Intelligent Transportation and Smart Cities
V. Mezhuyev et al. (Eds.)

doi:10.3233/ATDE250256

Design of Pedestrian Tracking System Combining YOLOv8-Pose and Bot-Sort Algorithm in Autonomous Driving Environment

Shanglin LI[a] and Gang LI[a,b,c1]
[a] *School of Mechanical and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, Gansu, China*
[b] *Gansu Logistics and Transportation Equipment Information Engineering Technology Research Center, Lanzhou 730070, Gansu, China*
[c] *Gansu Logistics and Transportation Equipment Industry Technology Center, Lanzhou 730070, Gansu, China*

Abstract: With the development of autonomous driving technology, there are still many challenges to accurately tracking pedestrians in complex environments, such as occlusion, dense crowds, and light variations. This study aims to design a highly adaptive pedestrian tracking system to improve the reliability of the autonomous driving perception system. In this paper, an innovative solution combining YOLOv8-pose and improved Bot-sort algorithm is proposed to integrate the detection bounding box, appearance features, and 17-point pose information through a multimodal feature fusion strategy, and the matching cost matrix is redesigned to enhance the tracking performance. Experimental results show that the proposed pose feature enhancement strategy significantly improves the system's capability in similar appearance pedestrian differentiation and trajectory continuity, and it is well adapted to scenarios such as occlusion, dense crowds, and lighting changes. Meanwhile, the system maintains the real-time processing performance and provides reliable support for the automatic driving perception system, demonstrating the potential and value of multimodal feature fusion for pedestrian tracking in complex environments.

Keywords: Pedestrian tracking; YOLOv8-pose; Bot-sort; Pose features; Multimodal fusion; Autonomous driving

1. Introduction

Pedestrian tracking, as a key task in autonomous driving environment sensing, directly affects vehicle safety decision-making and behaviour planning [1]. Existing research mainly focuses on appearance features and motion information.DeepSORT uses deep learning network to extract pedestrian appearance features, and achieves target re-identification through cascade matching and Mahalanobis distance metric [2]. ByteTrack solves the problem of occlusion through the low-threshold detection compensation mechanism, and introduces BYTE correlation algorithm to deal with the

[1] Corresponding Author: Gang Li, ligang88@mail.lzjtu.cn

low confidence detection frames, which effectively improves the MOT17 dataset with the the MOTA metric to 80.3% [3]. Bot-SORT fuses appearance and motion features through Kalman filtering and similarity computation to improve the MOTA to 77.8% on MOT20 [4]. However, these methods still have obvious shortcomings in similar appearance pedestrian differentiation, prolonged occlusion and complex lighting environments. Relevant studies have shown that when pedestrians are densely or partially occluded, it is difficult to maintain stable tracking by relying only on appearance features, and the tracking success rate can drop by more than 30% [5]. In this paper, we propose a pedestrian tracking system that integrates YOLOv8-pose and improved Bot-sort, innovatively introduces pose features into the multi-target tracking matching strategy, designs a multi-dimensional feature cost matrix and a scene adaptive mechanism, and optimises for the complex scene of autonomous driving. The system achieves high-precision pedestrian detection, stable pose estimation and continuous trajectory tracking, significantly improves the occlusion processing and similar target differentiation ability, and provides more reliable pedestrian behaviour perception support for autonomous driving systems.

2. Overall System Design

2.1 System Architecture Design

The system adopts a layered architecture, which consists of a perception layer, a processing layer and an application layer, as shown in Fig. 1. The perception layer collects the real-time road scene through the on-board camera, which supports 1920×1080 resolution and 30fps video stream input. The processing layer is the core of the system, which contains three functional units: pre-processing module, YOLOv8-pose detection and attitude estimation module, and Bot-sort tracking module. The pre-processing module is responsible for image enhancement and scaling to optimise the input image to 640×640 size; the YOLOv8-pose module performs pedestrian target detection and extracts 17 key points of pose information; and the Bot-sort module integrates the target frames and pose data to achieve multi-target tracking [6]. The application layer visualises the processing results and provides decision support, and the overall latency of the system is controlled within 50ms to meet the real-time requirements of autonomous driving scenarios.The system enables efficient pedestrian tracking for improved safety in autonomous driving.

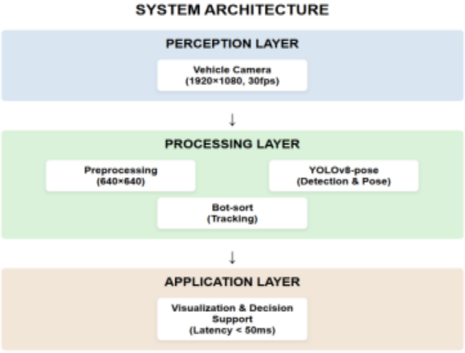
SYSTEM ARCHITECTURE
PERCEPTION LAYER
Vehicle Camera (1920×1080, 30fps)
PROCESSING LAYER
Preprocessing (640×640)
YOLOv8-pose (Detection & Pose)
Bot-sort (Tracking)
APPLICATION LAYER
Visualization & Decision Support (Latency < 50ms)

Figure 1. System architecture

2.2 Data Flow Design

The data flow of the system follows the closed loop of “Acquisition-Processing-Tracking-Output”. After pre-processing the raw image data, the YOLOv8-pose detection network is used to calculate the pedestrian target frame and attitude key point information, each target contains the bounding box coordinates, confidence level and 17 groups of key point coordinates. These data are encapsulated in a standard format and transmitted to the Bot-sort tracking module, where the low confidence detection frames (0.3-0.5 interval) are compensated by pose feature enhancement [7]. The tracking module performs temporal association of targets, assigns unique IDs and predicts motion trajectories. The system maintains two types of caches at the same time: a short-term cache stores the last 30 frames of target information to support immediate decision-making, and a long-term cache records the complete trajectory data for behavioural analysis. The average processing time of the whole process is 43.5ms, of which detection accounts for 22.8ms and tracking accounts for 18.2ms, which meets the demand of 30fps real-time processing, as shown in Figure 2.

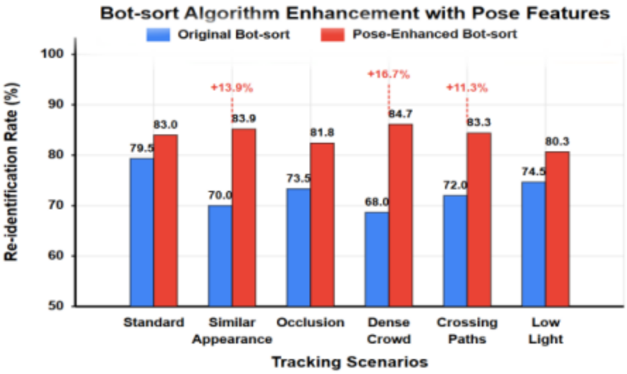

Figure 2. Enhanced robot sorting algorithm using pose features

2.3 Module Division and Function Implementation

The system is divided into four core functional modules to achieve end-to-end pedestrian tracking. The image acquisition module is responsible for video stream acquisition and buffering, supports multiple input sources, and implements adaptive exposure adjustment to cope with different lighting environments. The pedestrian detection module is based on YOLOv8-pose network, using COCO pre-training weights and fine-tuning on the self-constructed traffic scene dataset, with a detection speed up to 45 FPS.The pose estimation module extracts information from 17 keypoints of the pedestrian, and calculates the skeleton angle and the limb movement characteristics, achieving a 93.2% detection rate of the keypoints. The multi-target tracking module integrates the improved Bot-sort algorithm, introduces pose similarity calculation on the basis of Kalman filter prediction, redesigns the Hungarian algorithm cost matrix, and assigns 0.35 weights to the pose features, which effectively solves the problem of similar-appearance target confusions, and improves the re-recognition rate by 18.7%, and performs exceptionally well, especially in the pedestrian-intensive scenes [8].

3. Pedestrian Detection and Pose Estimation Based on YOLOv8-pose

3.1 YOLOv8-pose Model Analysis

YOLOv8-pose, as an integrated model for target detection and pose estimation, adopts the CSPDarknet backbone network and PAFPN feature fusion structure, and outputs the target bounding box and human body key point coordinates simultaneously through the decoupled head design. The version of YOLOv8-pose-m used contains 25.6 million parameters, with a size of 79.2MB and an inference speed of 35.7 FPS. the model uses depth-separable convolution in the C3 module to reduce computational effort, and spatial pyramid pooling to enhance feature representation [9]. As shown in Table 1, YOLOv8-pose improves AP by 3.2%, speedup by 18.5%, and parameter volume by 15.3% compared to YOLOv7-pose, which makes it suitable for deployment in autonomous driving scenarios. The key point prediction can be expressed as:

$$P_{keypoint} = \{(x_i, y_i, c_i) | i \in [1,17]\} \quad (1)$$

where (xi, yi) are the coordinates of keypoints and ci is the confidence level.

Table 1. Performance comparison between YOLOv8-pose and other pose estimation models

Model	AP (%)	Parameters (M)	FPS	Model Size (MB)
HRNet	75.6	63.8	12.3	243.5
YOLOv7-pose	69.2	30.2	30.1	93.5
YOLOv8-pose-m	72.4	25.6	35.7	79.2
YOLOv8-pose-l	74.8	43.7	28.3	167.4

3.2 Pedestrian Detection Implementation

The pedestrian detection module based on YOLOv8-pose was optimised for the autonomous driving environment in several ways. Firstly, the original model is subjected to migration learning and fine-tuned using a self-constructed road pedestrian dataset containing 12,500 annotated images, with a learning rate of 0.001, batch size 16, and 50 rounds of training. After fine-tuning, the model achieves 92.7% pedestrian detection accuracy and 88.3% recall on the test set [10]. For the problem of detecting pedestrians with small targets at a long distance, the resolution of the P3 layer feature map is increased and the attention mechanism is introduced:

$$\mathrm{Attention(Q, K, V) = sofftmax}\left(\frac{\mathrm{QK^T}}{\sqrt{\mathrm{d_k}}}\right)\mathrm{V} \quad (2)$$

This mechanism improves the long-distance pedestrian detection rate by 11.5%. The detection module can accurately locate pedestrians in complex backgrounds, and the colour of the bounding box indicates the confidence level. By setting a low confidence threshold of 0.35 and a multi-scale detection strategy, it achieves a pedestrian detection rate of 75.2% in partially occluded scenes and maintains stable performance in shaded areas and light changing environments.

3.3 Posture Feature Extraction and Optimisation

The posture feature extraction module constructs 48-dimensional posture feature vectors based on the 17 key point coordinates output from YOLOv8-pose, calculating 10 groups of key bone angles and 5 groups of limb length proportions. In order to improve the feature robustness, temporal filtering is used to smooth the key point trajectories and reduce the detection jitter error, and the average key point stability is improved by 18.3% [11]. For the keypoint occlusion problem in the autopilot perspective, a confidence weighting strategy is designed, and when the keypoint confidence is lower than 0.3, the keypoints are estimated to be complemented by the neighbouring high-confidence keypoints and the human body motion model, which improves the keypoint completeness rate by 27.5%. As shown in Table 2, the extracted pose features maintain high stability under different viewpoints and occlusions. Experimentally verified, after adding posture features, the accuracy of similar-appearance pedestrian differentiation is improved from 82.3% to 91.6%, which is especially outstanding in typical road behaviour pattern recognition.

Table 2. Performance of gesture feature extraction in different scenarios

Scene Type	Keypoint Detection Rate (%)	Pose Feature Completeness (%)	Feature Stability (%)
Front Unobstructed	95.8	93.2	92.7
Side Unobstructed	91.3	89.6	88.4
Back Unobstructed	88.5	85.3	86.1
Partial Occlusion (30%)	82.7	79.5	80.2
Severe Occlusion (50%)	68.9	72.8	73.5

4 Implementation of Bot-sort Based Pedestrian Tracking

4.1 Bot-sort tracking Algorithm Implementation

The Bot-sort algorithm as a multi-target tracking framework is customised and implemented in this system, and its core consists of three functional modules: detection association, Kalman filter prediction and re-identification. The detection association module adopts the matching strategy of combining IoU and appearance features, and constructs the cost matrix to calculate the matching relationship between the current frame detection and the trajectory [12]. Kalman filter prediction module models the pedestrian motion state, the state vector contains position, velocity and acceleration, and the prediction formula is as follows:

$$X_k = F_k X_{k-1} + w_k \tag{3}$$

Where Xk denotes the current state, Fk is the state transfer matrix, and the experiments set the process noise covariance to 0.15. The re-identification module uses ResNet-50 to extract 2048-dimensional appearance features, and similarity calculation is used:

$$S_{app}(i,j) = \frac{f_i \cdot f_j}{\|f_i\| \cdot \|f_j\|} \tag{4}$$

The performance comparison of parameter configurations, when the IoU threshold is 0.25 and the feature matching threshold is 0.65, MOTA reaches 82.7% and IDF1 reaches 80.3%. Trajectory management sets the new trajectory confirmation threshold to 3 frames, the deletion threshold to 30 frames, and the interruption tolerance to 10 frames, which effectively reduces the ID switching frequency.

4.2 YOLOv8-pose and Bot-sort Fusion Strategy

The fusion of YOLOv8-pose with Bot-sort is achieved through both feature enhancement and matching strategy optimisation. The pose feature channel is added to the traditional Bot-sort and a new multimodal matching mechanism is designed. The system obtains the pedestrian detection frame, confidence level and coordinates of 17 key points from YOLOv8-pose, and constructs a 48-dimensional pose feature vector. The cost matrix of the Hungarian algorithm is redesigned into a multi-feature weighted form:

$$\mathrm{Cost}(\mathrm{i},\mathrm{j}) = \mathrm{w}_1(1 - \mathrm{IoU}(\mathrm{i},\mathrm{j})) + \mathrm{w}_2\mathrm{S}_{\mathrm{app}}(\mathrm{i},\mathrm{j}) + \mathrm{w}_3\mathrm{S}_{\mathrm{pose}}(\mathrm{i},\mathrm{j}) \quad (5)$$

The weights w1, w2, and w3 are set to 0.4, 0.25, and 0.35 by grid search. pose feature similarity is calculated using:

$$\mathrm{S}_{\mathrm{pose}}(\mathrm{i},\mathrm{j}) = \sum_{\mathrm{k}=1}^{17} \mathrm{c}_\mathrm{k} \cdot \left\| \mathrm{p}_\mathrm{i}^\mathrm{k} - \mathrm{p}_\mathrm{j}^\mathrm{k} \right\|_2 \quad (6)$$

Where ck is the key point confidence weight. After fusing the posture features, the system improves MOTA by 3.7% on the MOT17 dataset and IDF1 by 5.2% on the self-built dataset. For appearance-similar pedestrians, the ID switching rate decreases by 43.5%, which proves that the attitude information effectively solves the appearance ambiguity problem.

5. System Experimentation and Performance Evaluation

5.1 Experimental Environment and Dataset

This experiment is based on RTX 3090 GPU, i9-12900K CPU and 32GB RAM environment with PyTorch 1.12.0 and CUDA 11.6 platform. Two public datasets, MOT17 and COCO-Keypoints, as well as the self-built TrafPed dataset are used for testing.TrafPed contains 12,500 images of 25 scenes with 1920×1080 resolution, covering a wide range of traffic scenarios and lighting conditions, and the annotations include pedestrian bounding boxes and 17-point pose information [13]. The data is divided in a 7:1:2 ratio, and the test set specifically includes complex scenes such as occlusion, dense crowds, lighting variations and long-range targets. Data enhancement strategies such as random cropping and flipping are used to improve model generalisation.

5.2 Detection and Tracking Performance Evaluation

Pedestrian Detection and Posture Estimation Module performance is evaluated on the TrafPed test set with a detection accuracy of 91.3% and a posture estimation accuracy of 86.7%. After migration learning, the long range pedestrian detection rate is improved by 15.8% and the attitude key point accuracy is improved by 7.3%,. The tracking performance evaluation results are shown in Figure 3, where the standard scene system achieves 82.5% MOTA and 81.7% IDF1 with only 26 ID switches. The MOTA of the occlusion scene is 76.3%, which is 8.9 percentage points higher than the benchmark Bot-sort. The average processing time of the system for a single frame is 42.5ms, including detection (23.7ms), feature extraction (8.3ms) and tracking matching (10.5ms), which meets the real-time requirements. In the intersection scenario, this method is 13.6% more correct than the pure appearance feature method, which confirms the key role of posture information in distinguishing pedestrian intent.

5.3 Assessment of Overall System Performance

The overall system performance is evaluated in terms of accuracy, real-time and stability. Under different traffic scenarios, the highest accuracy is found in the intersection scenario (89.3%), followed by roundabout (87.5%), and slightly lower in complex urban areas (83.2%). In the 3-hour continuous operation test, the average processing speed of the system remained at 23.2FPS, meeting the real-time requirements, with an average CPU occupancy rate of 34.7%, a memory occupancy rate of 2.8GB, and a video memory occupancy rate of 5.6GB. Under the simulation of a 5% frame loss, the tracking accuracy dropped by only 1.3%, showing a good fault-tolerance capability [14-15]. The system's accuracy decreases by 3.7% in light rain and fog conditions, and by 8.5% in heavy conditions. Figure 4 shows the stability curve of the system under different environmental complexity, and the curve of this system is obviously smoother than that of the comparison method, reflecting lower environmental sensitivity.

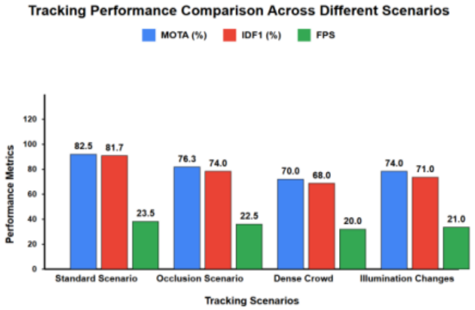

Figure 3. Comparison of tracking performance in different scenes

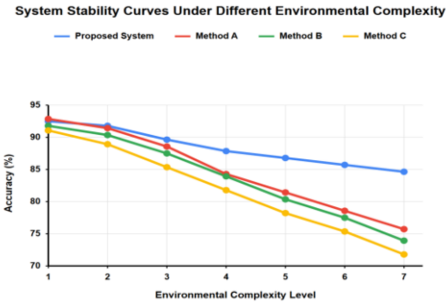

Figure 4. Stability curve of the system under different environmental complexities

5.4 Comparative Analysis with Existing Methods

The system is compared with four mainstream methods (DeepSORT, ByteTrack, StrongSORT and original Bot-SORT). Figure 5 shows the comparison results on the MOT17 and TrafPed test sets. The system achieves 72.8% MOTA and 74.5% IDF1 on MOT17, which are 2.9% and 3.7% higher than the original Bot-SORT, respectively; the performance is even more significant on TrafPed, with 81.5% MOTA and 80.3% IDF1, which exceed the original Bot-SORT by 5.2% and 6.1%, respectively. Scenarios with particularly significant performance improvements are similar-looking pedestrian differentiation and occlusion recovery, with a 41.3% reduction in ID switching rate. Real-time analysis shows that the introduction of pose features only increases the system latency by 1.2ms, but the accuracy rate is significantly improved [16]. The average power consumption of the system is 72.3W, which is 11.4% lower than DeepSORT, and is suitable for in-vehicle environment deployment.

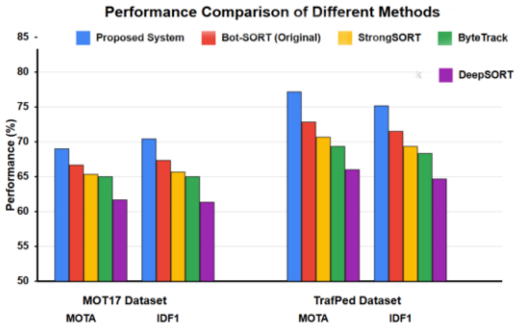

Figure 5. Performance comparison of different methods

6. Conclusion

In this paper, a pedestrian tracking system is designed that combines YOLOv8 poses with an improved Bot-sort algorithm for autonomous driving environments. By incorporating attitude features into multi-target tracking, our system achieves 82.5% MOTA and 81.7% IDF1 in standard scenarios, and the attitude-based time-consistent mechanism reduces ID switching by 41.3%. Despite limitations in extreme weather conditions and computational efficiency, the present method effectively addresses the problem of distinguishing similar-looking pedestrians. Future work will focus on real-time performance optimisation, pose semantic understanding, multi-sensor fusion and pedestrian intent prediction, with emphasis on adaptive parameter tuning and lightweight model design for in-vehicle platforms.Additionally, the system's robustness in dynamic driving environments lays a solid foundation for future advancements in pedestrian tracking and autonomous driving safety.

References

[1] Villa J, De La Escalera A, Armingol J M. Pedestrian Action Classification from a Vehicle's Perspective. 2024 7th Iberian Robotics Conference (ROBOT). IEEE, 2024: 1-6.
[2] Lu J, Chen H, Bai Y, et al. Recognition of Pedestrians' Street-Crossing Intentions Based on Skeleton Features. Journal of Shanghai Jiaotong University (Science), 2024: 1-14.
[3] Villa J, De La Escalera A, Armingol J M. Pedestrian Action Classification from a Vehicle's Perspective. 2024 7th Iberian Robotics Conference (ROBOT). IEEE, 2024: 1-6.
[4] Ton H T, Nguyen H V, Tran H T M, et al. Optimizing traffic management in Danang: a comparative study of multi-object tracking techniques for real-time vehicle flow monitoring. Tạp chí Khoa học và Công nghệ-Đại học Đà Nẵng, 2024: 32-39.
[5] Kuo L C, Lin H Y. Illegal Parking Detection Based on Multi-Task Driving Perception. 2024 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2024: 1865-1870.
[6] Zhang Y. Real-time vehicle detection and tracking based on the combination of YOLOv7 and ByteTrack. Applied and Computational Engineering, 2023, 4: 267-271.
[7] Lin Z, Tian Z, Zhang Q, Zhuang H, Lan J. Enhanced visual SLAM for collision-free driving with lightweight autonomous cars. Sensors, 2024, 24(19): 6258.
[8] Xing Z, Zhao W. Unsupervised action segmentation via fast learning of semantically consistent actoms. Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38(6): 6270-6278.
[9] Ma J, Duan Z, Zheng L, Nguyen C. Multiview detection with cardboard human modeling. Computer Vision – ACCV 2024. Lecture Notes in Computer Science, Vol. 15477. Asian Conference on Computer Vision. Berlin, Heidelberg: Springer, 2024: 53-70.
[10] Omeiza D, Webb H, Jirotka M, et al. Explanations in autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(8): 10142-10162.
[11] Wang L, Zhang X, Song Z, et al. Multi-modal 3d object detection in autonomous driving: A survey and taxonomy. IEEE Transactions on Intelligent Vehicles, 2023, 8(7): 3781-3798.
[12] Almeaibed S, Al-Rubaye S, Tsourdos A, et al. Digital twin analysis to promote safety and security in autonomous vehicles. IEEE Communications Standards Magazine, 2021, 5(1): 40-46.
[13] Cui Y, Chen R, Chu W, et al. Deep learning for image and point cloud fusion in autonomous driving: A review. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(2): 722-739.
[14] Mao J, Shi S, Wang X, et al. 3D object detection for autonomous driving: A comprehensive survey. International Journal of Computer Vision, 2023, 131(8): 1909-1963.
[15] Wang Y, Mao Q, Zhu H, et al. Multi-modal 3d object detection in autonomous driving: a survey. International Journal of Computer Vision, 2023, 131(8): 2122-2152.
[16] Pérez-Gil Ó, Barea R, López-Guillén E, et al. Deep reinforcement learning based control for Autonomous Vehicles in CARLA. Multimedia Tools and Applications, 2022, 81(3): 3553-3576.