Long-Term Pedestrian Detection and Tracking Framework Based on Model Predictive Control and YOLO

Kai Yang [20411990]

***Abstract*—This project presents a long-term multi-target pedestrian detection and tracking framework that integrates YOLOv5 with DeepSORT and enhances tracking robustness through Model Predictive Control (MPC). The system is designed to handle video input from files and live cameras. It outputs annotated videos and frame-by-frame text logs containing bounding-box coordinates and object IDs. Pedestrian detection is performed using a fine-tuned YOLO model, trained on a hybrid dataset composed of real-world pedestrian data (e.g., WIDERPerson) and synthetic scenarios to improve generalization in complex environments. After applying non-maximum suppression (NMS), the system uses DeepSORT for data association using a Kalman filter and cosine distance metric from a pre-trained ReID model. To address challenges in long-term tracking, such as temporary occlusions, re-identification, and motion prediction beyond the visible range, an MPC module is integrated into the pipeline. Experimental evaluation on test videos demonstrates that the proposed framework achieves accurate and consistent pedestrian tracking in challenging scenarios while maintaining real-time performance on low-performance hardware.**

I. Introduction

Pedestrian tracking has always been one of the core tasks in computer vision and has a wide range of practical application value, covering multiple scenarios such as urban security, sports broadcasting, etc. Traditional approaches relied on hand-crafted features and classic tracking frameworks, which often struggled in complex real-world scenarios involving occlusions and diverse appearances of pedestrians [1], [2]. In recent years, due to the rapid development of deep learning and multi-target tracking (MOT) algorithms, pedestrian tracking technology has made significant progress [3], [4].

On the detection side, early deep learning-based methods such as R-CNN improved detection accuracy by leveraging convolutional features [5]. Recurrent neural networks (RNNs) were also explored to incorporate temporal dependencies [6], although they often suffered from high computational cost. The emergence of one-stage detectors like YOLO (You Only Look Once) brought a breakthrough in real-time performance [7]. The YOLO series balances speed and accuracy well, making it a popular backbone for real-time pedestrian detection in MOT systems [8].

For multi-target tracking, Simple Online and Realtime Tracking (SORT) proposed by Bewley et al. [9] leverages Kalman filtering and the Hungarian algorithm for data association, achieving high speed but limited robustness under occlusion. DeepSORT extends SORT by integrating a deep appearance descriptor, significantly improving tracking robustness in crowded and dynamic scenes [10].

Despite these advancements, challenges remain, including the handling of long-term occlusion, domain adaptation for cross-scene tracking, and balancing accuracy with computational efficiency.

In this work, I propose a long-term pedestrian detection and tracking framework that integrates YOLOv5 and DeepSORT into a unified pipeline, with an additional Model Predictive Control (MPC) module to enhance tracking continuity over extended periods. The proposed framework has the following key features:

- **Human-centric detection**: A high-performance detection module fine-tuned on both real-world and synthetic datasets for improved generalization in diverse environments.
- **Long-Term Tracking**: Integration of Model Predictive Control (MPC) to enhance robustness under occlusion and prolonged visibility gaps.
- **Support for live Camera Input**: Supports both video files and live camera input, producing annotated output videos and frame-by-frame tracking logs.
- **Real-Time Performance on Low-End Hardware**: Maintaining real-time performance on low-performance hardware.
- **Integrated Graphical User Interface (GUI)**: The system includes a user-friendly graphical interface that allows users to select model weights.

II. Proposed Method

A. Overview of Proposed Method

The proposed method presents an integrated framework for long-term, multi-target pedestrian detection and tracking by combining a fine-tuned YOLOv5 object detector with DeepSORT tracking and a Model Predictive Control (MPC) module for enhanced temporal consistency.

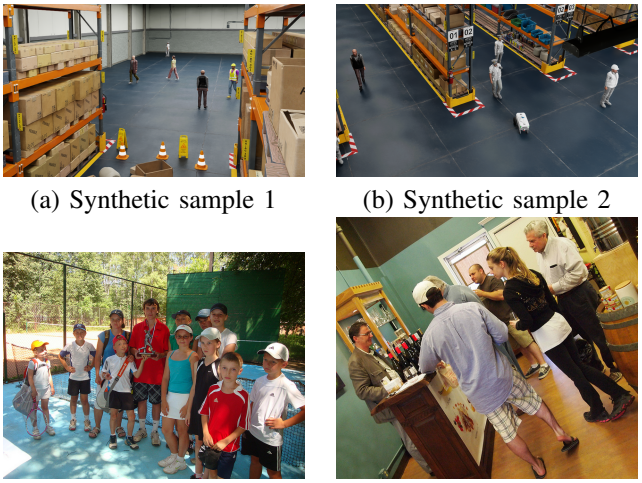
The filtered outputs are then formatted and passed into DeepSORT, which uses a combination of motion modeling (via Kalman filtering) and appearance descriptors (via a pre-trained ReID model) to assign persistent track IDs across frames. To address challenges such as identity switching, temporary occlusions, and prolonged detection gaps, an optional Model Predictive Control (MPC) module is introduced after the DeepSORT stage. MPC leverages recent trajectory history

to predict future positions of tracked pedestrians over a short horizon.

B. Detection Module

YOLOv5, developed by Ultralytics, is a PyTorch-based implementation offering lightweight architecture and real-time inference [11].It employs a one-stage detection pipeline that directly regresses bounding box coordinates and objectness scores, enabling fast and accurate object localization in a single forward pass.

In The framework, the YOLOv5 model is fine-tuned on a curated composed of both real-world pedestrian datasets (WIDERPerson) and synthetically generated pedestrian-rich scenes. In this work, synthetic dataset was generated by NVIDIA Isaac Sim, a high-fidelity simulation platform tailored for robotics. Meanwhile, domain randomization techniques were also applied to the simulated environment to introduce variability, such as camera noise and lighting changes. Domain randomization is a widely used approach for sim-to-real transfer, involving deliberate perturbations of parameters related to actuators, sensors, lighting [**?**]. To ensure a smooth transition from simulation to the real world, we integrated the synthetically generated data with real-world pedestrian datasets. This hybrid training approach enables the detection model to learn more diverse and generalized feature representations, thereby improving the adaptability and robustness to various real-world data.



(a) Synthetic sample 1 (b) Synthetic sample 2

(c) Real-world sample 1 (d) Real-world sample 2

Fig. 1. Examples of training data used for fine-tuning. Top: samples from synthetic dataset generated in NVIDIA Isaac Sim with domain randomization. Bottom: samples from real-world WIDERPerson dataset.

The outputs from YOLOv5 include bounding boxes, confidence scores, and class labels, which are further processed using Non-Maximum Suppression (NMS) to filter redundant detections before being passed to the tracking module.

C. Tracking Module with MPC Integration

The tracking module is built upon DeepSORT [10], which enhances the classical SORT algorithm by incorporating both motion and appearance cues. To further improve trajectory stability and short-term prediction in dynamic scenarios, we integrate a Model Predictive Control (MPC) layer that refines the trajectory estimation beyond conventional Kalman prediction.

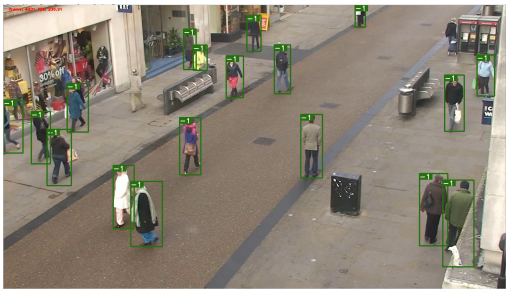


Fig. 2. Visualization of YOLOv5 detection outputs after Non-Maximum Suppression (NMS).

a) State Estimation via Kalman Filter.: Each track's state is defined as a vector $\mathbf{x} = [u, v, \gamma, h, \dot{u}, \dot{v}, \dot{\gamma}, \dot{h}]^T$ capturing position, size, and velocity. The Kalman filter follows a standard linear dynamical system:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}, \quad \mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q} \quad (1)$$

where $\mathbf{F}$ is the transition matrix and $\mathbf{Q}$ the process noise.

b) Appearance Matching.: To distinguish identities in crowded scenes, we compute an appearance descriptor $\mathbf{r} \in \mathbb{R}^{128}$ for each detection using a CNN. The cosine distance evaluates feature similarity:

$$d_{\cos}(\mathbf{r}_i, \mathbf{r}_j) = 1 - \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{\|\mathbf{r}_i\|\|\mathbf{r}_j\|} \quad (2)$$

c) Data Association.: Motion-based Mahalanobis distance d_M and appearance distance $d_{\cos}$ are combined to form the cost matrix for the Hungarian algorithm:

$$\text{Cost}_{i,j} = \lambda d_M(i,j) + (1-\lambda) d_{\cos}(i,j) \quad (3)$$

where $\lambda \in [0,1]$ controls the trade-off.

d) Trajectory Refinement via MPC.: Although Kalman filtering provides a one-step prediction, it lacks foresight into future dynamics. Therefore, we introduce a Model Predictive Controller (MPC) that optimizes predicted pedestrian positions over a horizon H.

Given a motion model:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t \quad (4)$$

we define the MPC cost function as:

$$J = \sum_{i=0}^{H} \left(\|\mathbf{x}_{t+i} - \mathbf{x}_{t+i}^{\text{ref}}\|_Q^2 + \|\mathbf{u}_{t+i}\|_R^2 \right) \quad (5)$$

where $\mathbf{x}^{\text{ref}}$ is the desired trajectory, Q, R are weight matrices, and $\mathbf{u}_t$ are control inputs. The MPC minimizes J subject to system dynamics and constraints, yielding smoothed, anticipative paths.

e) Benefit.: This hybrid framework improves both tracking continuity and trajectory smoothness, especially in scenarios with abrupt motion, occlusion recovery, or high crowd density.

Fig. 3. Visualization of Trajectory tracing via MPC

D. GUI

The proposed system is equipped with a user-friendly GUI built using Tkinter, allowing users to easily configure and run the tracking process. The interface enables users to:

- **Select Input Video**: Load a video file for processing.
- **Configure Parameters**: Adjust key parameters such as frame interval, confidence threshold, and IOU threshold.
- **Choose Output Folder**: Specify the folder where the processed results will be saved.
- **Start/Stop Tracking**: Begin tracking with a simple button click and stop it at any time.

The system runs the tracking process in a separate thread to ensure a responsive interface during video processing.

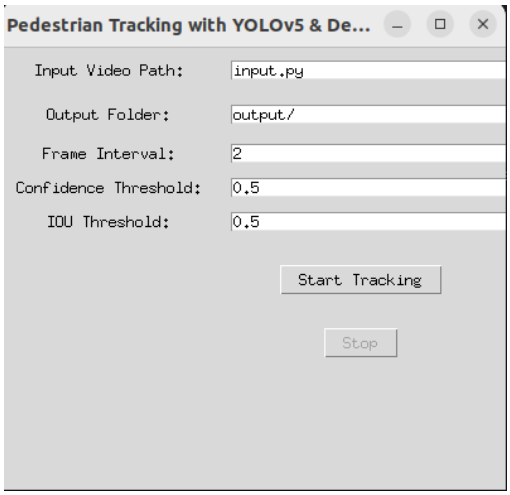


Fig. 4. Graphical User Interface (GUI)

III. EXPERIMENTS RESULT

To validate the feasibility and robustness of the proposed framework, I conducted experiments in real-world. The sliding window K was set to 10, and the prediction timestamp was set to 20. I use Robomaster AI robot as the hardware carrier platform, and the onboard computer based on Intel N100 ultra-low power processor for real-time processing. Rplidar S2 and Bosch BMI088 were adopted as Lidar and IMU sensor respectively.

A. Detection Evaluation

To assess the effectiveness of the fine-tuned YOLOv5 pedestrian detector, I conduct a benchmark evaluation on the **WIDERPerson validation set**, which contains diverse pedestrian appearances and scene complexities. Following the MOT setting and the nature of the application, I only evaluate detection performance for the `person` class. All other categories are excluded during both inference and metric calculation. I compare proposed model against two baselines:

- **YOLOv5s (pretrained)**: The original model trained on the COCO dataset.
- **YOLOv8n (pretrained)**: A lightweight version of the latest YOLOv8 detector.

proposed model, **YOLOv5s (fine-tuned)**, is trained on a hybrid dataset consisting of real-world pedestrian data and synthetic data generated using NVIDIA Isaac Sim with domain randomization.

a) Evaluation Metrics: We adopt standard object detection metrics:

- **mAP@0.5**: Mean Average Precision at IoU threshold 0.5.
- **mAP@0.5:0.95**: Average precision across IoUs from 0.5 to 0.95 in steps of 0.05.
- **Precision / Recall**: Used to plot PR curves.

b) Results: Table I summarizes the comparison. The fine-tuned model outperforms the COCO-pretrained YOLOv5s and also slightly surpasses YOLOv8n in mAP metrics.

TABLE I
DETECTION PERFORMANCE ON WIDERPERSON VALIDATION SET.

Method	mAP@0.5	mAP@0.5:0.95	Precision / Recall
YOLOv5s (pretrained)	72.1%	41.6%	0.78 / 0.64
YOLOv8n (pretrained)	75.4%	45.0%	0.82 / 0.67
YOLOv5s (fine-tuned)	**78.4%**	**48.7%**	**0.85 / 0.69**

B. Crowded and Occlusion Testing

To further evaluate the robustness and effectiveness of proposed framework in more challenging real-world environments, we conducted tests in scenarios with heavy pedestrian crowding and occlusion videos. These types of environments are particularly difficult for pedestrian tracking systems due to the increased object occlusions and the complex visual clutter.

I tested the proposed framework in a scenario where pedestrians are densely packed, and occlusions occur frequently. The benchmark for comparison is YOLOv8 combined with DeepSORT, which is a lightweight and recent object detection and tracking solution.

In the first image (Figure 5), I show the results of tracking with YOLOv8 and DeepSORT. In this crowded scenario, the system struggles to maintain consistent tracking, with multiple identity switches and missed detections due to occlusions. Meanwhile, the proposed method, as shown in the

second image (Figure 6), demonstrates improved tracking performance in the same environment. The addition of Model Predictive Control (MPC) helps predict the motion of occluded pedestrians, maintaining consistent object identity even in the presence of partial occlusions.

Fig. 5. Tracking performance with YOLOv8 + DeepSORT.

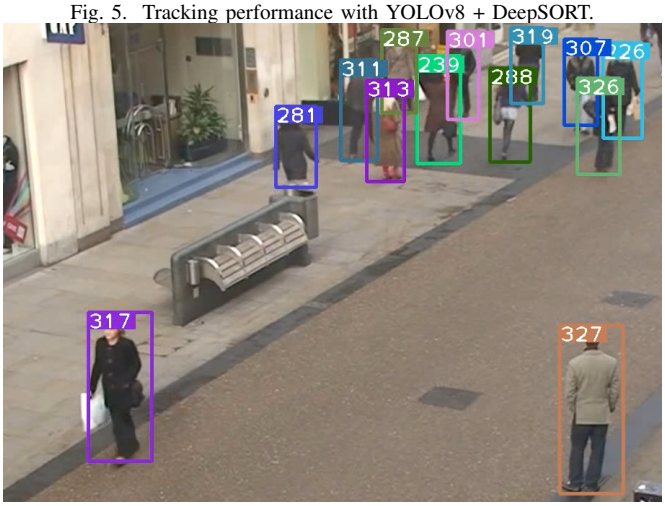


Fig. 6. Tracking performance with the proposed method.

The results clearly demonstrate that proposed framework improves the tracking accuracy and robustness, especially in scenarios where occlusions and complex interactions between pedestrians occur frequently. The combination of fine-tuned YOLOv5 for pedestrian detection and MPC for motion prediction contributes to significantly more stable and consistent tracking outcomes in crowded and occluded environments.

C. Real-time response test

To assess the real-time capability of the proposed framework, we measured the average inference time per frame and calculated the effective frame-per-second (FPS) during video processing. Tests were conducted on a system equipped with an Intel i3 CPU, and the video stream used for testing was at 720p resolution. The proposed framework achieves an average processing speed of **43 FPS**, which satisfies real-time requirements ($\geq$25 FPS).

IV. CONCLUSION

In this report, I proposed a long-term multi-target pedestrian tracking framework that integrates YOLOv5 for detection, DeepSORT for short-term identity association, and model predictive control (MPC) for enhanced motion prediction and robustness. The system is capable of processing both offline videos and live camera streams, providing annotated video output and structured logs containing bounding box locations and target identities for each frame.

To improve generalization and tracking accuracy in complex real-world scenarios, YOLOv5 model is fine-tuned on a hybrid dataset consisting of real-world pedestrian dataset (WIDER-Person) and synthetic dataset from a high-fidelity simulation environment. In the synthetic dataset generation process, I also adopted domain randomization techniques to bridge the gap between simulation and reality. DeepSORT is used to handle short-term data association based on motion and appearance features, while the MPC module is used to enhance long-term tracking performance, especially in the case of occlusion, identity switching, or temporary disappearance. This hybrid design enables the system to maintain high accuracy and robustness in a variety of tracking scenarios.

Experimental results confirm that the proposed framework can achieve accurate, stable and real-time pedestrian tracking even under resource-constrained hardware conditions.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.

[2] M.-H. Yang, F. Lv, W. Xu, and Y. Gong, "Detection driven adaptive multi-cue integration for multiple human tracking," in *2009 IEEE International Conference on Computer Vision*. IEEE, 2009, pp. 1554–1561.

[3] Y. Dai, Z. Hu, S. Zhang, and L. Liu, "A survey of detection-based video multi-object tracking," *Displays*, vol. 75, p. 102317, 2022.

[4] A. Singh, P. Kaur, and N. Kaur, "Object tracking using computer vision: A review," *Computers*, vol. 12, no. 6, p. 136, 2023.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," in *arXiv preprint arXiv:1603.00831*, 2016.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[8] G. Jocher, "ultralytics/yolov5: v2.0," 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370

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

[10] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, 2017.
[11] Ultralytics, "Yolov5 by ultralytics," 2020, https://github.com/ultralytics/yolov5.