ISSN: 2472-3703 DOI: 10.6911/WSRJ.202507_11(7).0004

Design and Implementation of Multi-Target Detection Based on YOLO and SORT Algorithms

Gang Zhao

Aircraft Strength Research Institute of China, Xi'an Shaanxi 710065, China

Abstract

Multi-target detection is one of the current directions in the field of computer vision, with broad application scenarios in intelligent transportation, security surveillance, and other domains. Traditional detection systems employ classifiers to evaluate different slices of test images. Previous detection algorithms suffered from slow speeds and difficulties in optimization. Through several generations of development, the YOLO (You Only Look Once) algorithm, with its end-to-end detection architecture, has significantly improved processing speed, making it more suitable for real-time scenarios. This paper adopts the YOLO model for object detection and further incorporates the SORT (Simple Online and Realtime Tracking) algorithm for object tracking. Ultimately, the study implements a vehicle and pedestrian detection system, completing tasks from algorithm implementation to model training, and further constructs a user-friendly graphical interface using the QT framework.

Keywords

Multi-target detection, target tracking, YOLO model, QT interface.

1. INTRODUCTION

Real-time target detection and tracking in dynamic multi-target scenarios [1] represent a significant challenge in the field of computer vision, with a wide range of applications in autonomous driving and intelligent surveillance. To achieve vision-based object detection and tracking under dynamic and multi-target conditions, this paper divides the task into two parts: object detection and object tracking [2]. In line with the functionalities implemented in this study, object detection is performed on image files, while object tracking is applied to video streams from cameras or video files [3]. Among these, the YOLO (You Only Look Once) series of algorithms [4] serves as the cornerstone of the detection module due to its outstanding real-time performance. Meanwhile, the DeepSORT algorithm [5] addresses the problem of identity preservation in complex scenarios within the tracking module.

In 2016, YOLOv1 proposed by Redmon et al., revolutionized target detection by framing it as a single-stage regression problem. The entire image is processed by a single convolutional neural network (CNN), which divides the image into grid cells. Each grid cell directly predicts the center coordinates, width, and height of the bounding box, as well as the probability of object existence and the distribution of category probabilities. Its core innovation lies in the end-to-end architectural design, which eliminates the regional proposal step of the traditional two-stage algorithm, enabling real-time detection at the hundred-millisecond level. YOLOv2 (2017) introduces the Anchor Boxes mechanism and batch normalization to significantly improve the recall ratio [6]. YOLOv3 (2018) employs multi-scale prediction and residual networks to enhance the detection capabilities of small targets. YOLOv4 (2020) integrates Mosaic data augmentation, CSPNet, and other strategies to achieve a new balance between precision and speed [7]. YOLOv5/v7/v8 (2020-present) focus on engineering optimizations to

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202507_11(7).0004

support more flexible deployment scenarios [8]. Non-Maximum Suppression (NMS) is a crucial post-processing technique employed during the detection phase. It filters out overlapping, low-confidence bounding boxes to ensure optimal detection results.

Target tracking needs to address the frame straddling association of detection frames. In 2017, the SORT (Simple Online and Realtime Tracking) algorithm predicted motion trajectories using Kalman filtering [9] and established the correlation between detection frames and trajectories through the Hungarian algorithm. However, its drawback is that it relies solely on motion features, making it susceptible to ID Switch in occlusion scenarios.

DeepSORT (2017) enhances the SORT algorithm by incorporating appearance feature association. It employs a pre-trained re-identification (ReID) network to extract target appearance embedding vectors and create appearance descriptors. By integrating the association metrics of motion features (Mahalanobis distance) and appearance features (Cosine distance), the robustness of tracking in occlusion scenarios is significantly improved, leading to a 45% reduction in ID Switch.

This paper implements dynamic multi-target sensing based on the YOLO-DeepSORT fusion frames:

Image processing: YOLO is used to perform single frame target detection.

Video streaming process: The YOLO detection results are input into DeepSORT to construct continuous trajectories through spatio-temporal correlation. This approach balances detection accuracy with tracking stability, offering an efficient solution for multi-target sensing in complex environments.

2. ALGORITHM FRAMEWORK

The YOLO algorithm redefines object detection as a regression problem. It utilizes a single convolutional neural network (CNN) to process the entire image, dividing it into grids and predicting class probabilities and bounding boxes for each grid. For each grid, the network predicts a bounding box along with a probability for each class (vehicle, pedestrian). Each bounding box can be characterized by four descriptors: the center coordinates, height, width, and the value associated with the object's class. Additionally, the algorithm predicts the probability of an object's presence within the bounding box. If the center of an object lies within a grid cell, that grid cell is responsible for detecting the object. There will be multiple bounding boxes within each grid. During training, we aim to have only one bounding box per object. Consequently, we designate a box to be responsible for predicting an object based on which box exhibits the highest overlap with the ground truth box. Finally, we implement a method called "Non-Max Suppression" to each class of objects to filter out bounding boxes with confidence scores below the threshold.

In this paper, we utilize four well-trained YOLO models: YOLO v3/v4, Tiny YOLO v3/v4. While the overall steps for these four models are quite similar, but there are notable differences in their underlying networks, multi-scale predictions, data augmentation techniques, and other aspects. For instance, YOLO v4 differs from YOLO v3 in several key areas as follows:

a) YOLO V4 utilizes CSPDarkNet53, whereas YOLO V3 employs DarkNet53;

b) Compared to the Feature Pyramid Network (FPN) used in YOLO V3, YOLO V4 incorporates Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN);

c) CutMix data augmentation and Mosaic data augmentation;

d) DropBlock regularization.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202507_11(7).0004

Regarding the selection of YOLO models, we can modify the model path in the code to select the corresponding model, or we can load the desired model using the model selection button in the QT interface, as shown in Fig. 1.

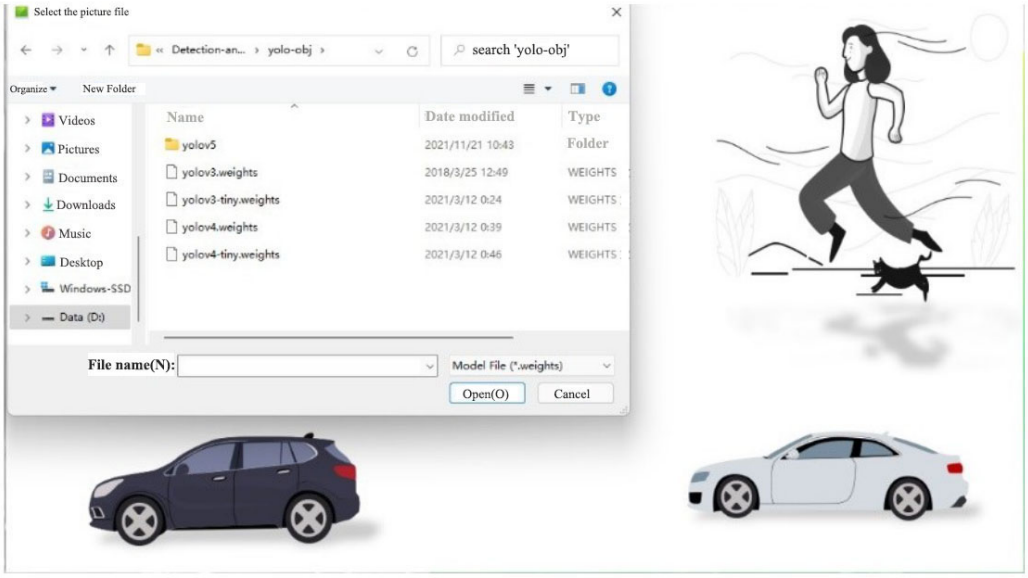

Figure 1. Selection of YOLO training model weights

The algorithm of this paper is drawn as a flowchart as shown in Fig. 2.

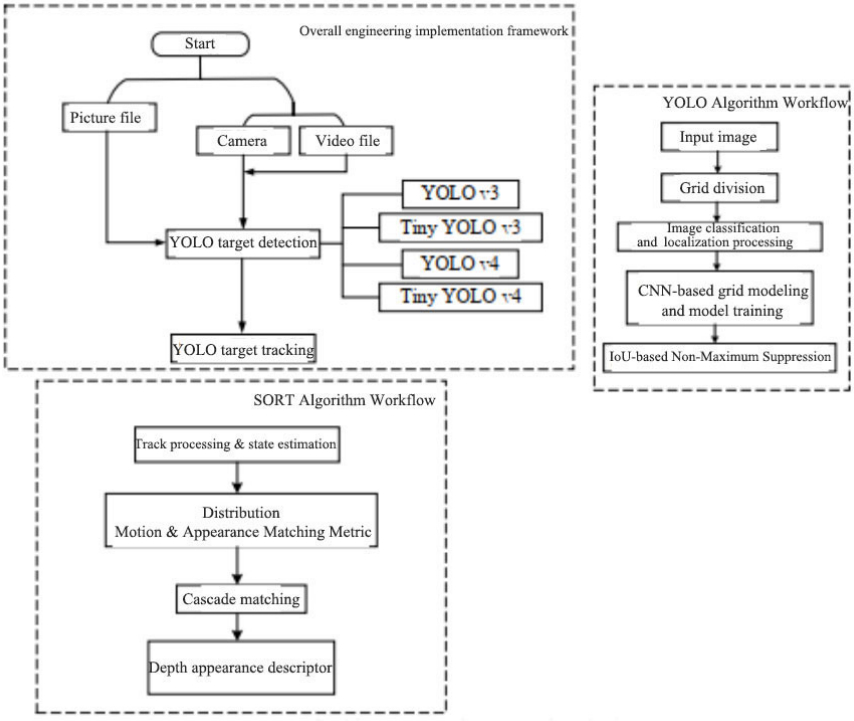

Figure 2. Algorithm Workflow for target detection

After target detection, it is essential to track the target. The author of the SORT algorithm paper employs the traditional Hungarian algorithm for matching detections. The primary methods utilized can be briefly summarized in the following three points:

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202507_11(7).0004

a) A fusion metric was employed to assess the degree of correspondence between detections and tracks. This metric encompasses the distance between predicted and observed positions in Kalman filtering within the martenspace, as well as the cosine distance of the apparent features between bounding boxes.

b) The apparent features of the bounding box consist of 128-dimensional attributes derived from a deep neural network.

c) In the Hungarian algorithm for matching detections and tracks, cascade matching is employed. It is important to note that cascade matching is not inherently superior to global assignment; rather, the limitations of the Kalman filter utilized in this paper to compute motion similarity result in improved outcomes when using cascade matching.

The algorithm starts with the following main steps:

a) Track handling and state estimation

State estimation: An 8-dimensional space is utilized to represent the state of the track at a specific moment in time $(u,v,\gamma,h,\dot{x},\dot{y},\dot{\gamma},\dot{h})$. This representation includes the position of the center of the bounding box, the aspect ratio, the height, and the corresponding velocity information in image coordinates. The update tracks are predicted using a Kalman filter that employs a homogeneous model and a linear observation model. The observation variables are (u,v,γ,h).

Track handling: First, for each track, there is a threshold value which is used to record the time elapsed from the last successful match to the current moment of the trajectory. When this value exceeds a predetermined threshold Amax, the track is considered terminated. Intuitively, this means that a track that cannot be matched for an extended period is considered to be terminated. Then, during the matching process, any detections that do not match successfully are considered to potentially generate new tracks. However, since these detections may be false alarms, the newly generated tracks are initially labeled as then observe whether the matching is successful in the subsequent frames. If the matching is successful, the track is considered new and labeled as confirmed, otherwise, it is considered as a false track, and the state is labeled as deleted.

b) Distribution

A match refers to a correspondence between currently active tracks and current detections. Active tracks are defined as those still being tracked, i.e., tracks with either tentative or confirmed status. The degree of matching between track and detection combines both motion and appearance information. The motion-based similarity is measured using the Mahalanobis distance between the detection and the track's Kalman filter-predicted position.

c) Cascade matching

When a track experiences prolonged occlusion, the continuous predictions of the Kalman filter lead to probability dispersion in its state estimates. In situations where two tracks compete for the same detection, the track with a longer occlusion duration often produces a smaller Mahalanobis distance. As a result, the detection is more likely to be assigned to the track that has been lost for a longer period. However, intuitively, the detection should be assigned to the track that is closest in time. The phenomenon occurs due to the dispersion of probabilities, which results from the inability to update the continuous predictions of the Kalman filter. Conceptually, the covariance matrix can be viewed as representing a normal distribution. In the absence of updates, the variance of this distribution progressively increases with continuous predictions. As a result, points that are farther away in Euclidean distance under the newly inflated distribution may yield the same Mahalanobis distance value as closer points did in the original distribution. This is precisely why our method introduces a cascade matching strategy: to prioritize more frequently seen objects. By considering only tracks with similar occlusion

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202507_11(7).0004

durations during each matching stage, we effectively eliminate the aforementioned association ambiguity.

3. RESULTS

Firstly, the experimental results from the initial phase of target detection are presented. As shown in the QT interface in Fig. 3, the total number of detected targets is 8, which includes both person and bicycles. The confidence level is 0.982 and the time taken for this process is 0.06 seconds.

Figure 3. Target detection for picture file 1

Immediately, we replaced all targets with bicycle-4 and observed that the confidence level is 0.93, with a time taken of 0.06 seconds. The target locations are also shown in Fig. 4.

Figure 4. The corresponding situation when the detection target is bicycle-4

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202507_11(7).0004

Similarly, replacing the detection target with person-5 provides information on his execution difficulty and target location. The confidence level is 0.77, and the time taken is 0.06 seconds, as shown in Fig. 5.

Figure 5. The corresponding situation when the detection target is person-5

Next, we transition to a different graph to assess whether we can still achieve target detection. Figure 6 indicates that the total number of detected targets is 9, with a confidence level of 0.99 and a time of 0.06 seconds, as shown in Fig. 6.

Figure 6. Target detection for picture files

4. CONCLUSION

Performing visual target detection and tracking in complex and dynamic environments with multiple targets is quite challenging. To achieve our objectives, we divide the task into two components: target detection and target tracking. When processing image files, we concentrate on target detection; however, for video streams or video files captured by a camera, we execute both target detection and tracking. This distribution of responsibilities is highly effective,

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202507_11(7).0004

allowing us to efficiently and accurately process various types of information carriers, which significantly enhances overall work efficiency.

In the video detection phase, we employ the YOLO algorithm, which is widely recognized for its effectiveness. This algorithm transforms the target detection process into a regression problem and utilizes a convolutional neural network to analyze the entire image. During the video tracking phase, we implement the SORT algorithm for target tracking. Finally, we use QT to develop a visualization interface, enhancing the clarity of the overall detection and tracking process.

REFERENCES

[1] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3645-3649. DOI: 10.1109/ICIP.2017.8296962

[2] Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. DOI:10.48550/arXiv.2004.10934

[3] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. arXiv e-prints, 2018. DOI:10.48550/arXiv.1804.02767.

[4] Zhang Y, Shen YL, Zhang J. An improved tiny-yolov3 pedestrian detection algorithm. Optik, 2019, 183, 17-23. https://doi.org/10.1016/j.ijleo.2019.02.038

[5] Bewley Al, Ge ZY, Ott L, Ramos F, Upcroft B. Simple Online and Realtime Tracking. 2016 IEEE International Conference on Image Processing (ICIP), 2016. DOI: 10.1109/ICIP.2016.7533003

[6] Dong WS, Mao XX, Fu DX. Target pose estimation based on YOLO model. Computer Measurement & Control, 2025.

[7] He QW, Zhang XX. Research on road traffic sign recognition under complex conditions. Electronic Science and Technology, 2025, 1-11. https://doi.org/10.16180/j.cnki.issn1007-7820.2026.04.002.

[8] Qiao SC, Zhao CY, Bai MY, Dang SS, Pan CY, Zhang MY. Research progress of crop disease detection based on YOLO Algorithms. Journal of Inner Mongolia Agricultural University (Natural Science Edition), 2025, 1-12.

[9] Liu Y, Ren XH, Liu BD, Liu WF. Detection method for small targets based on LDF-YOLO. Electronic Measurement Technology, 2025, 1-10.