxes.

*dominikkroupa@gmail.com
†herout@fit.vut.cz

Single-stage object detectors get their name from generating refined bounding boxes in one pass through the network. This can be accomplished with anchor-based methods *e.g.* *YOLOv7* [16] or anchor-free methods such as *YOLOv8* [7], which we use in our MTMCT solution.

2.2 Person Re-identification

State-of-the-art person re-identification (ReID) methods primarily fall into two categories: convolutional neural network (CNN)-based and transformer-based approaches. CNN-based models, particularly ResNet [6] architectures, have demonstrated strong performance in ReID tasks. To further improve feature extraction capabilities, OSNet [27] was introduced, designed to learn omni-scale feature representations. This architecture effectively captures both fine-grained details, such as small logos on clothing, and larger structures like body shape, making it well-suited for person ReID. Transformer-based models [22] have gained traction in ReID due to their ability to capture long-distance dependencies through self-attention mechanisms.

Loss functions play a crucial role in optimizing ReID models. Although softmax loss remains widely used, alternative loss functions such as ArcFace loss [3] and Circle loss [15] have been introduced to enhance inter-class separability. In our work, we utilize the OSNet model and evaluate its performance with both softmax loss and circle loss to determine their impact on re-identification accuracy.

2.3 Tracking

The Kalman filter [8], used for state estimation, has been a popular component within tracking method for several decades. *SORT* [2] is a well-known tracking algorithm that combines the Kalman filter with the Hungarian algorithm for online tracking. Its extension, *DeepSORT* [19], enhances tracking performance by incorporating deep-learned appearance-based features. Building upon this, *StrongSORT* [4] introduces more robust feature extraction and maintains a single feature representation of each track through the use of Exponential Moving Average (EMA). *ByteTrack* [24] is also based on the Kalman filter and features an association strategy considering high and low-confidence detections. *BoT-SORT* [1] builds on principles of *StrongSORT* and *ByteTrack*, using association with and without appearance features, based on the detection confidence. We use *ByteTrack* and *BoT-SORT* in our approach to MTMCT challenge.

3 Multi-Camera People Tracking

This section presents details about each step in the proposed offline framework for our solution to Track 1, composed of three main modules: *(i)* detection with position estimation, single-camera tracking, and feature extraction, *(ii)* post-processing, *(iii)* inter-camera association. A detailed visualization of the modules involved in our proposed pipeline can be seen in Figure 1.

To adapt the framework to work in an online manner, single-camera post-processing *(ii)* needs to be removed, and the inter-camera association *(iii)* needs to be modified in order to associate tracklets after each processed frame. Such modifications increase the count of global ID switches, reducing the quality of the tracking results. This performance trade-off is why we decided to adopt an offline framework.

While our pipeline utilizes established tracking algorithms within stages, recent works demonstrate alternative approaches. For instance, some methods [23] use clustering of detections across frames based on feature similarity and further correction of the clustering results, thus removing the need for conventional single-camera tracking. Others [18] first project multi-camera views into a unified Bird's-Eye View (BEV) representation and perform detection and tracking in the BEV space.

3.1 Pedestrian Detection

For this track, we experimented with single-stage detectors *YOLOv7* [16] and *YOLOv8* [7]. Three models were used: basic *YOLOv7* model and the best-performing models equipped with pose estimation *YOLOv8x-pose* and *YOLOv8x-pose-p6*, all pre-trained on the MS COCO dataset [10] with no additional training data. To be able to report precise ground positions of the pedestrians, we propose a regression method based on the object's keypoints detected by *YOLOv8-pose* model. We trained a regression model with keypoints and their confidence scores as input. The model is a feedforward neural network comprising of two hidden layers (128 and 16 units respectively), each using ReLU activation followed by Dropout (0.1). MSE loss was used to train the model. The model's output is the 2D position of the anchor point in the image space for each set of keypoints, as shown in Figure 2. This point is then transformed via a known homography to the world plane.

3.2 Pedestrian Re-identification

Person re-identification (ReID) is a crucial step both in single-camera and multi-camera tracking, where the goal is to match pedestrian detections across frames and cameras. For our approach, we utilize deep-learning person re-identification library called *Torchreid* [26] to experiment with ReID models. We chose the best performing model *osnet_x1_0* from the provided model ZOO, which is a variant of *OSNet* [27] that employs its baseline channel width (i.e. without any scaling), which would decrease accuracy.

3.2.1 Person Re-identification Dataset

To test behavior of the *osnet_x1_0* on the provided AI City Challenge Track 1 synthetic dataset [17], we decided

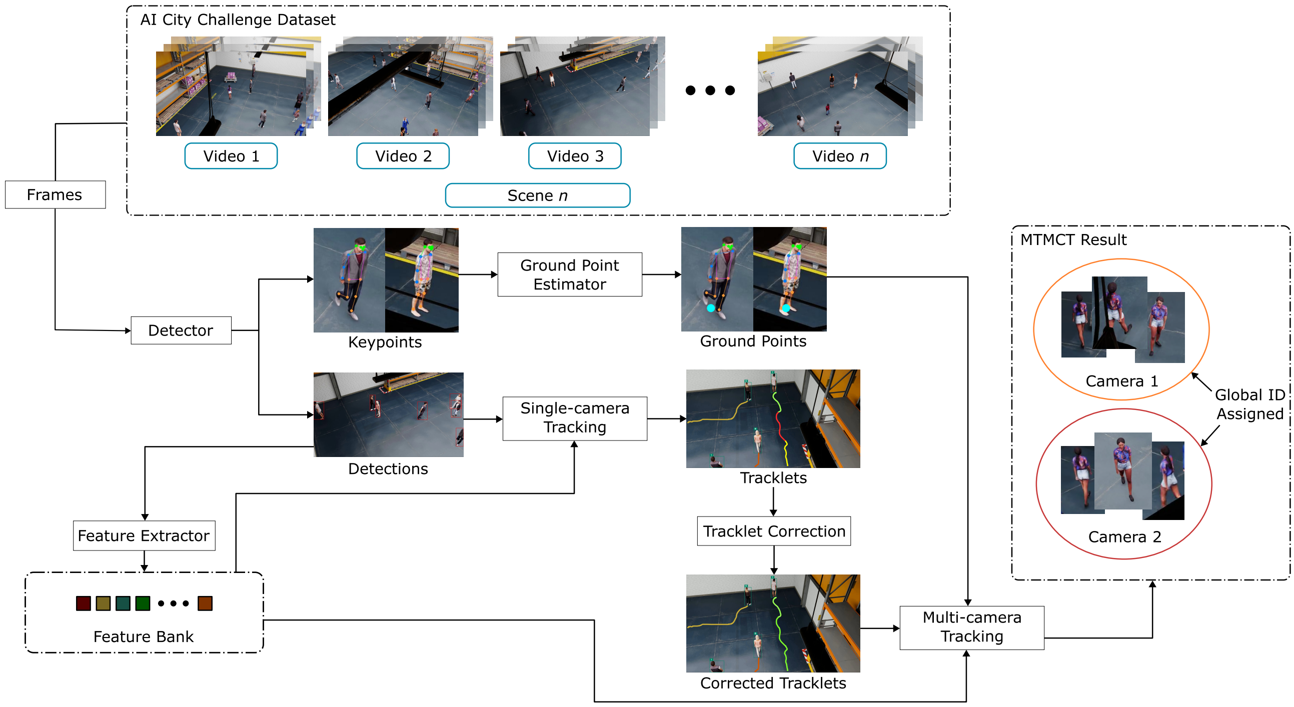

Figure 1: Detailed overview of the proposed offline MTMCT pipeline.

Figure 2: Detected skeleton keypoints (blue), estimated ground plane positions (green), ground truth values (red).

to use the training subset to create person ReID dataset. The created dataset consists of 429 unique identities with 103,309 images captured from 360 different cameras.

To create training and testing subsets, we first sorted the identities based on the number of images belonging to each person (from most to fewest). We then assigned identities to the train or test set by alternating based on their position (even or odd index) in the sorted list. This procedure resulted in training and testing sets each containing approximately half of the total identities and images.

For evaluation, the query set was constructed by selecting one image for each identity from camera in which it appeared within the test set.

3.2.2 Improving Inter-class Separability

To enhance the discriminative power of learned embeddings, we integrated circle loss into the Torchreid framework, training the model on our dataset alongside the standard softmax loss for comparison. Figure 3 (top) shows histogram separability between positive and negative pairs in the created dataset, using the *osnet_x1_0* model with provided weights trained on *Market-1501* [25] dataset using softmax loss. The class separability is reasonable but has room for improvement. After training the model on the train set of created ReID dataset, the inter-class separability improved, as can be seen in Figure 3, bottom. Model trained with circle loss yields results comparable to the softmax trained model.

3.3 Pedestrian Tracking

Previous MTMC tracking works [21, 11, 14, 9] use feature appearance vectors of objects in multiple object tracking followed by post-processing steps that include reduction of identity switches. In this paper, we experimented with both *ByteTrack* [24] and *BoT-SORT* [1]. While *BoT-SORT* promises fewer identification switches, it requires additional computational time when compared to *ByteTrack*, which, on the other hand, relies more on post-processing due to frequent cases of new ID assignment for long-term object occlusions, as can be seen in Figure 5, top.

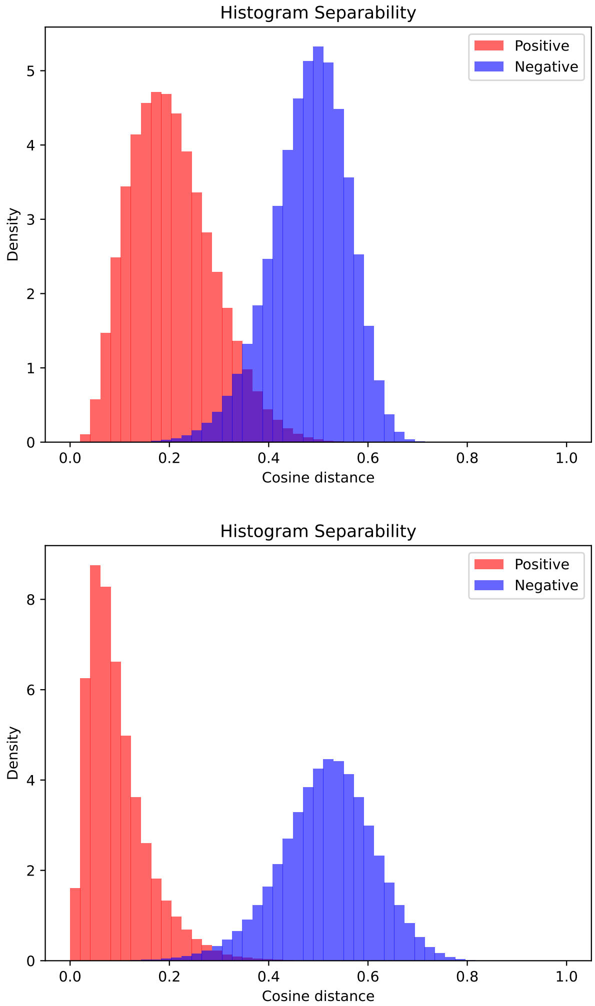

Figure 3: Comparison of histogram separability on the created ReID dataset. (Top) shows separability of the *osnet_x1_0* model trained on Market-1501 dataset and (bottom) shows the model trained on the test set of created ReID dataset. Both models were trained using softmax loss.

3.4 Post-processing

3.4.1 Single-camera Identity Switch Correction

Single-camera tracklets may suffer from identity switching due to chaotic pedestrian movement and mutual occlusion. To tackle this problem, we first calculate the overlap between bounding boxes after each tracker update during single-camera tracking. When the overlap between bounding boxes exceeds given threshold, the overlap time for the given tracklet is saved. Then, in post-processing, we use this information to check the pedestrian similarity distance before and after the overlap. Similarity bigger than a given threshold signals that identity switch occurred and the tracklet is split into two independent tracklets. An example of identity switching and its correction is shown in Figure 4.

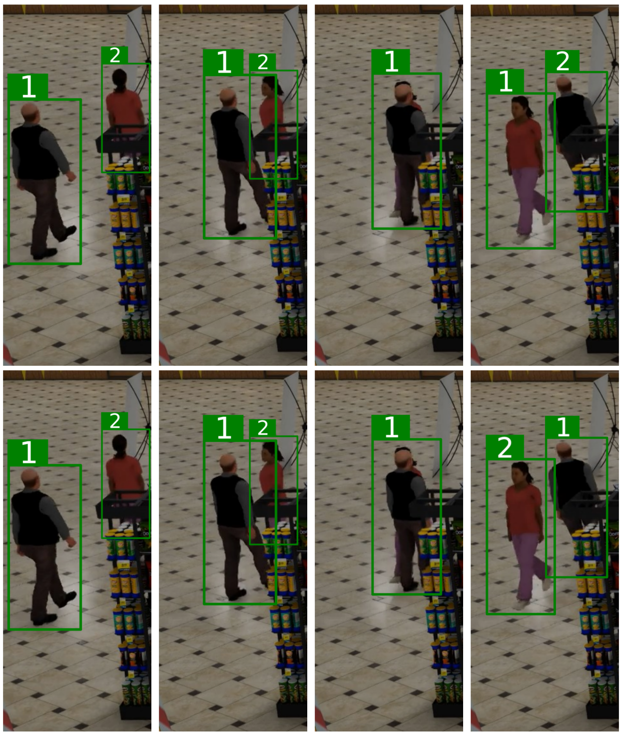

Figure 4: Tracking results before (top) and after (bottom) applying post-processing to deal with identity switch occurrences.

3.4.2 Single-camera Track Merging

Single-camera tracklets contain many disconnected tracklets with different identities belonging to the same pedestrian. This happens in cases of long-term pedestrian occlusion, as shown in Figure 5, top. We tackle this problem by creating a constraint set for each tracklet ID, including all other tracklet IDs that can be merged with the given tracklet based on time conditions. These time conditions allow us to merge two tracklets without overlap, meaning it could be the same pedestrian. In practice, constraint set creation looks as follows:

$$T_i = \{T_j \mid T_j \neq T_i \wedge \neg\text{overlap}(T_i, T_j)\}. \tag{1}$$

To further utilize spatiotemporal tracklet information, we add a distance constraint, checking whether trajectory T_i could be merged with trajectory T_j based on the traveled distance between them in the given time.

Inspired by [14, 9], after constraint sets creation, a distance matrix of size $N_c \times N_c$ is constructed where N_c is the number of tracklets in camera c. The matrix is filled by mean feature cosine distances between tracklets T_i, T_j:

$$A_{N\times N} = \begin{bmatrix} cos(T_1, T_1) & \dots & cos(T_1, T_n) \\ \vdots & \ddots & \vdots \\ cos(T_n, T_1) & \dots & cos(T_n, T_n) \end{bmatrix} \tag{2}$$

Then, the matrix indices are sorted from the lowest distance values to the highest by applying `argsort`(A).

Figure 5: Tracking results before (top) and after (bottom) applying post-processing to solve tracklet fragmentation.

These indices, i, j representing tracklets T_i, T_j respectively, are processed step by step. For tracklets to be merged, the following conditions must be met:

$$cos(T_i, T_j) < \tau \tag{3}$$

$$T_j \in \mathrm{c}(T_i) \tag{4}$$

where $c(\cdot)$ represents the constraint set for a given tracklet defined at the beginning of this section, $c(T_i)$ is a set of tracklet identities that can be merged with T_i. In case two or more tracklets are joined together, condition (4) is modified as follows:

$$T_j \in \mathrm{c}(T_1) \wedge \cdots \wedge \mathrm{c}(T_n), \tag{5}$$

where $T_1 \dots T_n$ represent tracklets already joined together into one single tracklet. After this process, the tracklets belonging to the same person can be merged, as shown in Figure 5.

3.5 Inter-Camera Association

In this final module, all single-camera tracklets are expected to be correctly merged so that each tracklet represents one pedestrian. In that case, to each tracklet T_i from camera C_i, only one tracklet T_j or none is assigned from camera C_j. We modify the approach from Section 3.4.2 to match pedestrians between cameras. This time, we depend only on the distance matrix matching; time constraints are not created because the same pedestrian may or may not have trajectories overlapping in time.

Distance matrix of size $N_{ci} \times M_{cj}$ is created, where N_{ci}, M_{cj} is the number of tracklets in camera c_i and c_j respectively (ignoring second-level subscript for better readability), is filled by mean feature cosine distances between tracklets T_{ci}, T_{cj}:

$$A_{N\times M} = \begin{bmatrix} cos(T_{i1}, T_{j1}) & \dots & cos(T_{i1}, T_{jm}) \\ \vdots & \ddots & \vdots \\ cos(T_{in}, T_{j1}) & \dots & cos(T_{in}, T_{jm}) \end{bmatrix}, \tag{6}$$

Table 1: Results of methods evaluated on the full test set.

Method	HOTA	DetA	AssA	LocA
Y7-byte	**31.52**	47.66	21.68	89.81
Y8x-ankles	26.07	39.82	17.55	81.87
Y8x-regress	31.02	48.2	20.45	91.55
Y8p6-bot	30.63	50.09	19.6	91.17

where T_{ix} and T_{jx} represent tracklet x from camera i or j. Then, matrix indices are sorted from the lowest distance values to the highest by applying $\texttt{argsort}(A)$. These indices i, j representing tracklets T_i, T_j, are processed step by step. For tracklets to be merged, distance condition (3) from Section 3.4.2 must be met.

4 Experiments and Results

The *AI City Challenge* [17] MTMC tracking dataset consists of synthetic data, generated using the *NVIDIA Omniverse Platform*, containing Full-HD videos at 30 FPS from a total of 1300 cameras. Camera matrices are provided for each camera. The videos are divided into 90 subsets, 40 for training, 20 for validation, and 30 for testing. The subsets in the dataset are captured from many different scenarios, such as from a store or warehouse. Our experiments are summarized in Table 1. Our results were evaluated on the official AI City Challenge server using the standard Higher Order Tracking Accuracy (HOTA) metric [12], which provides a unified score balancing detection, association and localization accuracy.

Y7-byte This approach integrates the *YOLOv7* detection model along with *ByteTrack* tracking algorithm and *Torchreid* model trained as described in Section 3.2. The feature vector distance threshold τ was set to 0.2. This method achieves a 31.52 HOTA score on the full test set.

Y8x-ankles In this method we used the *YOLOv8x-pose* model for detecting pedestrians and their keypoints. From these keypoints, the center of each pedestrian's ankle keypoints was used to estimate ground position more accurately. This method achieves a 26.07 HOTA score on the full test set.

Y8x-regress In this method, we continue to employ the *YOLOv8x-pose* model with the adjustment where we now feed the keypoints data along with their confidences to our regression model. This enhancement is designed to refine our estimation of a person's ground position, achieving improved accuracy even when a pedestrian's lower body is occluded. This method achieves a 31.02 HOTA score on the full test set.

y8p6-bot This approach incorporates the best *YOLOv8x-pose-p6* detection model, combined with the *BoT-SORT* tracker and the *CLIP* model for feature extraction along with our model for *CLIP* embedding reduction. Based on experiments done on our reduced validation dataset, which contained only the first scene, we set the feature vector threshold τ to 0.4. Additionally, we configured *BoT-SORT* to maintain a tracklet history of 10 seconds and allowed the matching of previous tracks with new ones even when the *IoU* between bounding boxes is 0. Given the tracker's employment of re-identification, we deactivated our *single-camera identity switch correction* mechanism mentioned in Section 3.4.1. This method achieves a 30.63 HOTA score on the full test set. Given the need to set feature distance threshold τ to 0.4 and given the AssA result of 19.6, it seems that using the *CLIP* model in conjunction with the reduction model is not enough to ensure better discriminative ability.

Summary While the *YOLOv7* model combined with *ByteTrack* and *Torchreid* achieved the best HOTA results, our regression model paired with *YOLOv8-pose* detection model closely competes. We believe that with further examination and optimization of detection, tracking, and re-identification parameters, this approach could surpass the *Y7-byte* configuration.

5 Conclusion

We participated in Track 1 of *2024 AI City Challenge*. Although our HOTA score of 31.5208 was not competitive (the winning score [20] was 62.2175), we believe that the findings presented in this paper can benefit other researchers. Furthermore, experiments with ReID conducted after the challenge ended suggest a possible increase in the association metric. Along with increasing association by changing tracking parameters and conducting ReID training experiments, future work will also focus on further reducing identity switches and improving inter-camera association. Future work could also explore leveraging more recent pose estimation models, which became available after the end of last year's AI City Challenge.

Acknowledgments

The authors would like to thank Ing. Marketa Jurankova Ph.D. for providing the pre-trained ground position estimation model used in this work.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing*, pages 3464–3468. IEEE, 2016.

[3] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, October 2022.

[4] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969. IEEE, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[7] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, January 2023.

[8] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[9] Fei Li, Zhen Wang, Ding Nie, Shiyi Zhang, Xingqun Jiang, Xingxing Zhao, and Peng Hu. Multi-camera vehicle tracking system for ai city challenge 2022. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3264–3272, 2022.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[11] Chong Liu, Yuqi Zhang, Hao Luo, Jiasheng Tang, Weihua Chen, Xianzhe Xu, Fan Wang, Hao Li, and Yi-Dong Shen. City-scale multi-camera vehicle tracking guided by crossroad zones. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops)*, pages 4124–4132, 2021.

[12] Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2):548–578, October 2020.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[14] Andreas Specker, Lucas Florin, Mickael Cormier, and Jürgen Beyerer. Improving multi-target multi-camera tracking by track refinement and completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3198–3208, 2022.

[15] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6397–6406, 2020.

[16] 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. *arXiv preprint arXiv:2207.02696*, 2022.

[17] Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Pranamesh Chakraborty, Sanjita Prajapati, Quan Kong, Norimasa Kobori, Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Fady Alnajjar, Ganzorig Batnasan, Ping-Yang Chen, Jun-Wei Hsieh, Xunlei Wu, Sameer Satish Pusegaonkar, Yizhou Wang, Sujit Biswas, and Rama Chellappa. The 8th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 7261–7272, June 2024.

[18] Yizhou Wang, Tim Meinhardt, Orcun Cetintas, Cheng-Yen Yang, Sameer Satish Pusegaonkar, Benjamin Missaoui, Sujit Biswas, Zheng Tang, and Laura Leal-Taixé. Mcblt: Multi-camera multi-object 3d tracking in long videos, 2025.

[19] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE International Conference on Image Processing*, pages 3645–3649. IEEE, 2017.

[20] Zhenyu Xie, Zelin Ni, Wenjie Yang, Yuang Zhang, Yihang Chen, Yang Zhang, and Xiao Ma. A robust online multi-camera people tracking system with geometric consistency and state-aware re-id correction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7007–7016, 2024.

[21] Jin Ye, Xipeng Yang, Shuai Kang, Yue He, Weiming Zhang, Leping Huang, Minyue Jiang, Wei Zhang, Yifeng Shi, Meng Xia, and Xiao Tan. A robust mtmc tracking system for ai-city challenge 2021. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 4039–4048, 2021.

[22] Mang Ye, Shuoyi Chen, Chenyue Li, Wei-Shi Zheng, David Crandall, and Bo Du. Transformer for object re-identification: A survey, 2024.

[23] Ryuto Yoshida, Junichi Okubo, Junichiro Fujii, Masazumi Amakata, and Takayoshi Yamashita. Overlap suppression clustering for offline multi-camera people tracking. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7153–7162, 2024.

[24] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *arXiv preprint arXiv:2110.06864*, 2021.

[25] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision*, 2015.

[26] Kaiyang Zhou and Tao Xiang. Torchreid: A library for deep learning person re-identification in pytorch. *arXiv preprint arXiv:1910.10093*, 2019.

[27] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.