Traffic & Transportation

An M-DeepSORT Algorithm for Pedestrian Detection and Tracking Based on Video Images – A Case Study in Ji-nan Subway Station

Wei ZHANG[1], Chuang ZHU[2], Yunchao QU[3], Guanhua LIU[4], Der-Horng LEE[5]

Original Scientific Paper
Submitted: 6 Mar 2024
Accepted: 28 Aug 2024

[1] 13705400820@163.com, Beijing Jiaotong University, School of Systems Science
[2] Corresponding author, zhuchuang@bjtu.edu.cn, Beijing Jiaotong University, School of Systems Science
[3] ycqu@bjtu.edu.cn, Beijing Jiaotong University, School of Systems Science
[4] dclgh@163.com, Beijing Jiaotong University, School of Systems Science
[5] dhlee@intl.zju.edu.cn, Zhejiang University, Zhejiang University; University of Illinois at Urbana-Champaign (ZJU-UIUC) Institute

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT
With ongoing urbanisation, the subway has become a vital component of modern cities, catering to the escalating demands of a mobile population. However, the increasing complexity of passenger flows within subway stations presents challenges for operations management. To optimise subway operations and enhance safety, researchers have focused on extracting and analysing pedestrian trajectories within subway stations. Traditional trajectory extraction methods face limitations due to manual feature design and multi-stage processing. Leveraging advancements in deep learning, this paper integrates M-DeepSORT with YOLOv5 and proposes a feature association matching approach that addresses trajectory drift issues through simultaneous consideration of motion and appearance matching. A confidence-based (CB) Kalman filtering method is proposed to address the issue of random noise in pedestrian detection within subway scenes. The introduction of a momentum-based passenger trajectory centre update method reduces jitter, resulting in smoother trajectory extraction. Experimental results affirm the effectiveness of the proposed algorithm in detecting, tracking and statistically analysing subway station corridor passenger flow trajectories, demonstrating robust performance in diverse subway station scenarios.

KEYWORDS
passenger trajectory tracking; CB Kalman filtering; trajectory update; momentum; M-DeepSORT.

1. INTRODUCTION

Cities are grappling with the increasing fluidity of population movement and the growing complexity of transportation demands, with urban public transit systems playing a pivotal role [1, 2]. Among these systems, the subway, recognised for its efficiency and speed, caters to the daily travel needs of a substantial number of passengers [3]. However, as urbanisation progresses, the organisation of passenger flows within subway stations becomes more intricate. In the pursuit of optimising operations, enhancing service quality and bolstering safety management capabilities, researchers are increasingly focusing on extracting and analysing pedestrian trajectories within subway stations [4]. By analysing passenger trajectories, we gain valuable insights into the travel behaviour patterns of passengers, encompassing aspects such as travel paths, dwell times and transfer habits. This process helps establish a more precise model for understanding travel behaviour. Beyond offering theoretical underpinnings for urban transportation planning, this analysis also enhances our comprehension of the spatial dynamics within subway stations, including popular areas, congestion points and flow channels. Consequently, it optimises the utilisation efficiency of station space. Operators can refine train departure intervals, enhance crowd guidance and establish judicious station facilities based on these insights,

thereby elevating transportation efficiency and overall service quality. Moreover, passenger trajectory analysis possesses the capability to forecast crowd congestion, guide and redirect passengers, and respond to emergencies. This functionality provides tangible support for the safety management and emergency event handling of subway stations. Furthermore, data-driven decisions, such as intelligent guidance services and safety evacuation plans based on passenger trajectories, are set to propel urban public transit systems toward greater efficiency and intelligence. Therefore, a comprehensive study of pedestrian trajectories within subway stations holds the promise of providing robust support for the modernisation and optimisation of urban public transit systems.

The current landscape of research on pedestrian trajectory tracking and extraction methods can be broadly categorised into two groups: approaches rooted in traditional machine vision and those leveraging deep learning techniques. Traditional machine vision-based pedestrian detection methods rely on pedestrian features such as height, the colour of clothes and histogram of oriented gradient (HOG), and employ classifiers like support vector machine (SVM) and random forest to locate pedestrians. Hand-crafted features play a crucial role in pedestrian detection, with the HOG [5] being a widely utilised descriptor. The fundamental concept underlying HOG is the representation of local object appearance and shape in an image through the intensity distribution of gradients or edge directions. The image is partitioned into small, connected regions, and for the pixels within each region, a histogram of gradient directions is generated. The final descriptor is a concatenation of these histograms. Recent advancements in this field introduced a local sub-descriptor known as colour self similarity (CSS) [6]. CSS compares colour histograms within a HOG-detected window, emphasising, for instance, the high similarity between colour histograms from two arms. Extensive research has been dedicated to pedestrian detection [7, 8], with more than sixteen different detectors being benchmarked [9] against various public datasets. A significant portion of this research focused on hand-designed features, as detailed in [9]. These features primarily relied on window-sliding techniques, employing support vector machines (SVM) for classification. Additionally, alternative methods, rooted in Viola and Jones' Adaboost framework [10], and several variations of the HOG method [11], were thoroughly evaluated in [9]. However, they heavily depend on manual feature design, requiring significant human intervention. When subway station environments change or passenger targets are blocked and deformed, these methods struggle to accurately track.

In recent years, convolutional neural network (CNN)-based deep learning technologies have become prevalent in pattern recognition and target detection. Deep learning detection algorithms are generally categorised into two types: two-stage algorithms based on candidate regions and single-stage algorithms based on regression. Girshick et al. [12] introduced the regional convolutional neural network (RCNN) in 2014, employing a three-step process: candidate region selection, CNN feature extraction and classification/boundary regression. Subsequent enhancements by Ren et al. [13] resulted in faster RCNN, improving candidate box generation. In contrast to two-stage algorithms, single-stage algorithms based on regression, such as single shot multibox detector (SSD) and you only look once (YOLO), predict category probability and object coordinates directly through CNN, offering faster detection speed. The emergence of CNNs has significantly impacted computer vision research groups specialising in pedestrian detection, demonstrating improved and more reliable results in recent analyses [14]. The proposal of the YOLO algorithm [15] greatly improves the detection speed of the single-stage detector. The latest YOLOv5, implemented in the PyTorch framework, maintains equivalent detection accuracy to YOLOv4 while achieving a nearly 90% reduction in model size, thereby lowering computational costs.

With the continuous enhancement of both accuracy and speed in target detection algorithms, the development of target tracking by detection (TBD) methods, relying on the outcomes of target detection, has progressed rapidly [13, 15, 16]. Simple online and realtime tracking (SORT) [17] stands out as a classical multi-target tracking algorithm. This tracker employs the Kalman filter for motion trajectory prediction, and the Hungarian algorithm optimises the allocation between detection results and the tracker's prediction outcomes. DeepSORT [16], building upon the SORT algorithm, introduces the cascade matching strategy to mitigate identity transformation-related mispredictions. Therefore, in recent years, the combination of YOLO and DeepSORT for pedestrian trajectory recognition has attracted widespread attention [18-21]. Song et al. [18] designed an intelligent helmet recognition system, which combined the multi-target tracking algorithm DeepSORT and YOLOv5 detector. Zhao et al. [19] utilised the YOLO algorithm to implement passenger boarding and alighting detection and statistics based on video images from buses. In the realm of pedestrian tracking, the most recent advancements in MOT research have offered novel perspectives for the identification and tracking of pedestrians within subway stations. In addition, scholars have proposed several novel

approaches for multi-object detection. Wang et al. [22] proposed a new instance of joint MOT approach based on graph neural networks (GNNs) to learn discriminative features for detection and data association. Fukui et al. [23] proposed an end-to-end MOT model, TicrossNet, which is composed of a base detector and a cross-attention module. This model does not require attached modules, such as the Kalman filter, Hungarian algorithm, transformer blocks or graph networks. Daniel Stadler and Jurgen Beyerer [24] introduced a novel occlusion handling strategy that explicitly models the relation between occluding and occluded tracks outperforming the feature-based approach, while not depending on a separate re-identification network. Zhang et al. [25] presented a simple, effective and generic association method, tracking by associating almost every detection box instead of only the high-score ones. To improve the performance of data association, Li et al. [26] developed a simple, effective, bottom-up fusion tracker for re-identity features, named SimpleTrack, and proposed a new tracking strategy which can mitigate the loss of detection targets. Yaoyao Si and Yi Zhang [27] proposed IAMOT, a simple yet effective network based on anchor-free architecture to utilise an additional attention module to weaken the occurrence of ID switches. Zhou et al. [28] proposed an adaptive joint learning approach, called UnionTrack, for MOT in this paper. It is handled with the problem of uniform learning between tasks in the online MOT system. In terms of pedestrian tracking, tracking-by-detection is the mainstream framework [29]. Xin Xiao and Xinlong Feng [30] proposed a comprehensive approach for pedestrian tracking, combining the improved YOLO object detection algorithm with the OC-SORT tracking algorithm. These research advancements have laid a foundation for our study. However, tracking pedestrians within a subway station presents numerous challenges, including complex environmental conditions, random noise and the diversity of pedestrian behaviours such as occlusions. The confined spaces and rapid movement dynamics within subway stations make accurate trajectory recognition and extraction difficult due to frequent occlusions and overlapping paths among pedestrians. Additionally, the dynamic nature of subway stations with fluctuating passenger volumes and varying movement patterns introduces uncertainties and complexities in trajectory recognition. Furthermore, the presence of random errors and jitter in pedestrian trajectories necessitates advanced algorithms capable of maintaining consistent and reliable tracking results. These multifaceted challenges collectively hinder the effectiveness of traditional tracking methods, highlighting the need for innovative approaches to enhance pedestrian trajectory recognition and extraction in subway environments. Therefore, this paper proposes an integrated and enhanced method for pedestrian trajectory recognition and extraction in subway stations by combining YOLOv5 and M-DeepSORT. Firstly, this paper integrates the YOLOv5 detector with M-DeepSORT, simultaneously enhancing passenger detection accuracy through the utilisation of the Distance-IoU loss. This approach effectively mitigates the risk of misjudgements caused by target occlusion or deformation. Secondly, addressing the issue of random noise in pedestrian detection within subway scenes, the paper proposes the confidence-based (CB) Kalman filtering method. This innovative approach enhances the robustness of pedestrian tracking in noisy subway environments. Thirdly, this paper introduces a novel momentum-based passenger trajectory centre update algorithm, which mitigates passenger trajectory jitter caused by random errors. This innovative approach leverages momentum information from previous observations, enabling a more stable and continuous estimation of passenger trajectory. Finally, to ascertain the practical applicability and robustness of our proposed model, extensive validation was conducted in diverse real-world scenarios, ensuring its effectiveness across various conditions and settings.

2. THEORETICAL BACKGROUND

2.1 Object detection

Compared with two-stage target detection algorithms, the YOLO series, including YOLOv5, excels by eliminating the initial rough positioning step, thereby directly obtaining object category and position information simultaneously. This streamlined approach significantly enhances detection speed. This acceleration is crucial for real-time tracking in dynamic subway environments. The grid-based approach of YOLOv5 divides the input image into S×S cells, with each cell assigned to detect objects whose centres fall within its boundaries. These cells predict precise position details, confidence scores and category labels for B bounding boxes, facilitating efficient pedestrian capture amidst subway passenger flows. YOLOv5 incorporates non-maximum suppression (NMS) to refine detection results. By filtering out bounding boxes with lower confidence scores and retaining those with the highest confidence score for subsequent loss function computation, it ensures high-quality target detections, a fundamental prerequisite for accurate pedestrian

trajectory tracking in subway systems. Furthermore, YOLOv5's implementation within the PyTorch framework is advantageous due to its lightweight model, minimising computational and memory costs, which is highly beneficial for resource-constrained subway passenger flow tracking systems. Lastly, YOLOv5 maintains detection accuracy comparable to YOLOv4 on benchmark datasets like COCO. This level of accuracy is essential for reliable and precise pedestrian tracking within the demanding subway environment. Therefore, based on its streamlined detection process, grid-based approach, NMS refinement, PyTorch integration and maintained detection accuracy, YOLOv5 is the apt choice as the foundational model for subway passenger flow pedestrian trajectory tracking.

2.2 Multi-object tracking

In recent years, with the advancements in target detection technology, tracking based on detection has emerged as the prevailing approach in the field of multi-object tracking (MOT). One notable example of this paradigm is the SORT algorithm, which combines the Kalman filter and the Hungarian algorithm. SORT utilises the intersection over union (IoU) between detection and tracking results as the cost matrix for the Hungarian algorithm, resulting in an effective and straightforward tracking methodology. However, SORT exhibits limitations in its inability to account for content feature matching, leading to frequent identity switches and interruptions in tracking when targets become occluded.

To address this issue, the DeepSORT algorithm integrates both motion and appearance information of targets into the association metric and incorporates a cascade matching strategy. This innovation dramatically reduces the occurrence of identity switches and tracking failures. The DeepSORT tracking process begins with identifying targets in each image using a detector. Subsequently, the tracker is initialised based on the detector's results, and the target's trajectory is predicted using the Kalman filter. Using the Hungarian algorithm, tracks are matched between consecutive images and the prediction results are continuously updated. DeepSORT assigns a tracker to the new detection results in each frame, with the emergence of a new track being confirmed if the tracker's predictions match the detection results for three consecutive frames; otherwise, the tracker is removed. In the context of this paper, DeepSORT is employed for tracking pedestrian trajectories within subway stations.

3. METHODS

To address the challenges posed by environmental noise, obstructions affecting pedestrians and mutual occlusions between pedestrians within subway stations, we have developed an innovative method for passenger recognition and tracking. Within the scope of extracting pedestrian trajectories within subway stations, the harmonious integration of advanced object detection and tracking technologies presents an enhanced method for the precise and continuous reconstruction of pedestrian movements. The object detection component efficiently identifies objects, including pedestrians, in images, yielding vital positional and class information. This is seamlessly complemented by the object tracking mechanism, proficiently monitoring objects across multiple video frames, thereby ensuring the creation of consistent trajectory sequences. The integration of these two components involves three key steps in the pedestrian trajectory extraction process:

1) Object detection: The system performs object detection on each frame of surveillance videos within the subway station, accurately locating pedestrians within the frames.
2) Object tracking: The tracking component utilises the extracted features from the detected pedestrians to track them seamlessly across consecutive video frames. It effectively associates objects in the current frame with their counterparts in the previous frame, maintaining trajectory continuity.
3) Trajectory generation: Successful tracking across multiple consecutive frames results in the creation of continuous pedestrian trajectories, depicting their paths of movement.

This comprehensive approach, combining robust object detection with advanced tracking techniques, not only ensures the effective extraction and tracking of pedestrian trajectories within subway stations but also provides invaluable, accurate data to support subway operation and management. This, in turn, contributes to optimising the passenger experience and enhancing overall service quality. The fundamental algorithmic framework for this approach is illustrated in *Figure 1*.

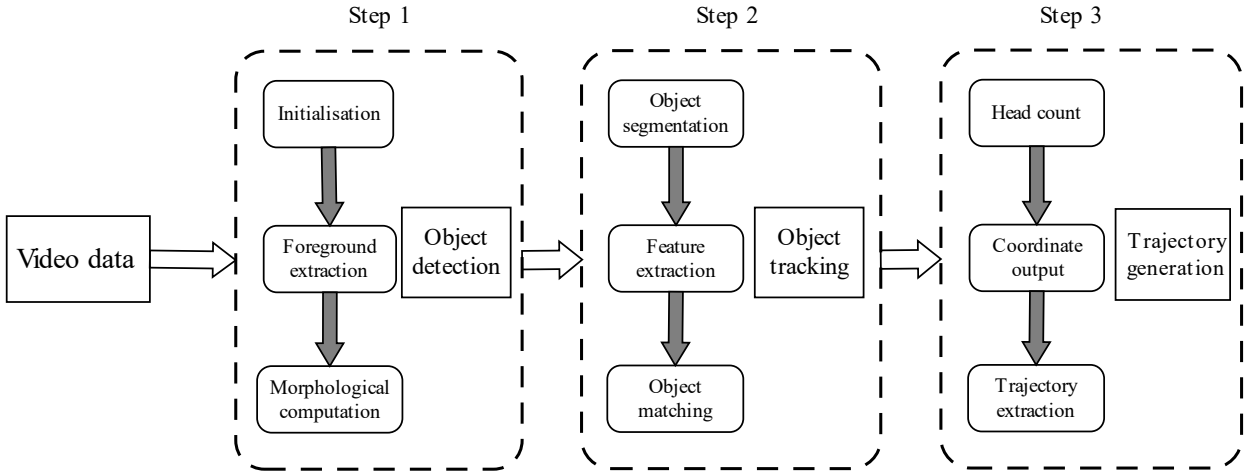


Figure 1 – Basic framework of the algorithm

In this study, we employ a combination of state-of-the-art methodologies to achieve precise and continuous pedestrian trajectory extraction. Specifically, our approach leverages YOLOv5 as the object detection component to efficiently identify and locate pedestrians within surveillance images. YOLOv5, recognised for its effectiveness in simultaneous multi-object detection and providing essential object information, plays a pivotal role in this process. Furthermore, we enhance the tracking aspect of our framework by incorporating a modified DeepSORT algorithm. This improved DeepSORT algorithm excels at tracking objects across multiple video frames, ensuring the generation of continuous and stable pedestrian trajectories. By harnessing the strengths of YOLOv5 for detection and the enhanced DeepSORT for tracking, we achieve a robust and accurate solution for pedestrian trajectory extraction within subway stations. Next, we will provide detailed introductions to YOLOv5 and our improved DeepSORT algorithm, respectively.

3.1 Improved YOLOv5 for passenger detection

In the context of passenger trajectory recognition within subway stations, the monitoring system typically relies on embedded devices with limited computing power, making it unfeasible to deploy large-scale detection models. Furthermore, passenger trajectory recognition relies on monitoring video, where the distance between passengers and cameras varies, leading to differences in the sizes of the tracked targets. YOLOv5 [31] offers distinct advantages with its compact model size, low computational overhead and high detection accuracy for small targets. This capability aligns perfectly with the need for accurate and real-time recognition of passenger trajectories in video frames, rendering it a highly practical choice for this application.

YOLOv5 is an advanced deep learning model designed for object detection, comprising four key components: the input module, backbone network, neck structure and head module, as shown in *Figure 2*. The input module preprocesses data through techniques such as mosaic data augmentation, adaptive anchor computation and adaptive image scaling, enhancing the model's adaptability to diverse datasets. The backbone network incorporates the focus structure, cross stage partial network (CSPNet) and spatial pyramid pooling (SPP) structure to extract features at different levels through deep convolution. The focus component downsamples the feature image via slicing operations while preserving the original image information. CSP reduces computation, improving inference speed; and SPP achieves feature extraction from the same feature map at different scales, contributing to enhanced detection accuracy. The neck structure combines feature pyramid networks (FPN) with path aggregation network (PAN). FPN transmits semantic information from top to bottom, while PAN transmits positional information from bottom to top, collectively reinforcing the network's feature fusion capabilities. The head module performs predictions on feature maps by calculating bounding box loss and employing non-maximum suppression (NMS) to generate target bounding boxes and predict categories.

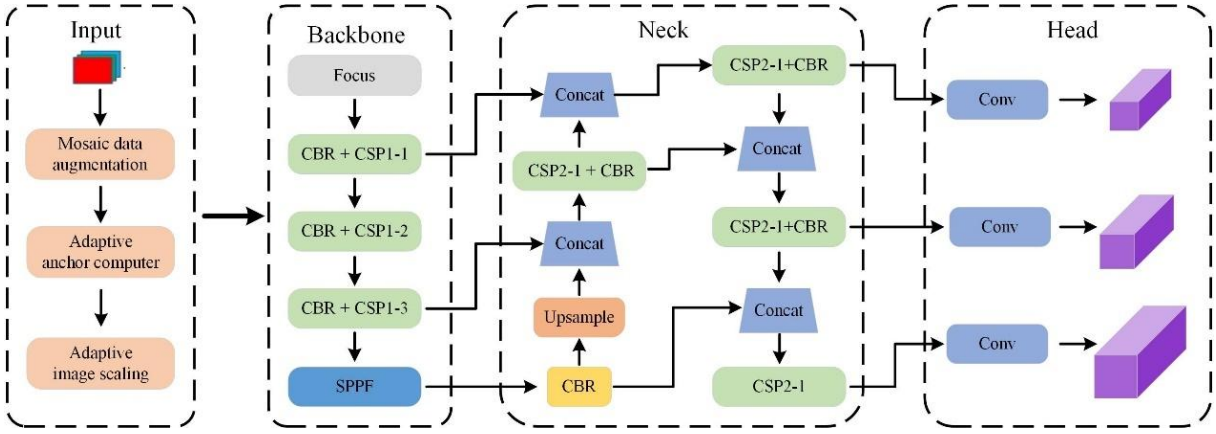


Figure 2 – The network structure of YOLOv5

In traditional YOLOv5 implementations, the IoU loss function has been conventionally adopted due to its effectiveness in providing a coherent framework for target detection. The regression loss function is a pivotal component in target detection algorithms, serving a critical role in evaluation and performance enhancement. The IoU loss exclusively focuses on the intersecting region between the predicted bounding box and the ground truth bounding box, demonstrating commendable scale invariance. The computation formula for IoU is as follows:

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \tag{1}$$

where $B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$ is the ground-truth and $B = (x, y, w, h)$ is the predicted box. Traditionally, ℓ_n-norm (e.g. $n = 1$ or 2) loss is applied to the coordinates of bounding boxes (B and B^{gt}) to quantify the spatial dissimilarity between bounding boxes, as advocated by various studies [12, 15, 32, 33]. However, as proposed in prior research [34, 35], the utilisation of ℓ_n-norm loss may not be the most suitable option for optimising the IoU metric. In the work by [34], the adoption of IoU loss is recommended to enhance the IoU metric, aiming for improved accuracy in object detection.

$$\mathcal{L}_{IoU} = 1 - \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \tag{2}$$

In general, the IoU distance effectively represents the spatial relationship between the prediction box (referred to as the projected box) and the detection box within the detection space. IoU always maintains a value greater than 0 and remains unaffected by any scaling of either the prediction frame or the detection frame. Nonetheless, the IoU loss function exclusively operates when bounding boxes exhibit overlap and does not provide any gradient information in scenarios where they do not overlap. Therefore, the IoU distance is not an ideal criterion for assessing the match between the detection box and the prediction box, as illustrated in *Figure 3.*

(a) (b)

Figure 3 – No spatial alignment exists between the detection frame and the prediction frame:
a) The detection frame is in close proximity to the prediction frame; b) The detection frame is distant from the prediction frame

From *Figure 3*, when the red detection frame does not overlap with the blue track prediction frame, both cases in *Figure 3a* and *Figure 3b* result in IOU values of 0. This makes it impossible to assess the difference in matching quality between them, which can lead to erroneous matches.

Additionally, situations may arise where the IoU value between the trajectory prediction box and the detection box is identical, even though the overlap positions differ. As shown in *Figure 4*, the IoU distance is likewise insufficient to assess the matching degree between the detection frame and the prediction frame in such cases.

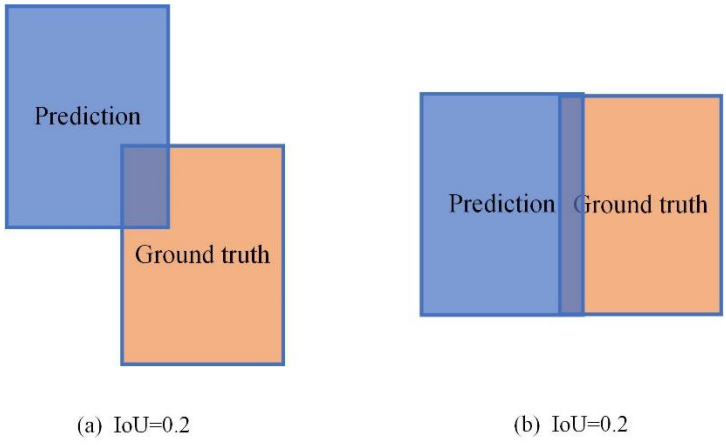


Figure 4 – Two types of overlap: a) Cross overlap; b) Horizontal overlap

Figure 4a and *Figure 4b* exhibit an identical IoU value, despite variations in the overlapping positions between the detection frame and prediction frame in these two scenarios. To address this challenge, Rezatofighi et al. [34] introduced position information into the distance measurement, allowing for a more effective assessment of the intersection between the detection frame and prediction frame more effectively. The generalised intersection over union (GIoU) distance introduces the minimum bounding box of the detection frame and prediction frame to address their spatial position relationship. The calculation process is detailed in *Equation 3*:

$$GIoU = IoU - \frac{|C - B \cup B^{gt}|}{|C|} \tag{3}$$

where C is the minimum frame area used to cover the detection frame B and prediction frame B^{gt}. This, in turn, suggests a lower degree of match between the two, and vice versa. Hence,

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \tag{4}$$

The spatial representation of GIoU is shown in *Figure 5*.

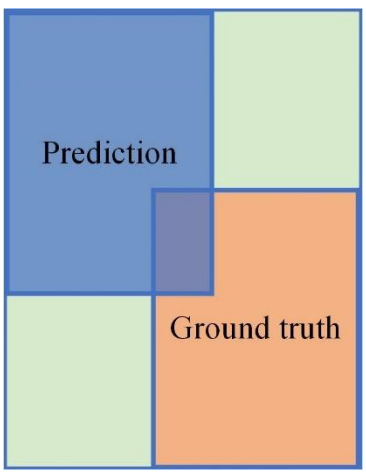


Figure 5 – Spatial representation of GIoU

As demonstrated in *Equation 1*, when the IoU is 0, signifying no overlap between the detection frame and the prediction frame, the value remains constant. Moreover, with a larger value of C, indicating a greater distance between the detection frame and the prediction frame, the GIoU becomes smaller, resulting in a higher GIoU distance. Due to the introduction of the penalty term, the predicted box will move towards the target box in non-overlapping cases.

Although GIoU can relieve the gradient vanishing problem for non-overlapping cases, it still has several limitations, especially when extracting the passenger trajectory in the subway station. In the context of tracking passenger trajectories in a crowded subway station, it is common to encounter situations where the prediction frame falls within the boundaries of the ground truth. From *Figure 6*, GIoU loss will totally degrade to IoU loss for enclosing bounding boxes. In this paper, we use a Distance-IoU (DIoU) loss [36] a penalty term on IoU loss to directly minimise the normalised distance between central points of two bounding boxes, leading to much faster convergence than GIoU loss. Generally, the DIoU-based loss can be defined as:

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} \tag{5}$$

where b and b^{gt} denote the central points of B and B^{gt}, $\rho(\cdot)$ is the Euclidean distance, and c is the diagonal length of the smallest enclosing box covering the two boxes.

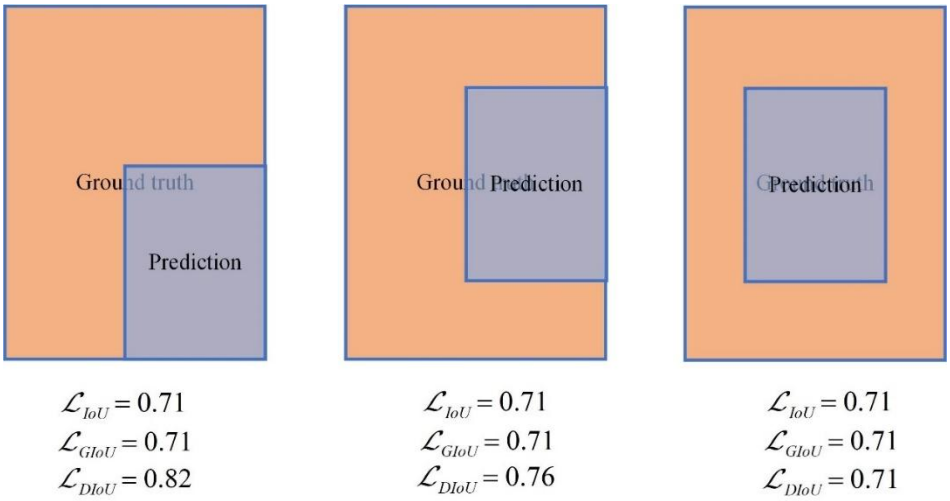


Figure 6 – Different cases of IoU, GIoU and DIoU

To improve the precision of pedestrian extraction in subway settings, a critical adjustment was made by replacing the IoU loss function in the YOLOv5 model with the DIoU loss function. This modification aims to offer a more comprehensive and accurate assessment of the overlap between predicted and ground truth bounding boxes, especially in contexts characterised by environmental noise, obstructions caused by the environment affecting pedestrians and mutual occlusions between pedestrians. The DIoU loss function integrates both overlap metrics and considers the distance between the centroids of the corresponding bounding boxes. This refinement is expected to enhance the model's robustness and accuracy, enabling more effective pedestrian detection and tracking within these challenging subway scenarios. This enhancement is pivotal for optimising performance and establishing a more reliable foundation for subsequent analyses of pedestrian behaviours and safety monitoring initiatives.

3.2 Improved DeepSORT for passenger trajectory tracking

DeepSORT

The SORT algorithm utilises a simple Kalman filter to manage frame-to-frame data correlation and employs the Hungarian algorithm for correlation measurement, demonstrating good performance at high frame rates. Nevertheless, SORT's reliance solely on motion information makes it accurate mainly when target state estimation uncertainty is low. Furthermore, in the pursuit of improved tracking efficiency, SORT removes unmatched targets over continuous frames, leading to the issue of frequent ID switches. DeepSORT addresses these limitations by incorporating appearance information, using a ReID model for feature extraction, and

reducing ID switches by 45%. DeepSORT, as described in [37], is a deep learning-based approach employed in this research for tracking individuals within the surveillance footage. It leverages patterns learned from detected objects in images, which are then combined with temporal information to predict the associated trajectories of the objects of interest. Each object under consideration is tracked using unique identifiers for subsequent statistical analysis. DeepSORT is adept at handling various challenges, including occlusion, multiple viewpoints, non-stationary cameras and annotating training data. To achieve effective tracking, it makes use of the Kalman filter and the Hungarian algorithm. The Kalman filter is applied recursively for improved association and can predict future positions based on the current position. The Hungarian algorithm is used for association and ID attribution to determine if an object in the current frame corresponds to the one in the previous frame. Initially, a faster R-CNN model is trained for person identification, and for tracking, a linear constant velocity model [38] is employed to describe each target within an eight-dimensional space, as follows:

$$\psi = [u, v, \lambda, h, x', y', \lambda', h']^T \tag{6}$$

where (u, v) represents the centroid of the bounding box, while λ is the aspect ratio and h denotes the image height. The remaining variables represent the respective velocities of the parameters. Subsequently, the standard Kalman filter is employed, assuming constant velocity motion and a linear observation model. In this model, the bounding box coordinates (u, v, λ, h) are treated as direct observations of the object state.

For each track k, the system calculates the total number of frames starting from the last successful measurement association a_k. If there is a positive prediction from the Kalman filter, a counter is incremented. When the track becomes associated with a measurement, this counter is reset to zero. Additionally, if the age of the identified tracks surpasses a predefined maximum value, it is assumed that the objects have left the scene, and the corresponding track is removed from the track set. In cases where there are no tracks available for some detected objects, new track hypotheses are initiated for each unidentified track of newly detected objects that cannot be linked to existing tracks. For the first three frames, these new tracks are classified as indefinite until a successful measurement mapping is established. If the tracks cannot be successfully mapped with measurements, they are deleted from the track set. The Hungarian algorithm is then employed to solve the mapping problem between the newly arrived measurements and the predicted Kalman states, considering both motion and appearance information, based on the Mahalanobis distance calculated between them as defined in *Equation 7*.

$$d^{(1)}(i, j) = \left(d_j - y_i\right)^T S_i^{-1} \left(d_j - y_i\right) \tag{7}$$

In this context, the projection of the distribution of the i^{th} track into measurement space is represented as (y_i, S_i), and the j^{th} bounding box detection is represented as d_j. The Mahalanobis distance takes into account the uncertainty by estimating the number of standard deviations that the detection deviates from the mean track location. This decision is denoted with an indicator that evaluates to 1 if the association between the i^{th} track and j^{th} detection is admissible (*Equation 8*).

$$b_{i,j}^{(1)} = 1[d^{(1)}(i, j) < t^{(1)}] \tag{8}$$

Using the motion matching approach based on Mahalanobis distance yields favourable short-term prediction and matching results. However, for long-term trajectory predictions, Mahalanobis distance matching can lead to ID changes. In such cases, introducing the strategy of minimising the cosine distance between metric features can mitigate the issue of losing IDs, making the matching evaluation more reasonable.

The cosine value of the angle between two vectors, denoted as X and Y, $X = [x_1, x_2, \ldots, x_n], Y = [y_1, y_2, \ldots, y_n]$, is known as the cosine distance, as shown below:

$$\cos\theta = \frac{x_1 y_1 + x_2 y_2 + \cdots + x_n y_n}{\sqrt{x_1^2 + x_2^2 + \cdots x_n^2}\sqrt{y_1^2 + y_2^2 + \cdots y_n^2}} \tag{9}$$

This formula describes the cosine distance between two vectors, X and Y, and is used to measure their similarity. A cosine distance closer to 1 indicates that the angle between the two vectors is closer to 0 degrees,

signifying greater similarity. A cosine distance closer to -1 suggests that the angle between the vectors is closer to 180 degrees, indicating dissimilarity. A cosine distance of 0 indicates orthogonality, implying no apparent similarity between the vectors. Utilise the minimum cosine distance to measure the similarity between the appearance features of the previous frame and the current frame. The minimum cosine distance is defined as follows:

$$d^{(2)}(i,j) = \min\left\{1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in \Re_i\right\} \tag{10}$$

where $d^{(2)}(i,j)$ represents the minimum cosine value between the trajectory i and the detection box j, T represents the surface feature information of the detection target box j, and $r^{(i)}$ represents the feature information of the trajectory i.

Once again, a binary variable is introduced to indicate whether an association is permissible based on the following metric:

$$b_{i,j}^{(1)} = 1[d^{(2)}(i,j) < t^{(2)}] \tag{11}$$

This study employs a weighted fusion of the aforementioned two distance metrics to calculate the matching degree of target bounding boxes, and the formula is as follows:

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{12}$$

where λ is a parameter, and its specific value depends on the dataset scenario of the application, adjusting the weight of the matching degree.

Where we call an association admissible if it is within the gating region of both metrics:

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)} \tag{13}$$

The impact of each metric on the overall association cost can be regulated by means of the hyperparameter λ. In DeepSORT, only appearance information is employed in the association cost term. Nonetheless, the Mahalanobis gate is still utilised to discard impractical assignments based on the potential object positions estimated by the Kalman filter.

Confidence-based (CB) Kalman filtering

In DeepSORT, the Kalman filter based on the linear motion hypothesis is used to model objects' motion. It consists of a state estimation step and state update step. In the first step, the Kalman filter produces estimates of current state variables, along with their uncertainties. Then these estimates are updated with a weighted average of the estimated state and the measurement. Specifically, it uses the measurement noise covariance Q_* to represent the measurement (i.e. detections in the current frame) noise scale. A larger noise scale means a smaller weight of the measurement during the state update step, since its larger uncertainty. In the Kalman algorithm [17], the noise scale is a constant matrix. However, intuitively different measurements contain different scales of noise. Especially, in the context of pedestrian trajectory tracking in subway environments, noise tends to be random. In substance, the measurement noise scale should vary with detection confidence. To tackle this issue, we draw inspiration from [39] and [40]. Consequently, we introduce a formula for adaptively calculating the noise covariance $\tilde{Q}_k$:

$$\tilde{Q}_k = (1 - a_k) Q_k \tag{14}$$

where Q_k is the preset constant measurement noise covariance and a_k is the detection confidence score at state k. The whole state update of our CB Kalman filter is shown in *Algorithm 1*, where the CB step is in Step 2. Experimental results show that it significantly improves the tracking performance, though our CB Kalman is simple.

Algorithm 1 – CB Kalman filter (state update step at state k)

Input: Measurement z_k; Measurement confidence a_k;

Predicted state estimate $\hat{y}_{k|k-1}$;

Predicted estimate covariance $P_{k|k-1}$;

The observation model H_k;

The measurement noise covariance Q_k;

1: $\tilde{s}_k = z_k - H_k\hat{y}_{k|k-1}$; (measurement pre-fit residual)

2: $\tilde{Q}_k = (1 - a_k)$; (CB covariance)

3: $R_k = H_k P_{k|k-1} H_k^T + \tilde{Q}_k$; (pre-fit residual covariance)

4: $W_k = P_{k|k-1} H_k^T R_k^{-1}$;(optimal Kalman again)

5: $\hat{y}_{k|k} = \hat{y}_{k|k-1} + W_k \tilde{s}_k$;(update state estimate)

6: $P_{k|k} = (1 - W_k H_k) P_{k|k-1}$;(update estimate covariance)

Output: Update state estimate $\hat{y}_{k|k}$; Update estimate covariance $P_{k|k}$

DeepSORT employs a Kalman filter based on a linear motion hypothesis for modelling object motion. This Kalman filter comprises two key steps: a state estimation step and a state update step. In the initial step, the Kalman filter generates estimations of current state variables, along with their associated uncertainties. Subsequently, these estimations are refined through a weighted combination of the estimated state and the measurement data.

Motion cost

In the initial phase of the DeepSORT tracking algorithm, the primary focus lies on leveraging appearance features to establish object associations. This is achieved by computing the matching cost based on appearance feature distances, allowing for an initial grouping of objects. During this stage, the contribution of motion information serves as a gating mechanism, ensuring that only objects exhibiting a certain degree of motion are considered for association. However, DeepSORT's unique strength becomes apparent in the subsequent stage, where it optimally resolves the assignment problem. Here, the algorithm adopts a more comprehensive approach by incorporating both appearance and motion information. This two-fold strategy aligns with the specific demands of scenarios prevalent in subway passenger flow, where the similarity in pedestrian appearances and frequent occlusions can significantly impact tracking accuracy. In the field of passenger trajectory recognition and extraction within subway stations, the selection of utilising a weighted fusion of the aforementioned distance metrics to evaluate the matching degree of target bounding boxes is a decision grounded in the need for a comprehensive approach. This choice stems from the realisation that the strengths of each distance metric, the Mahalanobis distance for motion matching and the minimum cosine distance for appearance matching, are particularly beneficial in specific contexts. The Mahalanobis distance effectively measures short-term prediction and matching accuracy, ensuring robust performance for immediate trajectory recognition. Conversely, the minimum cosine distance offers enhanced performance for long-term trajectory predictions, addressing the issue of potential identity shifts. By combining these distance metrics in a weighted fusion, our approach benefits from the strengths of both, providing a more holistic solution for passenger trajectory recognition and extraction in subway stations.

By simultaneously considering both appearance and motion data, DeepSORT offers an effective solution to address these challenges, ensuring robust multi-object tracking. This approach is in line with the strategies outlined in references [41] and [42], emphasising the significance of combining appearance and motion data to achieve superior multi-object tracking accuracy in real-world, complex environments. The cost matrix D is a weighted sum of appearance cost C_a and motion cost C_m as follows:

$$D = \lambda C_a + (1 - \lambda) C_m \tag{15}$$

where λ is a weighting parameter. In this manner, the approach effectively addresses DeepSORT's limitation in exclusively relying on appearance feature distances as the sole matching cost during the initial association stage. *Figure 7* illustrates the simultaneous consideration of appearance cost and motion cost during the

matching process. In this early phase, the use of motion distances as a gate is a less comprehensive solution. However, this method significantly enhances the accuracy of the object-tracking process. Adopting this combined approach of considering both appearance and motion information, substantially improves the overall precision of target tracking, mitigating the issues arising from using appearance features alone in the initial association stage.

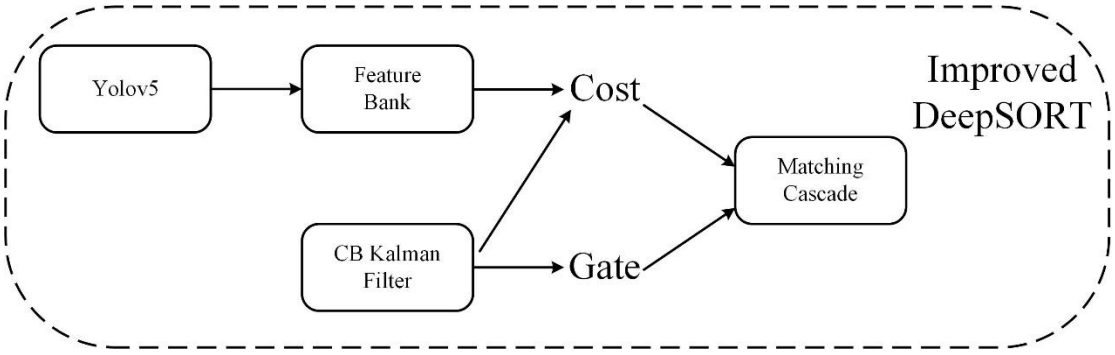


Figure 7 – Considering the appearance cost and motion cost

3.3 Passenger trajectory update algorithm based on momentum

During the process of extracting passenger trajectories using DeepSORT, it is important to consider the impact of random errors. These errors can introduce inaccuracies in the extracted passenger trajectory centres, causing deviations from the actual passenger trajectory centres. As a result, these random errors lead to passenger trajectory jitter or wobbling. This phenomenon can significantly affect the accuracy and stability of passenger trajectory extraction. To mitigate these issues, it is crucial to develop strategies that account for and minimise the effects of these errors, ensuring that the extracted trajectories are as precise and smooth as possible, thereby improving the overall performance of the DeepSORT algorithm in passenger tracking applications.

This paper introduces a novel momentum-based passenger trajectory centre update algorithm, it addresses the challenge of mitigating passenger trajectory jitter caused by random errors. In traditional passenger trajectory extraction methods, the coordinates obtained directly from DeepSORT are often considered as the actual trajectory of the passenger, as shown in the following formula:

$$\tilde{x}_{t+1,i} = \hat{x}_{t+1,i} \tag{16}$$

$$\tilde{y}_{t+1,i} = \hat{y}_{t+1,i} \tag{17}$$

where $\hat{x}_{t,i}$ and $\hat{y}_{t,i}$ are the passenger trajectory i in frame t obtained by the DeepSORT algorithm. This approach often overlooks the errors introduced during the trajectory updating process. Therefore, this paper proposes a momentum-based passenger trajectory update algorithm, which can be represented by the following formula:

$$\tilde{x}_{t+1,i} = \beta \cdot \hat{x}_{t+1,i} + (1-\beta) \cdot \tilde{x}_{t,i} \tag{18}$$

$$\tilde{y}_{t+1,i} = \beta \cdot \hat{y}_{t+1,i} + (1-\beta) \cdot \tilde{y}_{t,i} \tag{19}$$

where β is the momentum update parameter. This innovative approach leverages momentum information from previous observations, enabling a more stable and continuous estimation of passenger trajectory centres. By incorporating this momentum-based update mechanism, we significantly reduce the impact of random errors on the accuracy and smoothness of the extracted passenger trajectories. The momentum-based update algorithm ensures that the estimated passenger trajectory centres are less affected by minor fluctuations in the data. As a result, the trajectories become more consistent and less prone to abrupt deviations, even in the presence of noisy or imprecise input data. This improvement in passenger trajectory stability has a direct positive impact on the performance of the DeepSORT algorithm. It enhances the accuracy and reliability of passenger tracking in complex scenarios, such as subway station environments, where precise trajectory information is vital for various applications, including crowd management, safety protocols and service

optimisation. The momentum-based approach offers a promising solution for enhancing passenger trajectory extraction and, consequently, the overall efficiency of public transportation systems.

The proof of the reduction in passenger trajectory jitter achieved by the momentum-based trajectory update algorithm is presented below. Assuming the actual centre coordinates of the passenger trajectory i in frame t is $(x_{t,i}, y_{t,i})$ and the passenger trajectory centre coordinates obtained using the DeepSORT algorithm are $(\hat{x}_{t,i}, \hat{y}_{t,i})$, with an error in center tracking denoted as $\varepsilon_{t,i}$, where $\varepsilon_{t,i}$ follows a Gaussian distribution (μ, σ^2). Additionally, the passenger trajectory centre obtained using a momentum update algorithm is represented as $(\tilde{x}_{t,i}, \tilde{y}_{t,i})$. Therefore, we can derive the following relationship:

$$\tilde{x}_{0,i} = \hat{x}_{0,i} \tag{20}$$

$$\tilde{x}_{t+1,i} = \beta \cdot \hat{x}_{t+1,i} + (1-\beta) \cdot \tilde{x}_{t,i} \tag{21}$$

The error between the true value and the predicted value is represented as:

$$\varepsilon_{t,i} = \hat{x}_{t,i} - x_{t,i} \tag{22}$$

This error accounts for the difference between the actual passenger trajectory centre coordinates (True Value) and the passenger trajectory centre obtained using the DeepSORT algorithm, including any errors (Predicted Value). Next, we can obtain the following relationship:

$$\tilde{x}_{t+1,i} = \beta \cdot (\varepsilon_{t+1,i} + x_{t+1,i}) + (1-\beta) \cdot \tilde{x}_{t,i} \tag{23}$$

The variance of passenger trajectory fluctuations obtained from the momentum-based trajectory update algorithm is represented as:

$$E[(\tilde{x}_{t+1,i} - E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i}))^2|\tilde{x}_{t,i}] = \beta^2 \cdot \sigma^2 < \sigma^2 \tag{24}$$

Certainly, we will prove the equation step by step. The variance is a measure of how much data points deviate from the mean. In this case, we want to calculate the variance of passenger trajectory fluctuations based on the momentum-based trajectory update algorithm. Passenger trajectory variance (variance of passenger trajectories) can be represented as:

$$E[(\tilde{x}_{t+1,i} - E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i}))^2|\tilde{x}_{t,i}] \tag{25}$$

where $\tilde{x}_{t+1,i} = \beta \cdot \hat{x}_{t+1,i} + (1-\beta) \cdot \tilde{x}_{t,i}$, Then, we calculate the conditional expectation:

$$E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i}) = E(\beta \cdot \hat{x}_{t+1,i} + (1-\beta) \cdot \tilde{x}_{t,i}|\tilde{x}_{t,i}) \tag{26}$$

Since $(1-\beta) \cdot \tilde{x}_{t,i}$ is known constant, it can be taken out of the expectation:

$$E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i}) = \beta \cdot E(\hat{x}_{t+1,i}|\tilde{x}_{t,i}) + (1-\beta) \cdot \tilde{x}_{t,i} \tag{27}$$

Let $E(\hat{x}_{t+1,i}|\tilde{x}_{t,i}) = \hat{x}^e_{t+1,i}$, therefore:

$$E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i}) = \beta \cdot \hat{x}^e_{t+1,i} + (1-\beta) \cdot \tilde{x}_{t,i} \tag{28}$$

Next, we computer $\tilde{x}_{t+1,i} - E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i})$:

$$\tilde{x}_{t+1,i} - E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i}) = \beta \cdot \hat{x}_{t+1,i} + (1-\beta) \cdot \tilde{x}_{t,i} - (\beta \cdot \hat{x}^e_{t+1,i} + (1-\beta) \cdot \tilde{x}_{t,i}) \tag{29}$$

Simplifying, we get:

$$\tilde{x}_{t+1,i} - E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i}) = \beta \cdot (\hat{x}_{t+1,i} - \hat{x}^e_{t+1,i}) \tag{30}$$

Then, we calculate the expectation of its square:

$$E\left[\left(\tilde{x}_{t+1,i} - E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i})\right)^2 \middle| \tilde{x}_{t,i}\right] = \beta^2 \cdot E\left[(\hat{x}_{t+1,i} - \hat{x}^e_{t+1,i})^2 \middle| \tilde{x}_{t,i}\right] \tag{31}$$

Next, we substitute specific values for calculation. Where $E(\hat{x}_{t+1,i}|\tilde{x}_{t,i}) = \hat{x}^{e}_{t+1,i}$ and $Var(\hat{x}_{t+1,i}|\tilde{x}_{t,i}) = \sigma^2$, thus:

$$E\left[\left(\tilde{x}_{t+1,i} - E(\tilde{x}_{t+1,i}|\tilde{x}_{t,i})\right)^2 \middle| \tilde{x}_{t,i}\right] = \beta^2 \cdot E\left[\left(\hat{x}_{t+1,i} - \hat{x}^{e}_{t+1,i}\right)^2 \middle| \tilde{x}_{t,i}\right] = \beta^2 \cdot \sigma^2 \tag{32}$$

Since $\beta \in (0,1)$, it follows that $\beta^2 \cdot \sigma^2 < \sigma^2$. Therefore, *Equation 24* is proofed. Similarly, the update process for $\tilde{y}_{t,i}$ mirrors that of $\tilde{x}_{t,i}$. So, the momentum-based passenger trajectory update algorithm can reduce trajectory jitter.

3.4 Passenger detection and tracking

In this paper, the proposed intelligent passenger recognition and tracking system is designed to detect passengers and subsequently extract their trajectories. The detailed algorithmic workflow is illustrated in *Figure 8*.

— Step 1: The video stream is fed into the YOLOv5 detector, resulting in the identification of passengers. Then, the system computes the central coordinates, aspect ratio, height and their corresponding speeds in image coordinates for the detection boxes of each passenger.
— Step 2: The Kalman filter uses the acquired central coordinates, aspect ratio, height and corresponding speeds as direct object observations, subsequently calculating the predicted target position. The current frame's detection results are compared with the CB Kalman filter's predictions. Upon a successful match, the CB Kalman filter updates the tracking process and proceeds to target tracking in the next frame.
— Step 3: Instances where a track lacks matching detection results can occur when detections are missed, and situations where detection results do not match with any existing track may arise when a new target enters the scene. Both scenarios result in a failed match. To address this, the DIoU is calculated to facilitate a secondary match between the predicted and undetected boxes. Following a successful match, the Kalman filter updates the new track.
— Step 4: A new track is established for detection boxes that repeatedly fail to match, labelled as unconfirmed tracks. When an unconfirmed track successfully matches three times, it is promoted to a confirmed track, and steps 2 and 3 are repeated. The state of a prediction box that repeatedly fails to match is assessed to determine whether the track should be retained or deleted. If the track is marked as unconfirmed, it will be deleted. If the track is designated as confirmed but fails to match within its lifespan, it will also be deleted. Otherwise, the track is maintained, and steps 1–3 are repeated.
— Step 5: When the target is successfully tracked in multiple consecutive frames, a continuous trajectory is formed, representing the pedestrian's movement path. By analysing the generated trajectory data, insights into the pedestrian's movement patterns, congested areas and other relevant information can be gained. With the aid of visualisation tools, pedestrian trajectories can be presented graphically, assisting subway operators in gaining a better understanding of passenger flow dynamics.

In conclusion, the above five steps constitute the key procedures of the proposed model. The model's computational complexity primarily stems from the YOLOv5 object detection component and the M-DeepSORT object tracking component. To further optimise the system's performance and address computational considerations, we have made several enhancements to balance accuracy and efficiency. Firstly, the YOLOv5 algorithm is slightly modified to improve accuracy while maintaining low model complexity. These modifications involve fine-tuning the computational framework and adjusting associated processing rules, thereby enhancing detection performance while maintaining low computational overhead. Secondly, in the tracking component, the DeepSORT algorithm is improved by enhancing the Kalman filtering process. Specifically, a CB Kalman algorithm is designed to provide more accurate state predictions and updates, leading to more reliable tracking performance with minimal computational overhead. Finally, to ensure that the generated trajectories closely reflect the actual movement patterns, a momentum update strategy is introduced. This strategy refines the trajectories by incorporating velocity and acceleration information, resulting in more realistic movement paths.

By implementing these improvements, the method achieves a high level of accuracy in detecting and tracking pedestrians while ensuring real-time performance, which is crucial for deployment in resource-constrained environments such as subway stations. This balance between accuracy and computational efficiency is essential for the practical application of the system in enhancing subway operations.

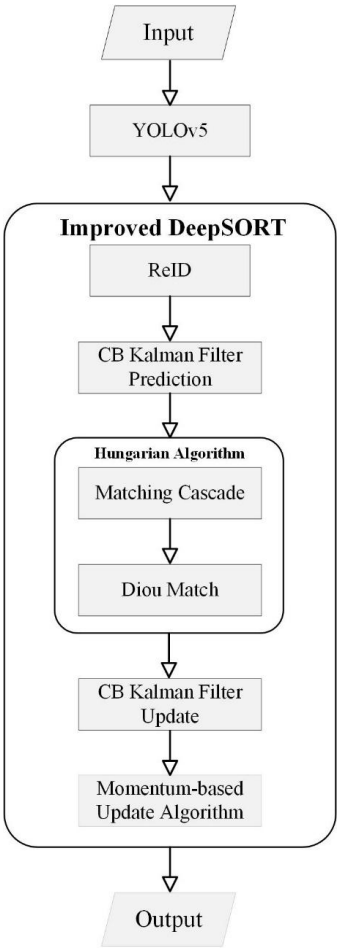


Figure 8 – The algorithm flow of detection and tracking

4. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we will first introduce, in Section 4.1, commonly used metrics for assessing the effectiveness of the model, providing a basis for a comprehensive evaluation of its performance. Subsequently, in Section 4.2, we validate the model using the multiple object tracking 2016 (MOT16) dataset to examine its robustness and accuracy under standardised testing conditions. Finally, in Section 4.3, we will perform practical application validation to assess the model's applicability and effectiveness in real-world scenarios.

4.1 Metrics

The output of the MOT task provides a representation that encompasses several key aspects, including: (a) identification of the objects present in each frame (detection). (b) Localisation of these objects, indicating their positions in each frame (localisation). (c) Determination of whether objects in different frames are related, indicating whether they belong to the same object or different objects (association). In the context of tracking passenger trajectories within a subway station, the process encompasses several crucial stages. Initially, the system is tasked with the recognition of individual passengers within the video footage, a task that involves identifying the presence of passengers and determining their initial positions. Subsequently, the localisation step aims to precisely pinpoint the locations of these passengers in each video frame, thereby tracking their movements over time. To establish continuous passenger trajectories, the association step comes into play, where the system must determine whether passengers detected in different video frames correspond to the same individuals. Ultimately, through these coordinated efforts, passenger trajectories are generated, depicting the paths traversed by individual passengers as they navigate through the subway station.

In the evaluation of multi-object tracking accuracy (MOTA) [43], the matching process operates at the level of individual detections. It establishes a one-to-one mapping between predicted detections (prDets) and ground-truth detections (gtDets) within each frame. When a prDet and a gtDet are successfully matched, they are considered true positives (TPs). Any prDets that remain unmatched are labelled as false positives (FPs),

representing additional predictions, while any gtDets without corresponding matches are categorised as false negatives (FNs), representing missing predictions. For a successful match to occur, prDets and gtDets must exhibit sufficient spatial similarity, necessitating the introduction of a similarity score, denoted as S (e.g. IoULoc for 2D bounding boxes). Furthermore, a similarity threshold, α, is defined, ensuring that matches are only established when $S \geq \alpha$. It is important to note that multiple matching scenarios may arise, and the final MOTA and multiple object tracking precision (MOTP) scores are computed in a manner that optimises their accuracy, as elaborated below. The evaluation metrics used in the experiment are as follows:

MOTA: MOTA quantifies three distinct categories of tracking errors, which encompass detection-related errors including false negatives (FNs) and false positives (FPs), along with the association error denoted as IDS (ID switches). To calculate the ultimate MOTA score, these errors are tallied, the sum is divided by the total number of ground-truth detections (gtDets), and the result is then subtracted from one.

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t gtDet_t} \tag{33}$$

where the evaluation index false positive (FP) is the number of false detections, false negative (FN) is the number of missed detections, identity switch (IDS) indicates the number of identity exchanges of all tracking targets. t is the frame index and gtDet is the number of ground truth objects. MOTP assesses the precision of the detector's spatial localisation and is defined by the following equation:

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{34}$$

where c_t represents the total number of matches at frame t, and $d_{t,i}$ represents the distance between the hypothetical bounding box and the real bounding box.

IDF1 [44]: ID F1 Score. The ratio of correctly identified detections over the average number of ground-truth and computed detections.

ML: Mostly Lost Targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.

MT: Mostly Tracked Targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span.

FN: The total number of false negatives (missed targets).

4.2 Comparison of different methods

To validate the effectiveness of the proposed method, we initially conducted model validation tests using the MOT16 dataset. This dataset serves as a valuable benchmark for evaluating tracking algorithms. MOT16 [45] dataset is a widely recognised benchmark in the field of multi-object tracking. As part of the MOT challenge, MOT16 provides a standardised dataset with real-world video sequences that encompass various tracking challenges. These sequences include scenarios such as pedestrian tracking in crowded environments and challenging lighting conditions. Ground-truth annotations are available for each sequence, offering precise object identities and bounding box coordinates for evaluating tracking algorithms in terms of accuracy, identity preservation and spatial localisation. To evaluate the model's effectiveness, we conducted testing and validation using the MOT16-11 dataset, which is chosen due to its similarity to scenarios encountered in subway station environments.

Figure 9 – Pedestrian recognition partial dataset of MOT 16-11

Next, we compared the performance of this dataset in the DeepSORT model and the model proposed in this paper. As input to DeepSORT we rely on detections provided by Yu et al. [46]. They have trained a faster RCNN on a collection of public and private datasets to provide excellent performance. The results are presented in the following *Table 1*.

Table 1 – Tracking results of different methods

Model	MOTA	MOTP	IDF1	FN	ML	MT	Recall
DeepSORT	61.6%	66.185	74.6%	3841	35%	17%	61.9%
The proposed model	76.1%	273.813	79.2%	2080	14%	39%	79.4%

Table 1 illustrates a comparative analysis of results obtained for the same dataset using both the DeepSORT and the proposed YOLOv5+M-DeepSORT methods. The quantitative analysis of the results reveals a significant performance disparity between the DeepSORT and the proposed YOLOv5+M-DeepSORT methods in the context of subway passenger trajectory tracking. The proposed model achieved a notably higher MOTA at 76.1%, showcasing a substantial improvement over DeepSORT's 61.6%. This enhancement is particularly relevant in real-world subway scenarios, where accurate tracking is crucial for ensuring passenger safety and optimising transit operations. In terms of MOTP, the proposed model demonstrated superior performance with a value of 273.813, representing a substantial increase compared to DeepSORT's 66.185. This heightened precision is instrumental in accurately capturing and predicting passenger movements within the subway environment. Furthermore, the proposed model exhibited a superior IDF1 score of 79.2%, indicating an improved ability to precisely delineate object boundaries. This is particularly valuable for subway security applications, where precise detection contributes to efficient surveillance and threat identification. FN, the proposed model exhibited a decrease to 2080 from DeepSORT's 3841, indicating a significant reduction in missed detections. This highlights the improved sensitivity of the proposed model in correctly identifying and tracking objects. This elevated recall rate translates to a more comprehensive tracking of subway passengers, enhancing the system's overall reliability. Nevertheless, it is crucial to highlight that the proposed model exhibited a relatively lower ML at 14%, in contrast to DeepSORT's higher value of 35%. A lower ML indicates that a smaller proportion of the tracked ground truth trajectories have been tracked for at most 20% of their lifespan. This reflects the higher efficiency of the tracking algorithm and a more stable tracking of targets. The recall metric, which measures the model's ability to correctly identify and track objects, also showcased improvement with the proposed model achieving 79.4%, compared to DeepSORT's 61.6%. The proposed model exhibits a relative improvement of 29% in terms of recall compared to the DeepSORT model. This elevated recall rate translates to a more comprehensive tracking of subway passengers, enhancing the system's overall reliability. In summary, the quantitative results underscore the real-world significance of applying the YOLOv5+M-DeepSORT model to subway passenger trajectory tracking, showcasing substantial improvements in tracking accuracy, precision and object detection metrics.

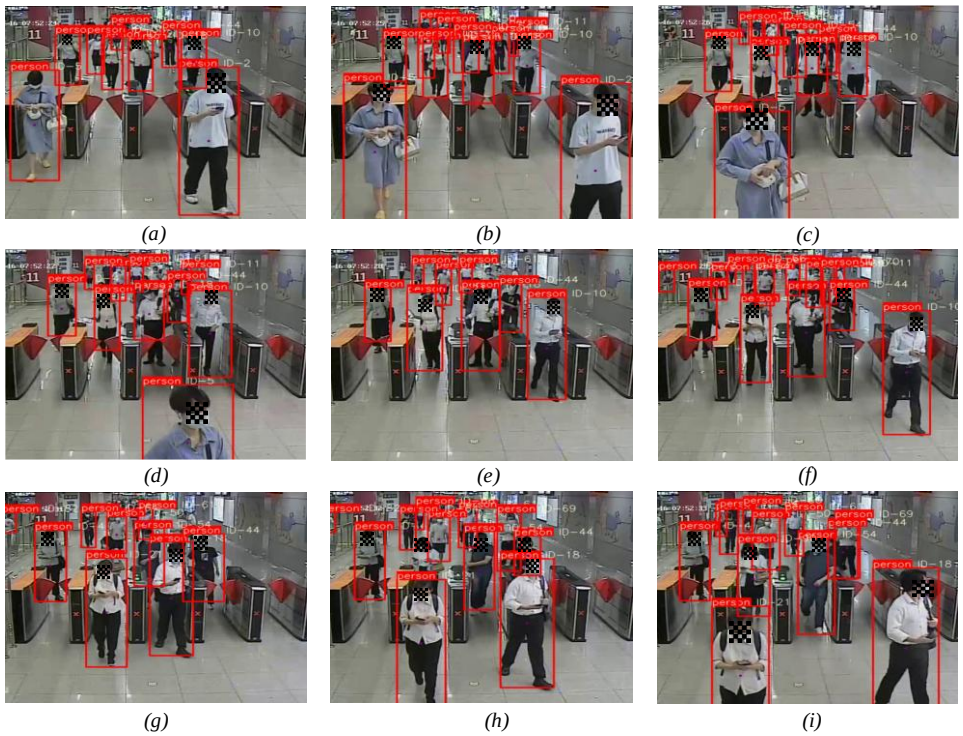
4.3 Experimental results and analysis of subway passenger tracking scenario

To evaluate the effectiveness of the passenger detection and tracking model presented in this paper, we conducted tests using surveillance video data recorded in the context of Jinan subway scenarios. Specifically, we examined images under various challenging conditions, including target occlusion, densely packed and small targets scenes. The corresponding test results are depicted in *Figure 10*. *Figure 10a* demonstrates the model's effectiveness in identifying passengers even when they are partially obscured by buildings or other objects. Notably, *Figure 10a* and *Figure 10b* demonstrate that, following a brief occlusion period, the tracking IDs of individuals with IDs 3 remain unchanged. *Figure 10c* showcases the model's performance in scenarios with densely packed targets, revealing its ability to distinguish and independently detect each target within a clustered group. Even in situations where passengers are mutually obscured in a crowd or partially hidden by buildings, our proposed model demonstrates precise identification of each individual. Additionally, *Figure 10a and Figure 10b* demonstrate the model's effective detection of small targets. Even if the ID 11 target becomes larger, it can still be accurately detected and identified, and the ID remains unchanged. Even in scenarios with different passenger densities and partial occlusion, as depicted in *Figure 10a to Figure 10d,* the proposed model maintains accurate detection without repeated switches in target IDs. *Figure 11a to Figure 11i* shows the passenger

detection results of a time-series video data, with one frame extracted per second. These figures illustrate that the proposed algorithm can accurately identify passengers in scenarios with a high number of passengers and partial occlusions. The comprehensive test results affirm the robustness and versatility of the proposed passenger detection and tracking algorithm across challenging subway scenarios.

(a) *(b)* *(c)* *(d)*

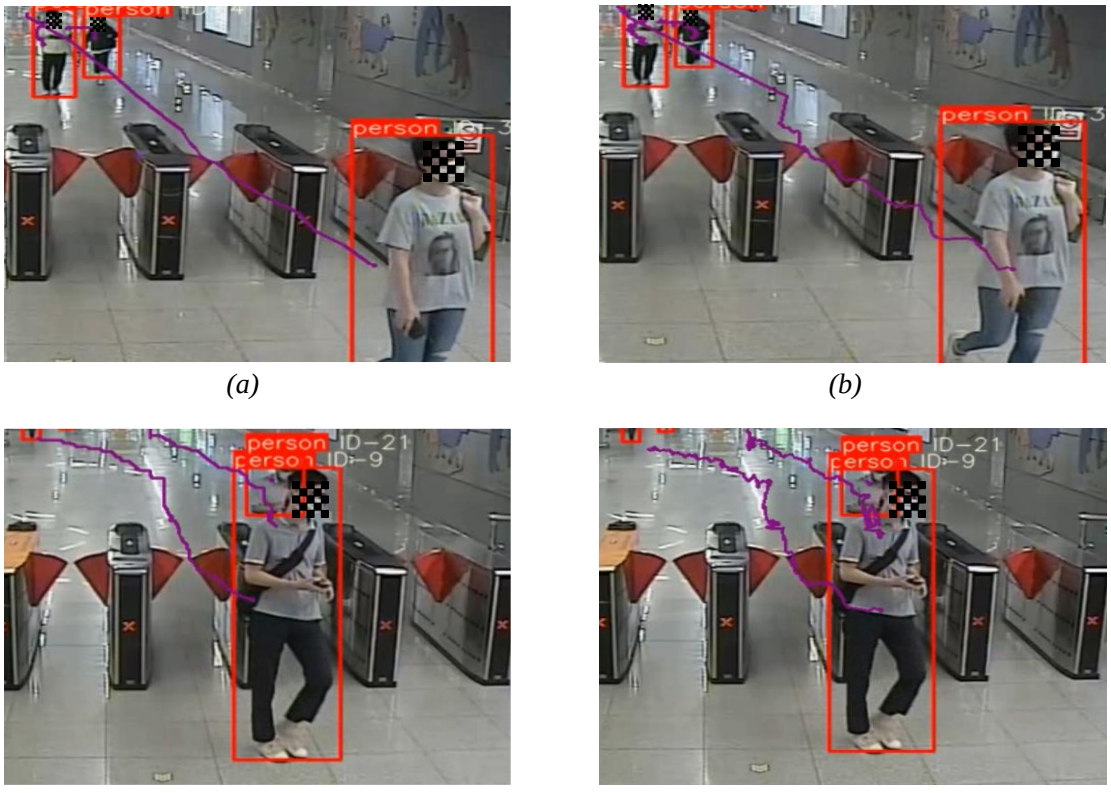
Figure 10 – The test results of subway station passenger detection: a) Detection results of passengers occluded by buildings; b) Passenger detection results without building occlusion; c) Detection results of mutual occlusion between passengers; d) Detection results after the target becomes larger



(a) *(b)* *(c)* *(d)* *(e)* *(f)* *(g)* *(h)* *(i)*

Figure 11 – The detection result with high number of passengers: a) Initial frame; b) Second frame; c) Third frame; d) Fourth frame; e) Fifth frame; f) Sixth frame; g) Seventh frame; h) Eighth frame; i) Ninth frame

Moreover, to further validate the effectiveness of our proposed method, we conducted passenger trajectory tracking and presented visualisations, as depicted in *Figure 12*. *Figure 12a and Figure 12b* present the results in sparsely populated passenger scenarios, showcasing the outcomes without and with the utilisation of the momentum-based passenger trajectory updating algorithm, respectively. Similarly, *Figure 12c and Figure 12d* illustrate the outcomes in scenarios with multiple passengers appearing simultaneously, contrasting the results obtained without employing the momentum-based passenger trajectory updating algorithm and with its application, respectively. In order to quantitatively assess the superiority of the algorithm proposed in this paper, we used *Figure 12c and Figure 12d* as examples and compared the results of employing the momentum-based passenger trajectory updating algorithm with those obtained without its application in detecting passenger scenarios. The variance of passenger trajectory coordinates was computed for both cases. The research findings indicate that the utilisation of the momentum-based passenger trajectory updating algorithm significantly reduces trajectory jitter, decreasing the amplitude of jitter by 15.87%. This, in turn, more accurately reproduces the actual walking trajectories of passengers. The figure reveals that, whether in sparsely populated or densely crowded passenger scenarios, the momentum-based passenger trajectory updating method accurately identifies each passenger, extracts their trajectories, mitigates trajectory data fluctuations, reduces trajectory data errors, and enhances the accuracy of extracted trajectories.



(a) *(b)*

(c) *(d)*

Figure 12 – Passenger trajectory tracking with different methods: a) Single passenger trajectory result with the momentum-based algorithm; b) Single passenger trajectory result without the momentum-based algorithm; c) Multiple passengers trajectory result with the momentum-based algorithm; d) Multiple passengers trajectory result without the momentum-based algorithm

Experimental results demonstrate the significant advantages of the proposed method in subway station scenarios. It not only improves trajectory accuracy but also increases reliability, adding practical value to the analysis of passenger behaviours within subway stations. This strategy not only provides additional data support for the operational management of subway stations but also establishes a more reliable foundation for future decision-making processes, such as intelligent guidance services and safety evacuation plans.

5. CONCLUSION

This paper introduces an intelligent system designed for pedestrian trajectory recognition and extraction within subway stations, employing YOLOv5 as the detector integrated with M-DeepSORT for tracking. The

dataset used for pedestrian trajectory recognition is curated from diverse subway scenarios, encompassing factors such as varying pedestrian densities, occlusions and different proximity situations. These scenarios contribute to the system's robustness and applicability. The experimental results underscore the algorithm's efficacy across diverse scenarios, exhibiting reliable statistical accuracy. The proposed method demonstrates consistent performance, effectively recognising and tracking pedestrians in subway station footage irrespective of factors such as distance, lighting conditions and the presence of occlusions. The method proposed in this paper outperforms the DeepSORT approach in passenger trajectory tracking, exhibiting a notable improvement of 23.5% in MOTA. This method achieves higher levels of accuracy and precision in comparison, establishing its superiority in tracking passenger movements within the subway station environment. In addition, the momentum-based passenger trajectory updating method proposed in this paper addresses the challenges encountered by traditional trajectory extraction methods when handling passenger behaviours, mitigating the impact of noise or instability that often leads to trajectory jitter.

In conclusion, the proposed model offers a method for precise recognition and extraction of passenger trajectories in crowded subway station environments. It addresses key challenges in multi-object tracking and enhances real-world traffic surveillance and management. Moreover, passenger trajectory analysis has the capability to predict crowd congestion, guide and redirect passengers and respond to emergencies. By accurately tracking passenger trajectories, our method not only provides real-time data on pedestrian traffic volumes, calculates space occupancy and estimates queue lengths but also aids in optimising the layout of facilities and equipment within subway stations. This information is crucial for optimising traffic flow and improving crowd management. Detailed trajectory analysis helps optimise flow lines, reducing bottlenecks and improving overall movement efficiency, resulting in a smoother and more orderly flow of people. Insights into passenger movement trajectories in congested areas during peak hours or disruptions enable the development of strategies to redirect flow and alleviate congestion, thereby enhancing safety and convenience. Although the model is compact, it is not yet optimized for embedded deployment. Future work will focus on further improving DeepSORT, as well as compressing and pruning YOLOv5, to enable deployment on edge devices while maintaining a balance between accuracy and processing speed.

ACKNOWLEDGEMENT

This work is supported by the Fundamental Research Funds for the Central Universities (2024YJS109), Key Technology Project of Transportation Industry (2022-MS3-088) and Science and Technology Project of Shandong Provincial Department of Transportation (2022B09-02). We would like to express our gratitude to Ji-nan Metro Company for providing data support for this work.

DECLARATIONS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

[1] Chuang Z, et al. Designing boarding limit strategy by considering stop-level fairness amid the COVID-19 outbreak. *Transportmetrica A: Transport Science.* 2023;1-30. DOI: 10.1080/23249935.2023.2167500.

[2] Chuang Z, et al. Joint optimization of bus scheduling and seat allocation for reservation-based travel. *Transportation Research Part C: Emerging Technologies.* 2024;163:104631. DOI: 10.1016/j.trc.2024.104631.

[3] Qiaochu C, et al. Simulation and optimization of pedestrian regular evacuation in comprehensive rail transit hub - a case study in Beijing. *Promet – Traffic&Transportation.* 2020;32(3):383-397. DOI:10.7307/ptt.v32i3.3318.

[4] Wei L, et al. Experimental study for optimizing pedestrian flows at bottlenecks of subway stations. *Promet – Traffic&Transportation*. 2018;30(5):525-538. DOI 10.7307/ptt.v30i5.2715.

[5] Navneet D, Bill T. Histograms of oriented gradients for human detection. *IEEE computer society conference on computer vision and pattern recognition 2005*, 20-26 Jun 2005, San Diego, CA, USA. 2005. p. 886-893. DOI: 10.1109/CVPR.2005.177.

[6] Stefan W, et al. New features and insights for pedestrian detection. *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 13-18 June 2010, San Francisco, CA, USA, 2010, pp. 1030-1037, DOI: 10.1109/CVPR.2010.5540102.

[7] Rodrigo B, et al. Ten years of pedestrian detection, what have we learned? *Computer Vision - ECCV 2014 Workshops.* ECCV 2014. Lecture Notes in Computer Science, 8926. Springer, Cham. DOI: 10.1007/978-3-319-16181-5_47.

[8] Markus E, Dariu M. Monocular pedestrian detection: Survey and experiments. *IEEE transactions on pattern analysis and machine intelligence*, 2008; 31(12):2179-2195. DOI: 10.1109/TPAMI.2008.260.

[9] Piotr D, et al. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 2011;34(4):743-761. DOI: 10.1109/TPAMI.2011.155.

[10] Paul V, Michael J. Robust real-time face detection. *International journal of computer vision*, 2004;57:137-154. DOI: 10.1023/B:VISI.0000013087.49260.fb

[11] David G. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 1150-1157. DOI: 10.1109/ICCV.1999.790410.

[12] Ross G, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition,* Columbus, OH, USA, 2014, p. 580-587, DOI: 10.1109/CVPR.2014.81.

[13] Shaoqing R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* p. 1137-1149, 1 June 2017, DOI: 10.1109/TPAMI.2016.2577031.

[14] Shanshan Z, et al. How far are we from solving pedestrian detection? , *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, p. 1259-1267, DOI: 10.1109/CVPR.2016.141.

[15] Joseph R, et al. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, p. 779-788, DOI: 10.1109/CVPR.2016.91.

[16] Nicolai W, et al. Simple online and realtime tracking with a deep association metric, *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, p. 3645-3649, DOI: 10.1109/ICIP.2017.8296962.

[17] Alex B, et al. Simple online and realtime tracking, *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016, p. 3464-3468. DOI: 10.1109/ICIP.2016.7533003.

[18] Huajun S, et al. Detection and tracking of safety helmet based on DeepSort and YOLOv5. *Multimed Tools Appl*, 2023;82(7):10781-10794. DOI: 10.1007/s11042-022-13305-0.

[19] Jiandong Z, et al. Detection of passenger flow on and off buses based on video images and YOLO algorithm. *Multimed Tools Appl*, 2022;81(4):4669-4692. DOI: 10.1007/s11042-021-10747-w.

[20] Jiandong Z, et al. Detection of crowdedness in bus compartments based on ResNet algorithm and video images. *Multimed Tools Appl*, 2022;81(4):4753-4780. DOI: 10.1007/s11042-021-11008-6.

[21] Vladimir M, et al. Pedestrian detection in video surveillance using fully convolutional YOLO neural network, *Automated Visual Inspection and Machine Vision II*, Munich, Germany, 2017;103340Q (2017). DOI: 10.1117/12.2270326.

[22] Yongxin W, et al. Joint object detection and multi-object tracking with graph neural networks, *2021 IEEE international conference on robotics and automation (ICRA)*, Xi'an, China, 2021, p. 13708-13715. DOI: 10.1109/ICRA48506.2021.9561110

[23] Hiroshi F, et al. Multi-object tracking as attention mechanism. *2023 IEEE International Conference on Image Processing (ICIP)*, Kuala Lumpur, Malaysia, 2023, p. 1761-1765. DOI: 10.1109/ICIP49359.2023.10222207

[24] Stadler, D., and Beyerer, J. Improving multiple pedestrian tracking by track management and occlusion handling. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, p. 10958-10967. DOI: 10.1109/CVPR46437.2021.01081.

[25] Yifu Z, et al. ByteTrack: Multi-object tracking by associating every detection box. *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, Tel Aviv, Israel, 2022, p. 1-21. DOI:10.1007/978-3-031-20047-2_1.

[26] Jiaxin L, et al. SimpleTrack: Rethinking and Improving the JDE Approach for Multi-Object Tracking. *Sensors*, 2022;22(15):5863. DOI: 10.3390/s22155863.

[27] Yaoyao S, Yi Z. Multi-object tracking with integrated heads and attention mechanism. *Neurocomputing*, 2022;510:95-106. DOI: 10.1016/j.neucom.2022.09.045.

[28] Xiaolong Z, et al. Multi-object tracking based on attention networks for smart city system. *Sustainable Energy Technologies and Assessments*, 2022;52. DOI: 10.1016/j.seta.2022.102216.

[29] Zhihong S, et al. A Survey of multiple pedestrian tracking based on tracking-by-detection framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020;31(5):1819-1833. DOI: 10.1109/TCSVT.2020.3009717.

[30] Xin X, Xinlong F. Multi-object pedestrian tracking using improved YOLOv8 and OC-SORT. *Sensors*, 2023;23(20):8439. DOI: 10.3390/s23208439.

[31] Glenn J, et al. Zenodo, 2020. https://github.com/ultralytics/yolov5 [Accessed 12th May 2023].

[32] Seung-Hwan B. Object detection based on region decomposition and assembly. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8094-8101. DOI: 10.1609/aaai.v33i01.3301809.

[33] Kaiming H, et al. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, p. 2980-2988, DOI: 10.1109/ICCV.2017.322.

[34] Hamid R, et al. Generalized intersection over union: A metric and a loss for bounding box regression, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, p. 658-666, DOI: 10.1109/CVPR.2019.00075.

[35] Jiahui Y, et al. Unitbox: An advanced object detection network. *Proceedings of the 24th ACM international conference on Multimedia*, Oct, 2016, p. 516–520. DOI: 10.1145/2964284.2967274.

[36] Zhaohui Z, et al. Distance-IoU loss: Faster and better learning for bounding box regression, *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020; 34, 07, 12993-13000. DOI: 10.1609/aaai.v34i07.6999.

[37] Feng Y, et al. Video object tracking based on YOLOv7 and DeepSORT. arXiv preprint arXiv:2207.12202, 2022

[38] Nicolai W, Alex B. Deep cosine metric learning for person re-identification. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA, 2018, p. 748-756. DOI: 10.1109/WACV.2018.00087.

[39] Yunhao D, et al. Giaotracker: A comprehensive framework for Mcmot with global information and optimizing strategies in visdrone 2021, *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada, 2021, p. 2809-2819. DOI: 10.1109/ICCVW54120.2021.00315.

[40] Yunhao D, et al. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023;25:8725-8737. DOI: 10.1109/TMM.2023.3240881.

[41] Yifu Z, et al. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 2021;129:3069-3087. DOI: 10.1007/s11263-021-01513-4.

[42] Zhongdao W, et al. Towards real-time multi-object tracking, *Computer Vision – ECCV 2020 Lecture Notes in Computer Science*, Springer, Cham. DOI: 10.1007/978-3-030-58621-8_7.

[43] Keni B, Rainer S. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008;1-10. DOI: 10.1155/2008/246309.

[44] Ergys R, et al. Performance measures and a data set for multi-target, multi-camera tracking. *Computer Vision – ECCV 2016 Workshops Lecture Notes in Computer Science*, 2016;9914. Springer, Cham. DOI: 10.1007/978-3-319-48881-3_2.

[45] Anton M, et al. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. https://arxiv.org/abs/1603.00831 [Accessed 23th May 2023].

[46] Fengwei Yu, et al. Poi: Multiple object tracking with high performance detection and appearance feature. *Computer Vision–ECCV 2016 Workshops*, Amsterdam, The Netherlands, 2016, p. 36-42. DOI: 10.1007/978-3-319-48881-3_3.

张伟，朱闯，屈云超，刘冠华，Der-Horng LEE

基于视频图像的 M-DeepSORT 行人检测与跟踪算法：以济南地铁站为例

摘要：
随着城市化进程的推进，地铁已成为现代城市的重要组成部分，满足了日益增长的乘客出行需求。然而，地铁站内客流的日益复杂化对运营管理提出了挑战。为优化地铁运营并提升安全性，研究人员将目光聚焦于地铁站内行人轨迹的提取与分析。传统的轨迹提取方法由于依赖手工特征设计和多阶段处理，存在一定的局限性。本文结合深度学习的最新进展，将 M-DeepSORT 与 YOLOv5 进行集成，提出了一种特征关联匹配方法，通过同时考虑运动与外观匹配来解决轨迹漂移问题。针对地铁场

景中行人检测的随机噪声问题，本文提出了一种基于置信度（CB）的卡尔曼滤波方法。此外，本文引入了一种基于动量的乘客轨迹中心更新方法，有效减小了轨迹抖动，可以提取到更平滑的乘客轨迹。实验结果验证了所提算法在检测、跟踪和统计分析地铁站走廊客流轨迹中的有效性，并展示了其在不同地铁站场景中的鲁棒性能。

关键词：
乘客轨迹跟踪；CB 卡尔曼滤波；轨迹更新；动量；M-DeepSORT