Dean&Francis
ISSN 2959-6157

Robomaster Robot Automatic Targeting System Based on YOLOv11n and DeepSORT

Feiyang Liu

Leicester International Institute,
Dalian University of Technology,
Liaoning, China, 116024
jasonlfy@mail.dlut.edu.cn

Abstract:

Robomaster is a competitive robotics event where only hits on a robot's four armor plates count as valid damage. To achieve fast and precise tracking of these armor plates, especially in the face of the enemy robot's high-speed movements and the complex background of the arena, this paper proposes an automatic aiming system. The system integrates the YOLOv11n detection algorithm for target recognition and the DeepSORT tracking algorithm for continuous target tracking. The YOLOv11n model is responsible for detecting the armor plates in real-time, while DeepSORT keeps track of the target's trajectory across multiple frames. To handle target occlusion and improve tracking accuracy, the system uses a Kalman filter to predict the robot's movement and incorporates deep appearance features for more robust data association. This combination helps maintain precise tracking even when the target is partially blocked. Experimental results show that the proposed system significantly enhances the robot's ability to track and hit the enemy's armor plates, leading to improved accuracy in dynamic and challenging competition scenarios.

Keywords: Target Tracking, Computer Vision, YOLOv11n, DeepSORT, Kalman Filter

1. Introduction

In Robomaster competitions, only hits on an enemy robot's armor plates count as valid damage. Additionally, robots perform "small gyro-style" spins at high speeds to evade projectile impacts, resulting in low hit rates when operators rely solely on manual aiming[1]. Developing an automatic targeting system capable of real-time target detection and stable tracking is crucial. By processing real-time video streams captured by onboard cameras, visual algorithms can automatically locate armor plates, continuously track their movement trajectories, and provide precise information to the gimbal control system.

This paper designs a real-time, precise, and robust automatic aiming system: First, all targets are detected in each video frame. Then, a correlation algorithm links these detections across different frames

to form a coherent motion trajectory. The system employs the high-performance, lightweight YOLOv11n [2] as the detector, combined with the DeepSORT [3] tracking algorithm. Automatic targeting systems hold vast application prospects in the field of robotics, spanning multiple sectors including military, industrial, and medical domains. In the military sector, these systems significantly enhance weapon accuracy and reaction speed. In industrial settings, these systems can be integrated into automated production lines to boost efficiency and precision. Within healthcare, they facilitate precise surgical targeting and radiation therapy, enabling doctors to perform procedures with greater accuracy. Overall, the widespread adoption of automatic targeting systems not only elevates operational efficiency across industries but also brings convenience to both production and daily life.

2. Related Work

2.1 Object Detection Algorithm

With the advancement of deep learning, convolutional neural networks (CNNs) have become the mainstream technology for object detection. CNN-based detectors can be categorized into two-stage and single-stage approaches. Two-stage detectors first generate a series of regions of interest (ROIs) that may contain objects, followed by detailed classification and localization within these regions[4]. While this method offers high accuracy, it incurs significant computational overhead. Due to constraints in onboard computer hardware, real-time performance is limited.

Single-stage detectors eliminate the step of generating candidate regions. Algorithms like the YOLO series [5-6] simultaneously predict object bounding boxes and categories. This design significantly boosts detection speed, making it highly suitable for applications requiring real-time responses. The YOLOv11n model selected in this paper is a newer lightweight version that maintains high detection performance while substantially reducing computational complexity [2].

2.2 Multi-Target Tracking Algorithm

Multi-object tracking involves associating multiple appearances of the same object within a video to generate a single trajectory for each target. The DeepSORT algorithm employs the Hungarian algorithm [7], using the Intersection over Union (IoU) between prediction and detection boxes as the matching criterion, while incorporating a deep learning-based appearance model. It utilizes a pre-trained neural network to extract a feature vector—the appearance descriptor—for each detected object. During data association, DeepSORT integrates both motion and appearance information: it uses the Mahalanobis distance to measure motion matching, while employing the cosine distance between feature descriptors to assess appearance similarity. Consequently, DeepSORT becomes more robust in data association, capable of re-identifying a target after brief occlusion.

3. System Design and Implementation

3.1 Overall System Architecture

The system architecture consists of two core modules: the detection module and the tracking module. The overall workflow is illustrated in Figure 1.

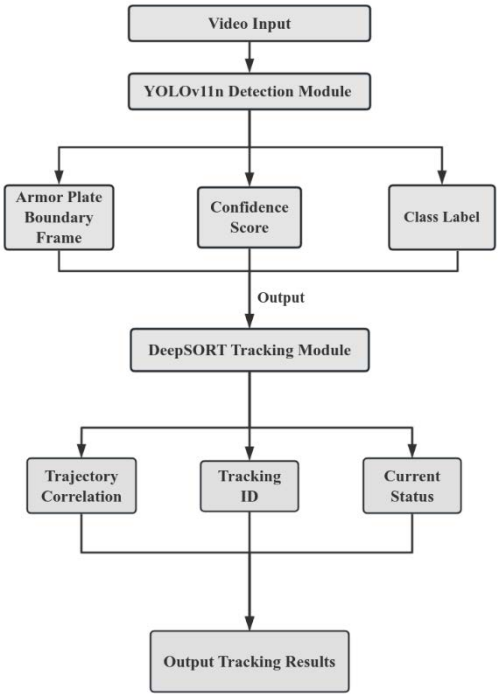

Figure 1. Overall Framework Diagram of the Auto-Aiming System

a. Video Input: The onboard camera captures real-time footage and feeds each frame into the system.

b. Detection Module: YOLOv11n performs object detection on each frame, outputting bounding boxes for all detected armor plates, their respective confidence scores, and category labels (red armor plate, blue armor plate, and plate serial numbers).

c. Tracking Module: Detection results from the current frame are passed to the DeepSORT tracking module. These results are associated with target trajectories. It maintains an ID for each tracked armor plate using a com-

ISSN 2959-6157

bination of motion prediction and appearance matching. The final output includes tracking results for each target: bounding boxes, stable tracking IDs, and current status.

3.2 YOLOv11n-based Detection Module

Considering the real-time requirements and limited computational resources of the robot, this paper selects YOLOv11n as the detection module algorithm. YOLOv11n is a lightweight version within the YOLO series, achieving a balance between computational speed and accuracy [2].

The network architecture of YOLOv11n consists of three main components[8]:

a. Backbone: CSPDarknet is employed as the feature extraction network. It efficiently extracts multi-level features from input images while maintaining a minimal number of parameters.

b. Neck: The PA-FPN architecture is utilized for multi-scale feature fusion. This mechanism combines top-down and bottom-up paths, effectively integrating high-level semantic information with low-level positional information—crucial for recognizing objects of varying sizes.

c. Head: This component generates the final prediction results. It processes the fused feature maps and outputs a vector containing the bounding box coordinates (x, y, width, height), confidence scores, and class probabilities for each detected armor plate.

3.3 DeepSORT-based Tracking Module

The tracking module is responsible for maintaining the ID of each armor plate across consecutive video frames. The DeepSORT algorithm is employed, which builds upon the SORT algorithm by incorporating appearance information to enhance tracking robustness in complex scenes [3,9]. The workflow of DeepSORT is as follows[10]:

a. State Prediction: For each target from the previous frame, a Kalman filter is used to predict its state in the current frame. The state vector $x=[u,v,r,h,\dot{u},\dot{v},\dot{r},\dot{h}]$ contains the target's bounding box center coordinates (u, v), aspect ratio (r), height (h), and velocity information. The Kalman filter prediction helps narrow the search area for the target.

b. Data Association: The goal is to match predicted trajectories with new detections from YOLOv11n. DeepSORT employs a cascaded matching strategy combining motion and appearance information:

(1) Motion Matching: Calculates motion similarity between predicted states and new detections using Mahalanobis distance, accounting for prediction uncertainty from the Kalman filter. Detections close to predicted states are considered matching candidates.

(2) Appearance Matching: For targets unmatched by motion alone, an appearance model is employed. A pre-trained ID network extracts appearance features from each detection. Appearance similarity is evaluated by calculating cosine distances with existing trajectory features. Smaller cosine distances indicate higher similarity.

(3) Match Cascade: DeepSORT prioritizes matching recent targets based on the number of frames since their last match. Trajectories updated most recently receive priority, effectively handling short-term occlusions. The Hungarian algorithm is ultimately applied to find optimal matches.

c. Trajectory Management:

(1) Trajectory Confirmation: A newly detected object that cannot be matched to an existing trajectory is initialized as a provisional trajectory. It is only confirmed and assigned an ID after successful matching within several frames.

(2) Trajectory Deletion: If a trajectory fails to match any detection within the set maximum frame count, it is deleted to prevent the accumulation of redundant trajectories.

By integrating motion and appearance information, DeepSORT effectively handles occlusions, multi-object interactions, and rapid motion, ensuring stable tracking of the armor plate.

4. Experiments and Results Analysis

4.1 Experimental Setup

(1) Hardware Platform: The experiment was conducted on an onboard computing device configured with an Intel Core i7-13620H CPU, 16GB of memory, and an NVIDIA GeForce RTX 4060 GPU. Image acquisition utilized an industrial camera with a resolution of 1440×1080 and a frame rate of 249 FPS, as shown in Figure 2.

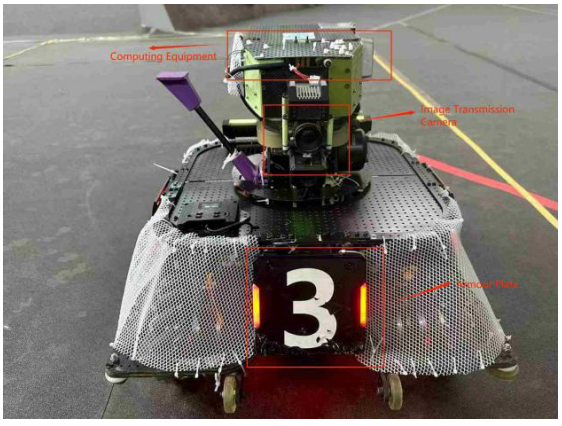

Figure 2. Physical Robot Appearance

(2) Software Environment: Operating system Ubuntu 20.04, deep learning framework PyTorch, with GPU acceleration enabled via CUDA and cuDNN. Image process-

ing and video input/output utilize the OpenCV library.
(3) Dataset: This paper created a dataset featuring complex scenarios including varying lighting conditions, distances, fast-moving targets, and partial occlusions. Armor plates were annotated with bounding boxes to train YOLOv11n and evaluate tracking performance, enhancing model robustness. Training results are shown in Figure 3.

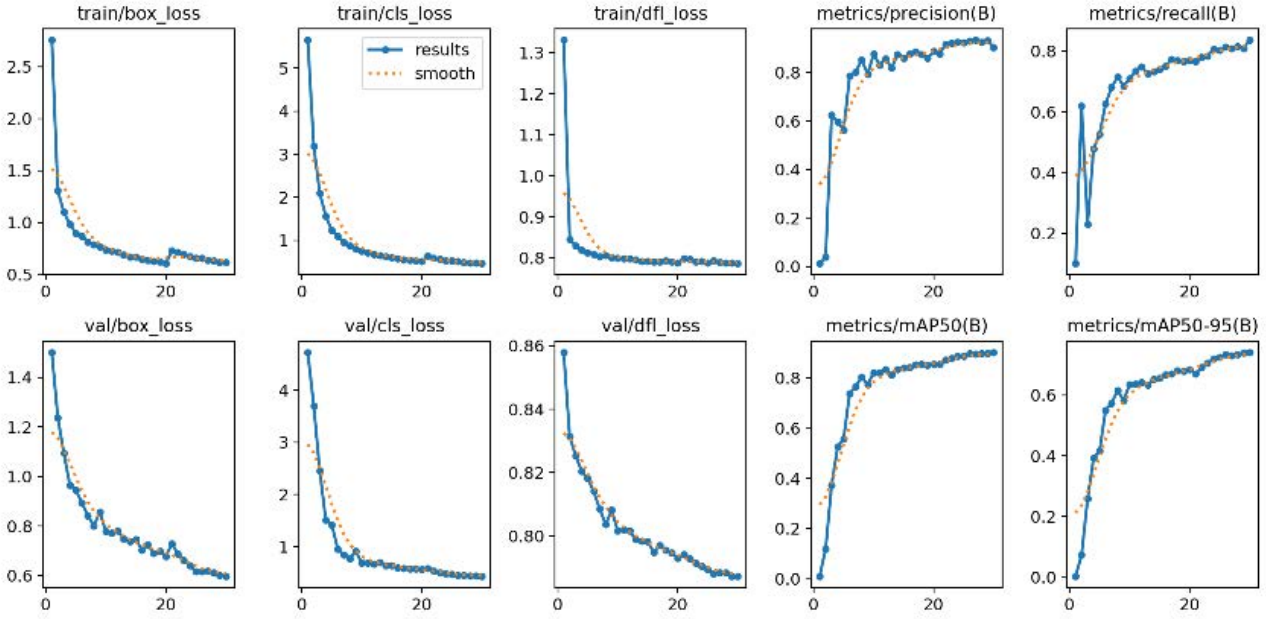

Figure 3. Training Results

4.2 Analysis of Experimental Results

This paper evaluates the performance of automatic targeting systems by measuring their hit rates in a laboratory simulation of real combat scenarios. The target robot was controlled to move at a rotational speed of approximately 3 rad/s and a translational speed of approximately 0.5 m/s. Hit rates were tested using manual aiming, traditional vision + SORT algorithm, and the proposed YOLOv11n + DeepSORT algorithm. The test results are shown in Table 1.

Table 1. Accuracy Data of of Hit

Striking Method	Translation	Rotation	Translation+ Rotation
Manual Aiming	13.2%	35.7%	9.6%
Traditional Vision + SORT	60.7%	80.6%	65.3%
YOLOv11n+DeepSORT	79.5%	87.8%	72.4%

The system was also tested under several complex scenarios. Even when the armor plate was partially obstructed, it maintained stable tracking of multiple targets. When two robots crossed paths, the system successfully prevented identity confusion, with tracking performance demonstrated in Figure 4.

ISSN 2959-6157

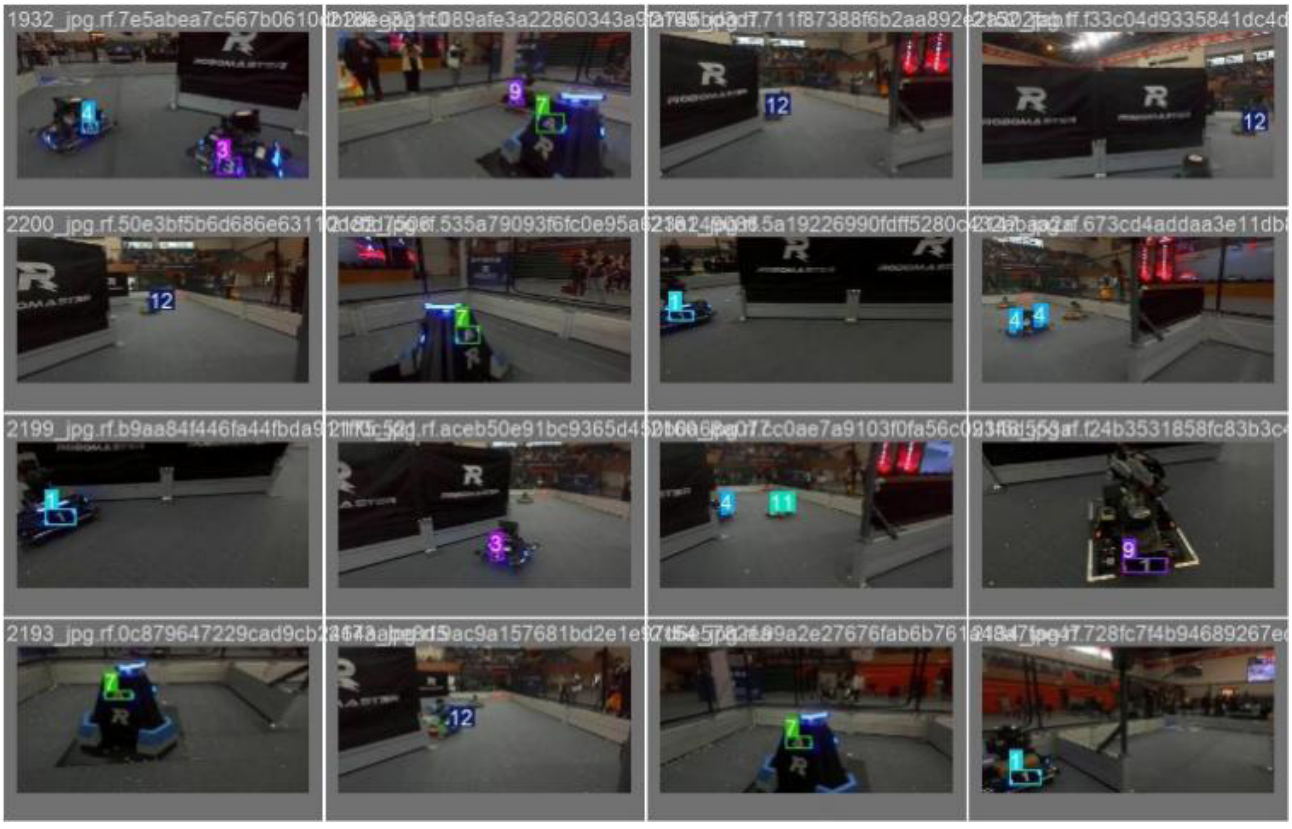

Figure 4. Tracking Effect

To quantitatively evaluate the performance of multi-object tracking systems, this paper employs the CLEAR MOT evaluation metrics, which primarily include[11]:

a. Multi-Object Tracking Accuracy (MOTA): This is a comprehensive metric that integrates three types of errors: false positives (FP), false negatives (FN), and identity switching (IDSW), along with the total number of ground truth targets (GT). It provides a holistic assessment of tracking performance and is calculated as follows:

$$MOTA = 1 - \frac{FP + FN + IDSW}{GT} \tag{1}$$

A higher MOTA score indicates better tracking performance.

b. Multi-Object Tracking Precision (MOTP): This metric measures the alignment between the predicted bounding boxes of correctly tracked objects and their ground truth bounding boxes, typically calculated as the average Intersection over Union (IoU) of all matched pairs. A higher MOTP indicates more precise localization.

c. Identity Switching Rate (IDSW): This metric counts how many times a tracked trajectory switches its corresponding ground truth identity. A lower IDSW is better, indicating stronger identity persistence capability.

d. Frames Per Second (FPS): This metric evaluates the overall processing speed of the system, indicating whether it meets real-time requirements.

Although the SORT algorithm is simple and efficient, it is susceptible to occlusions and fast-moving objects in multi-object and complex scenes, often leading to target loss (misses) and identity switching (IDSW) issues. DeepSORT builds upon SORT by incorporating deep learning models, specifically feature extraction networks, to generate unique feature vectors for each object. This enables DeepSORT to consider not only positional and motion information but also visual characteristics during object matching. Consequently, DeepSORT achieves greater accuracy in distinguishing similar objects within complex scenes, thereby reducing IDSW and false negatives (FN), and enhancing the precision and stability of multi-object tracking.

The proposed system YOLOv11n + DeepSORT achieves the highest MOTA score while significantly reducing identity switches (from 98 to 21) compared to traditional vision + SORT, with only a slight decrease in FPS. This demonstrates its superior overall tracking performance.

Experiments have demonstrated that this system achieves significantly improved strike accuracy against robots compared to manual aiming and traditional auto-aiming.

5. Conclusion

This paper designs an automated targeting system integrating YOLOv11n detection with DeepSORT tracking. Compared to traditional visual recognition methods, it achieves higher accuracy in Robomaster competitions. Compared to SORT algorithms, it significantly reduces ID switching for more stable tracking. It demonstrates excellent performance across various scenarios, including varying distances, lighting conditions, occlusions, and close-range interactions. Although the primary objectives have been achieved, the system still has several viable avenues for future improvement: Continue pruning, quantization, and TensorRT acceleration; enhance re-identification and trajectory prediction after long-term occlusion; implement end-to-end tracking with Transformers; and integrate results with gimbal PID and predictive aiming.

References

[1] Liang, R. Z., Li, C. M., Xie, X. P., et al. (2021). Development of Computer Vision Applications Based on the RoboMaster Competition. Science and Innovation, (3), 106-108.

[2] Li Wei, Li Chunshu, Che Jin, et al. (2025). Road Vehicle Object Detection Algorithm Based on Improved YOLOv11n. Internet of Things Technology, 19, 33-38.

[3] Wojke, N., Bewley, A., & Paulus, D. (2017). Simple Online and Realtime Tracking with a Deep Association Metric. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 3645-3649).

[4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS) (Vol. 28).

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779-788).

[6] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[7] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2(1-2), 83-97.

[8] Khanam, R., & Hussain, M. (2024, October 24). YOLOv11: An overview of the key architectural enhancements. arXiv:2410.17725.

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

[10] El-alami, A., Nadir, Y., & Mansouri, K. (2024). A modified lightweight DeepSORT variant for vehicle tracking. International Journal of Advanced Computer Science and Applications, 15(10).

[11] Li, M., Liu, M., Zhang, W., Guo, W., Chen, E., & Zhang, C. (2024). A Robust MultiCamera Vehicle Tracking Algorithm in Highway Scenarios Using Deep Learning. Applied Sciences, 14(16), 7071.