applied sciences

MDPI

Article

A Target Re-Identification Method Based on Shot Boundary Object Detection for Single Object Tracking

Bingchen Miao [1], Zengzhao Chen [1,2,3], Hai Liu [1,2] and Aijun Zhang [4,*]

[1] Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China
[2] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China
[3] National Intelligent Society Governance Experiment Base (Education), Central China Normal University, Wuhan 430079, China
[4] China Telecom Corporation Henan Branch, Zhengzhou 450016, China
[*] Correspondence: zhangaijunha@163.com

Abstract: With the advantages of simple model structure and performance-speed balance, the single object tracking (SOT) model based on a Transformer has become a hot topic in the current object tracking field. However, the tracking errors caused by the target leaving the shot, namely the target out-of-view, are more likely to occur in videos than we imagine. To address this issue, we proposed a target re-identification method for SOT called TRTrack. First, we built a bipartite matching model of candidate tracklets and neighbor tracklets optimized by the Hopcroft–Karp algorithm, which is used for preliminary tracking and judging the target leaves the shot. It achieves 76.3% mAO on the tracking benchmark Generic Object Tracking-10k (GOT-10k). Then, we introduced the alpha-IoU loss function in YOLOv5-DeepSORT to detect the shot boundary objects and attained 38.62% mAP$_{75:95}$ on Microsoft Common Objects in Context 2017 (MS COCO 2017). Eventually, we designed a backtracking identification module in TRTrack to re-identify the target. Experimental results confirmed the effectiveness of our method, which is superior to most of the state-of-the-art models.

Keywords: target re-identification; single object tracking; object detection; YOLO; DeepSORT

check for updates

Citation: Miao, B.; Chen, Z.; Liu, H.; Zhang, A. A Target Re-Identification Method Based on Shot Boundary Object Detection for Single Object Tracking. *Appl. Sci.* **2023**, *13*, 6422. https://doi.org/10.3390/app13116422

Academic Editors: Shunli Zhang, Xin Yu, Kaihua Zhang and Yang Yang

Received: 8 May 2023
Revised: 20 May 2023
Accepted: 23 May 2023
Published: 24 May 2023

1. Introduction

Visual single object tracking (SOT) is a basic computer vision task that refers to the detection, extraction, recognition, and tracking of the single object selected in the video sequence, which can realize an understanding of the object's behavior [1]. According to the object specified in the first frame in the video, the SOT network predicts the location and size of the object in subsequent frames. Visual SOT has a wide range of application prospects in aspects such as video surveillance [2], automatic driving [3], and human–computer interaction [4]. It is also the foundation for other computer vision tasks [5].

With the development of SOT, many practical and efficient algorithms have emerged [6]. One type of SOT model is based on correlation filters, such as KCF [7], SRDCF [8], and ECO [9]. This kind of method's idea is that the cyclic convolution operation is completed by a fast Fourier transform through the obtained correlation filter in the next frame to realize the positioning of the target center point [10]. Another type of model is based on the Siamese network, which is designed to transform the object tracking task into a feature-matching task between a given template image and candidate images [11]. Models such as SiamFC [12], DaSiamRPN [13], and SiamAttn [14] are implemented based on this idea. In recent years, other types of Transformer [15] based models, such as RGB-T [16], STARK [17], MixFormer [18], and OSTrack [19], have become one of the most popular SOT methods due to their advantages of simple model structure and performance–speed balance. Another fundamental task in computer vision, object detection, is closely related to object tracking. Object tracking networks are often used to track detected objects for long-term object detection [20]. Current object detectors based on deep learning have become mainstream models, mainly divided into two-stage detectors and single-stage detectors. The former uses a proposal mechanism to obtain image regions that are more likely to contain objects in order to achieve target prediction, mainly including R-CNN [21], Fast R-CNN [22], and Faster R-CNN [23]. The latter directly detects objects from anchors after extracting input image features, mainly including SSD [24], YOLO series [25], and DETR [26]. For example, the detectors of the YOLO series directly pre-define anchors with multiple scales at all locations of the image and predict the category probability and confidence score, greatly improving the speed of object detection [27].

However, there are some challenges that have been difficult to solve for SOT models, mainly including target occlusion, out-of-view, deformation, and scale variation. These challenges can easily result in an inability to continue tracking or result in tracking errors. Figure 1 shows some examples in the tracking benchmark dataset GOT-10k [28].

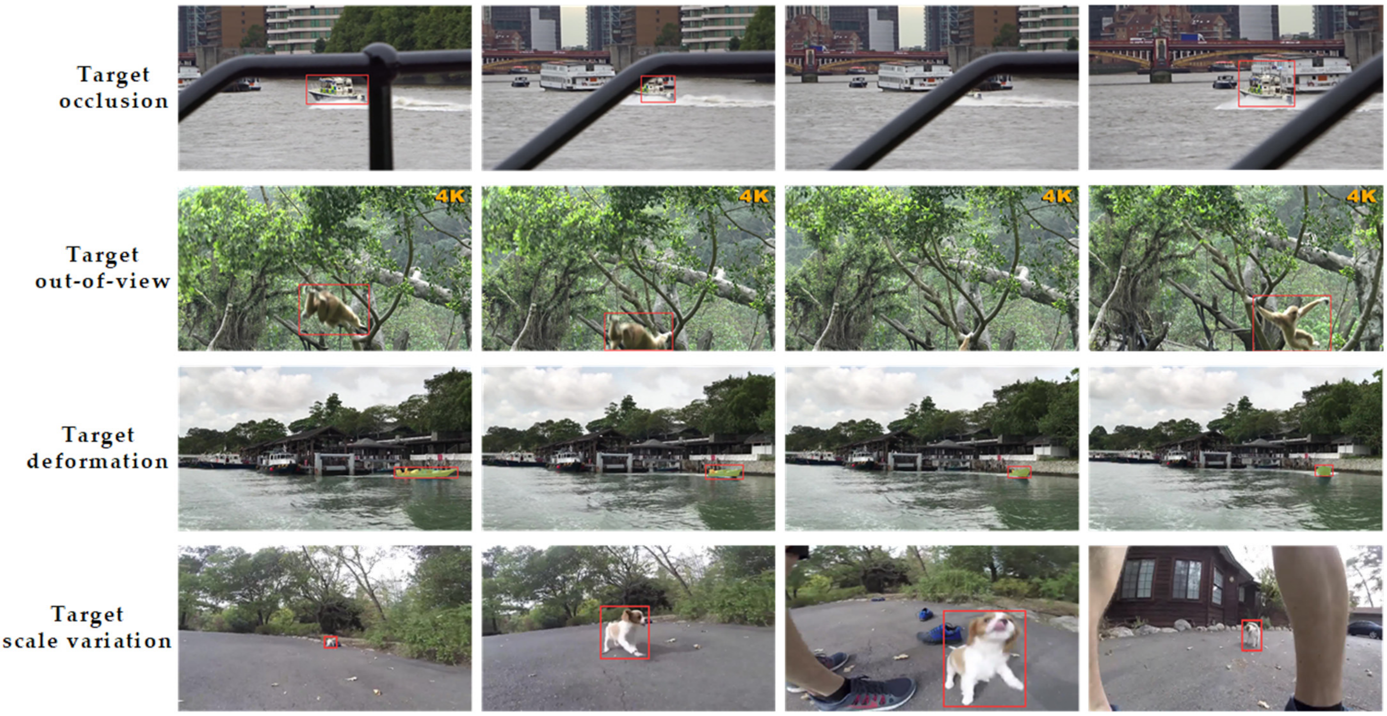


Figure 1. Example of SOT challenges in benchmark dataset GOT-10k.

In fact, not only in SOT but also in other tasks of computer vision, such as multi-object tracking (MOT), object detection, and person re-identification (Re-ID), these four major challenges are urgent issues to be addressed. The specific descriptions of these challenges are as follows:

(1) Target occlusion. Due to the complex environment in videos, when the tracked targets are moving or when other objects in the shots are moving, the tracked targets may be obscured by other objects for a period of time. In Figure 1, the moving ship is obstructed by the railing in front of the camera.

(2) Target out-of-view. This situation often occurs in videos where the tracked targets are moving rapidly, regardless of whether the shots are fixed or moving. Because the targets are generally not aware of the range of the shots, as they move, it may cause them to leave the cameras. The monkey's rapid movement causes it to leave the shot in Figure 1.

(3) Target deformation. Tracked targets with irregular shapes and sizes or actively changing their own appearance can cause this situation, especially when they move. In

Figure 1, the small boat changes its plane shape in the shots due to its rotational motion.

(4) Target scale variation. This situation occurs when the tracked targets approach or move away from the shots. The small dog's running back and forth in front of the camera changes its scale within the lens in Figure 1.

Even SOT models based on Transformers find it difficult to cope with these challenges, resulting in tracking errors. The target out-of-view in these challenges, which means that the object leaves the shot and reappears, is more likely to appear in videos than we imagine, making it harder for SOT models to be well applied to real life. Through our analysis, the main factors may be as follows:

(1) There is no uniform standard for videos, which are shot from different perspectives, and it is common for objects to move in and out of view. Videos, especially daily videos, are often casually shot, capturing scenes that are not the entire actual space but a limited fan-shaped area. At the same time, due to the ease of interaction between objects in videos and the uncertainty of their behavior, it is easy to cause objects to leave the shot.

(2) The SOT algorithms are generally weak in the re-identification of objects out of shots [29]. Most SOT models focus on enhancing the performance of tracking by improving the ability of feature extraction and relation modeling [19]. However, when objects are obscured or change their appearance, the effective features of the object itself are extremely limited [30]. After the target leaves the shot, the model will choose the object with the highest confidence score in the current frame for long-term tracking, causing an inability to achieve target re-identification in a short period of time.

Figure 2 shows the two possible reasons for poor tracking performance.

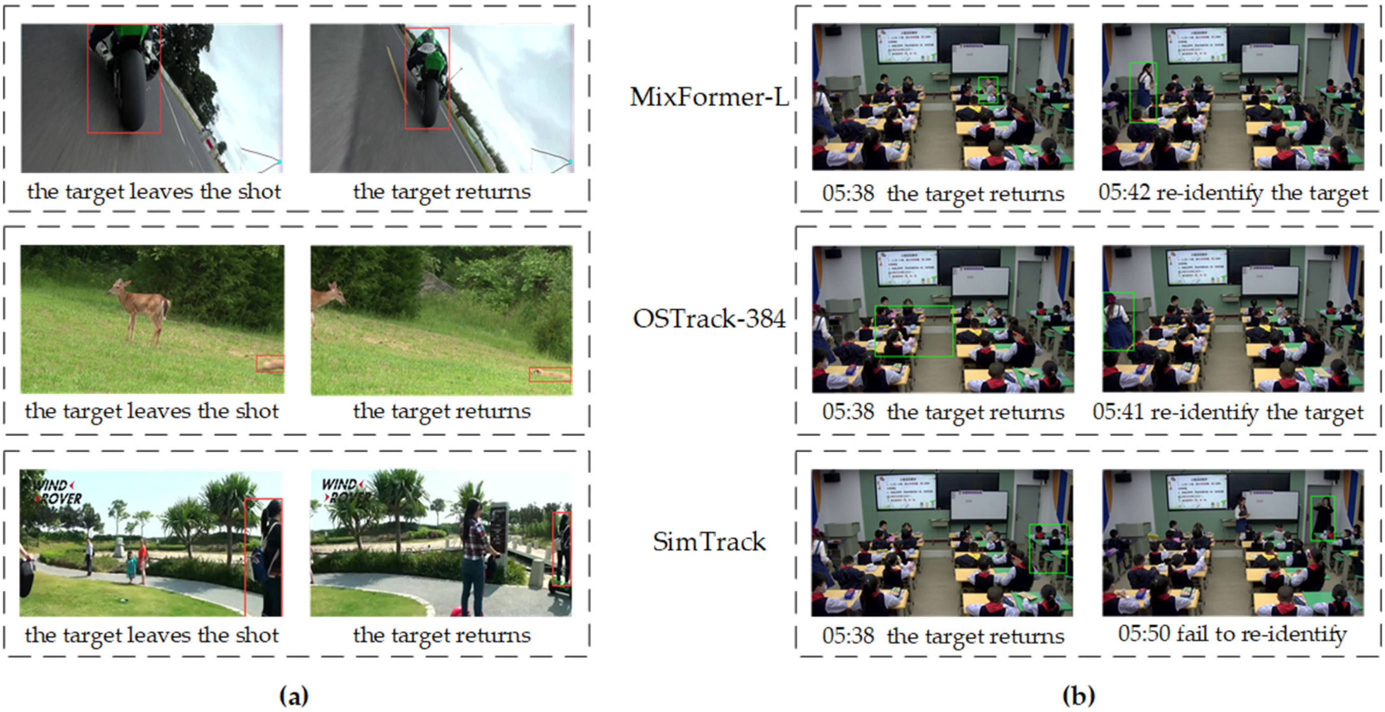


Figure 2. Factors of unsatisfactory tracking results. (**a**) Targets' frequently moving in and out of the shots. (**b**) The current SOT algorithms are generally weak in re-identification of the target after it leaves the shot.

In attempting to address the problem of poor tracking results of target out-of-view, this paper proposes a target re-identification method based on shot boundary object detection for single object tracking, called TRTrack. Specifically, we build a bipartite matching model of candidate tracklets and neighbor tracklets optimized by the Hopcroft–Karp algorithm, which is used for preliminary tracking and judging the target leaves the shot. Then, TRTrack carries out object detection at the shot boundary by improved YOLOv5-DeepSORT, in which the original loss function is substituted with the alpha-IoU. The backtracking identification module in TRTrack crops the detected objects and inputs them back into the tracking model as the search region image. Finally, the target with high confidence score will be re-identified.

The contributions of this paper are shown below:

- To deal with the problem of unsatisfactory tracking results resulting from target out-of-view, we propose a target re-identification method based on shot boundary object detection for single object tracking, called TRTrack;
- We build a bipartite matching model of candidate tracklets and neighbor tracklets optimized by the Hopcroft–Karp algorithm to judge the target leaves the shot and introduce the alpha-IoU loss function to YOLOv5-DeepSORT to enhance object detection capability;
- Through a wide range of experiments by self-built videos dataset CLV and benchmark dataset, TRTrack is verified to be applied well for target re-identification in most video tracking tasks.

The rest of this paper is structured as follows: Section 2 presents the related work, including disappearing objects re-identification methods, a single object tracking model based on Transformer, and an object detection model following a top-down approach. Section 3 describes the proposed methodology of the target re-identification method TRTrack with a preliminary object tracking module, boundary object detection module, and backtracking identification module. Section 4 shows the detailed experiment and results. Section 5 concludes our work.

2. Related Work

2.1. Methods of Disappearing Target Re-Identification

The target re-identification mentioned in this paper is fundamentally different from a major task of computer vision, namely, person re-identification (Re-ID). Re-ID is geared to recognize the same person through videos obtained from different cameras [31]. It can be categorized into person detection, person tracking, and person retrieval [32]. The introduction of infrared technology has become a new development hotspot for Re-ID, and people have also begun to attach importance to infrared technology in other fields [33–35]. By contrast, the target re-identification in this paper only refers to the identification of targets that leaves the shot and reappears in SOT tasks.

Target out-of-view is essentially a special case of target disappearing, for which there are various algorithms that strive to resolve, especially for another special case, target occlusion. In fact, the target out-of-view is essentially the same as the target being obscured by the shot boundary.

The LMCF [36] model proposed by Wang et al. determines whether the object is obscured or disappears by observing the change degree of self-created metric APCE. Liu et al. [37] designed the BM Net with a multi-stream convolutional-LSTM network, which predicts the position of the target in subsequent frames based on its past trajectory. The Siam R-CNN [38] model uses the mechanism of object redetection to input the nearest object into the redetection network to judge and retrieve the object. Chen et al. [30] created the NeighborTrack, which uses the confidence score output from the object tracking model to automatically utilize the information of neighbor regions that are not obscured to re-identify the tracking target. For multi-object tracking, Ahn et al. [39] introduced an attention-based re-identification model, which extracts feature vectors from images to correlate objects based on their appearance effectively.

However, most of these methods rarely use the Transformer [15], leading to their lack of tracking performance and computation speed. They aim more at reducing the tracking errors caused by the object being obscured in the shots instead of re-identifying the object that leaves the shot and reappears.

2.2. Single Object Tracking Model Based on Transformer

Transformer stands out in object tracking after Dosovitskiy et al. [40] formally introduced the Transformer model into computer vision, especially in the field of human pose estimation [41] or head posed estimation [42].

SOT models based on Transformers can be divided into two- and one-stream frameworks [19]. The basic difference lies in how to carry out feature extraction and relation modeling on the template image and the search region image:

(1) Two-stream framework. This framework first inputs the template and the search region into the backbone of the model and shares the weight. It then concatenates the output results to feed into the Transformer. Finally, the location of the object is predicted by classification, regression, and other methods. In recent years, Chen et al. [43] proposed TransT, which fuses iterative features through stacked self-concern layers and cross-concern layers. The STARK [21] model implemented by Yan et al. connects a new template with the search region in a way that automatically updates template images. Lin et al. [44] proposed SwinTrack based on the total attention mechanism instead of using CNN and other neural networks. These algorithms have satisfactory tracking accuracy, but their inference efficiency is not very fast because of heavy relation modeling.

(2) One-stream framework. In this framework, the template and the search region are concatenated before they are input into the backbone, and the subsequent process is similar to that of the two-stream framework. A typical example is the MixFormer [18] developed by Cui et al., which introduces a mixed attention module to build an information interaction channel between the template-search image pairs. The OSTrack [19] proposed by Ye et al. connects the template and the search region to bidirectional information flows to combine feature learning with interaction. Chen et al. [45] built the SimTrack, which is a simplified tracking model using the Transformer as a backbone for relation modeling and feature extraction. These algorithms achieve not only high tracking accuracy but also fast inference speed, thereby balancing between performance and speed.

However, regardless of the two-stream framework or one-stream framework, most single object tracking methods based on Transformer focuses on enhancing the performance by improving the relation modeling and feature extraction capabilities and lack targeted solutions to common object tracking cases such as object occlusion and deformation [29]. When the object is obscured or changes its appearance, the effective features of the object itself are extremely limited [30]. Target out-of-view is a special case of target disappearing, so it is difficult to re-identify the target in a short time when it reappears.

2.3. Object Detection Method Following the Top-Down Approach

Object detection is a basic task in computer vision, with the purpose of identifying categories and predicting the position of objects in image sequences. It is widely used in fields such as pedestrian recognition [46], autonomous driving [47], and crop planting [48].

Object detection algorithms can be separated into top-down and bottom-up methods, with the main distinction being the period of holistic object generation and evaluation. The top-down approach is still the most commonly used method nowadays, including two-stage and one-stage methods:

(1) Two-stage method. This method uses a proposal mechanism to decrease negative candidates generated by anchors and outputs the object detection results consisting of prediction bounding boxes and corresponding probabilities of object category through the detection network, such as CNNs. R-CNN [21], proposed by Donahue et al., combines region proposals with CNNs and can predict and partition objects by applying high-capacity convolutional neural networks. Girshick [22] designed the Fast

R-CNN object detection model, which uses deep convolutional networks to classify object proposals efficiently; R-FCN [49], developed by Dai et al., contains position-sensitive score maps to resolve the contradiction between image classification and object detection. However, this method often results in a long training time and slow testing speed due to a large amount of repeated computation of convolutional features.

(2) One-stage method. This method is created to realize object detection directly from anchors after extracting the input image features without using any proposal elimination mechanisms. Wei et al. [24] presented the SSD, which removes the generating proposal and image feature extraction modules and implements all the work of the model into a single network. RetinaNet [50] is designed to address imbalance problems of object detection by optimizing the standard cross entropy loss. Redmon et al. [25] converted the object detection task into a regression problem of object bounding boxes and corresponding category probabilities while proposing a single object detection network, YOLO, which has attained optimal accuracy and speed performance. This type of method can optimize the detection performance from end to end and has very high computation efficiency.

Regardless of whether a two- or a one-stage method is used, inherent problems arise in the object detection model, such as frequent ID switching of the detected object may result from occlusions. Therefore, extra models such as DeepSORT [20] can be used to improve object detection results. DeepSORT introduces deep learning into the SORT [51] algorithm and reduces identity switching by adding appearance descriptors.

3. Proposed Method

The overall architecture of TRTrack, a target re-identification method based on shot boundary object detection for single object tracking, is shown in Figure 3.

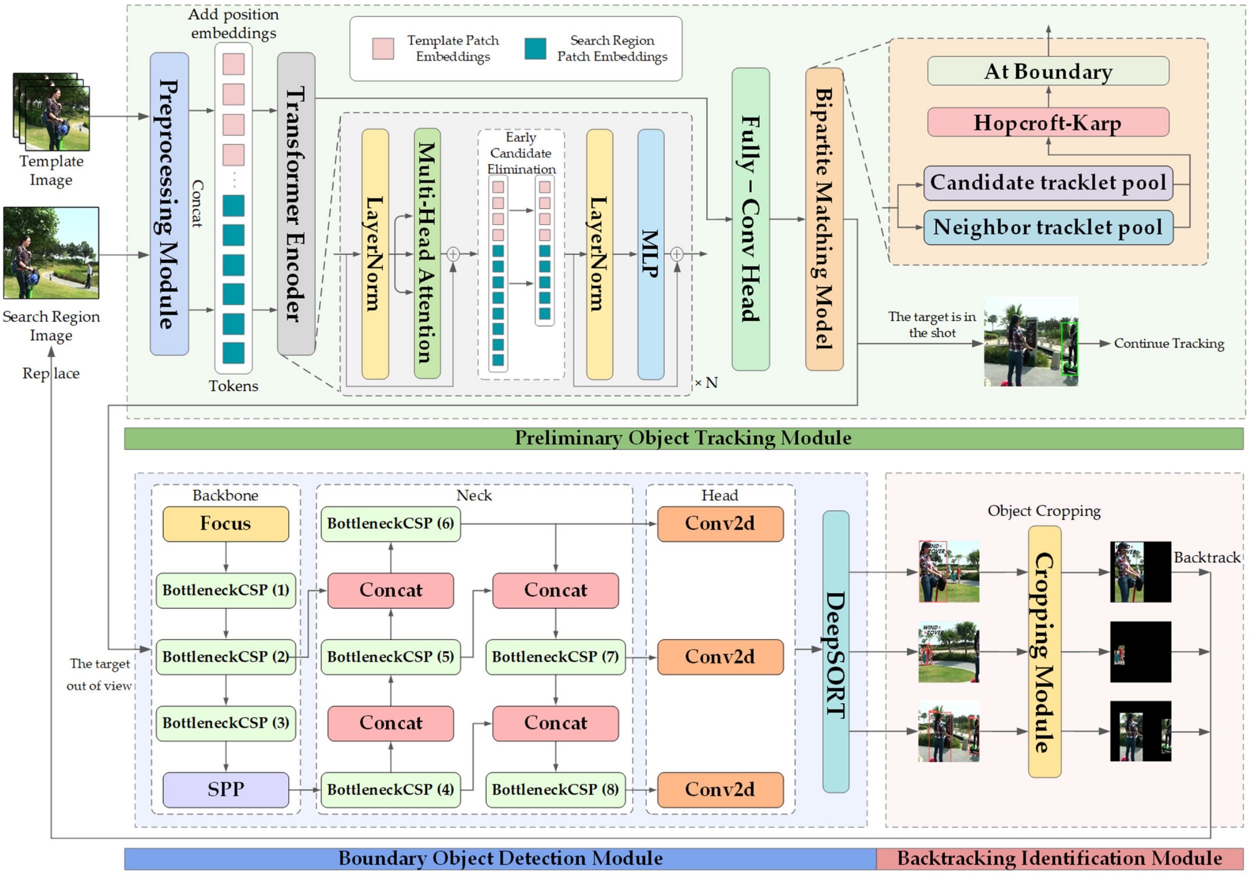


Figure 3. Overall architecture of TRTrack model. It consists of three modules: preliminary object tracking,

boundary object detection, and backtracking identification. First, for any frame I_T with template image patch $z \in R^{3\times H_z \times W_z}$ and search region image patch $x \in R^{3\times H_x \times W_x}$, TRTrack inputs them into the preliminary object tracking module for preliminary tracking. Then, the tracking results from the head will be optimized by the bipartite matching model, which is introduced the Hopcroft–Karp algorithm to judge whether the target leaves the shot. If not, the tracking will continue; otherwise, tracking will be stopped. In the latter case, the search region image patch x of the frame I_T will be input into the boundary object detection module to detect the objects within a certain shot boundary range. Finally, in the backtracking identification module, the detected objects will be cropped out and used to replace the original search region in the next frame and input back into the tracking model. Eventually, TRTrack judges whether the target reappears according to the confidence score S^t from object tracking head and realizes the rapid target re-identification.

3.1. Preliminary Object Tracking Module

In the preliminary object tracking module, the base structure of OSTrack-384 [19] with powerful SOT performance-speed balance perfectly fits our goal, which is selected as the main component of the module. The post-processor NeighborTrack [30] is used to optimize the preliminary tracking results. It is a method that uses the confidence score output from the backbone of an object tracking model to automatically derive the information of neighbor regions that are not obscured.

The bipartite matching model in TRTrack maintains both the candidate pool P^c and the neighbor pool P^n. The former is used to select the most suitable prediction tracking object, and the latter is used to verify the selected prediction tracking object. It converts the association problem between the template and search region in SOT into the bipartite matching problem between the candidate tracklets $S^c = P^c$ and the neighbor tracklets $S^n = P^n \cup \{\eta\}$, where $\eta = \{b_{t-1}, \ldots, b_{t-\tau}\}$ represents the real prediction tracking boxes, t is the sequence number of frames at any time, and τ is the number of frames to be traced. The weight w_{ij} on the edge between two nodes, $\xi_i^t \in S^c$ and $\zeta_j^t \in S^n$, represents the average Jaccard overlap calculation between the two tracklets. This means that if $\xi_i^t = \left(b_{t-1}^c, \ldots, b_{t-\tau}^c\right)$ and $\zeta_j^t = \left(b_{t-1}^n, \ldots, b_{t-\tau}^n\right)$, the formula of the weight is denoted as

$$w_{ij} = \frac{1}{\tau} \sum_{k=t-\tau}^{t-1} IoU(b_k^c, b_k^n) \tag{1}$$

where IoU represents the Jaccard overlap calculation between the two prediction bounding boxes, which reflects the similarity between the candidate tracklets S^c and the neighbor tracklets S^n.

It is worth noting that the Hopcroft–Karp algorithm, which can achieve the maximum matching of the bipartite graph and has high computation efficiency, is introduced to implement bipartite matching. The time complexity of it is $O = \left(|E|\sqrt{|V|}\right)$, where $|E|$ is the number of edges of the bipartite graph and $|V|$ is the number of vertices in the bipartite graph. The specific formula of the improved bipartite matching result is as follows

$$mat_t = HK(S^c, S^n) \tag{2}$$

where mat_t indicates the bipartite matching result of the I_t frame. If S^c does not match S^n, which means the candidate tracklet ξ_m^t does not match the target tracklet η, it indicates that the target object is obscured or disappears. A function at_bdy is added to judge whether the prediction bounding box b_t of the current frame I_t is at the shot boundary. The equation is as follows

$$\begin{aligned} at\_bdy(b_t) = \left(x \in \left(0, \tfrac{W}{13}\right)\right) \cap \left(y \in \left(0, \tfrac{H}{13}\right)\right) \cap \left(x + w \in \left(\tfrac{12W}{13}, W\right)\right) \\ \cap \left(y + h \in \left(\tfrac{12H}{13}, H\right)\right) \end{aligned} \tag{3}$$

where $b_t = (x, y, w, h)$ are the bounding box containing abscissas and ordinates of the upper left corner and the size, W is the width of the video, and H is the length of the video. The final judgment of whether the target leaves the shot is shown as

$$out_t = \widetilde{mat_t} \cap at\_bdy(b_t) \tag{4}$$

where $\widetilde{mat_t}$ represents the inverse value of mat_t, and $out_t \in \{0, 1\}$ is the judgment variable of whether the target leaves the shot in the frame I_t. If the value is 1, the target has left the shot. Conversely, a value of 0 means that the target is still in the shot.

Figure 4 shows the feature maps obtained in the tracking network of the preliminary tracking module in TRTrack. The color in the feature maps reflects the similarity estimation between each position of the search region image and the target, and the greener the color indicates that the position of the current image is more likely to be the target.

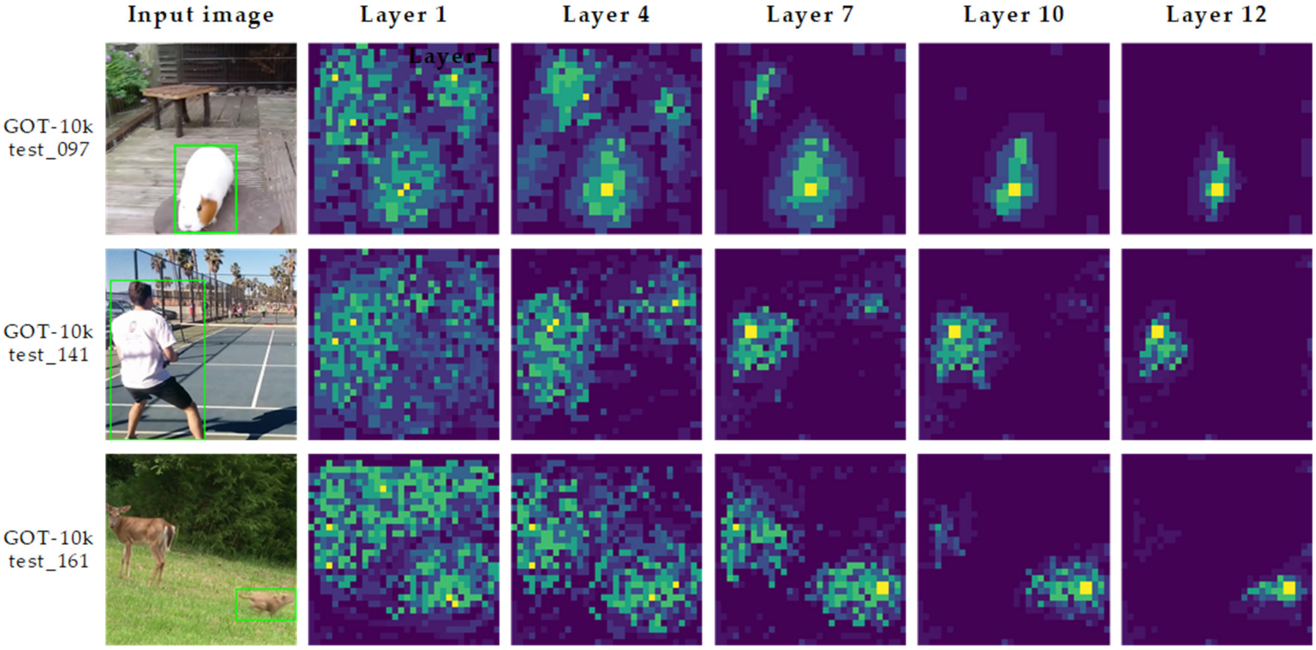


Figure 4. Feature maps obtained in the tracking network of the preliminary tracking module. It can be seen that as the number of network layers increases, the feature map becomes more and more focused on the target.

3.2. Boundary Object Detection Module

In videos, the positions of the out-of-view targets returning from outside to inside the shots are uncertain, meaning that the target can reappear from almost any location at the shot boundary. Therefore, the YOLOv5-DeepSORT model, with excellent accuracy of object detection and inference speed, becomes a suitable tool for detecting the target in the boundary object detection module.

He et al. [52] utilized the power transformations to existing IoU loss functions, such as GIoU and CIoU, to develop a new IoU loss function, alpha-IoU. The α in the alpha-IoU represents a power parameter that provides greater flexibility for the detector to achieve different bounding box regression accuracy. Abundant experimental results reveal that the alpha-IoU can realize more accurate object detection by weighting the gradient and loss of objects with high IoU values in object detection models [53].

The original loss function of YOLOv5 is replaced by the alpha-IoU in the boundary object detection module of TRTrack to enhance the object detection accuracy and flexibility of the boundary object detection module. Based on the idea of alpha-IoU, we set the loss function only takes effect when the parameter α is greater than 1. The formula is as follows

$$L_{\alpha-IoU} = \frac{1 - IoU^{\alpha}}{\alpha},\ \alpha > 1 \tag{5}$$

where IoU means the original Jaccard overlap calculation representing the intersection ratio of the prediction bounding box and the ground truth. In this case, the whole loss value of YOLOv5 in TRTrack is as follows

$$L_{total} = L_{conf} + L_{class} + {L_{\alpha-IoU}}' \tag{6}$$

where L_{conf}, L_{class}, and ${L_{\alpha-IoU}}'$ indicate the confidence loss, the classification loss, and the actual alpha-IoU loss, respectively. The specific alpha-IoU calculation equation is shown as

$${L_{\alpha-IoU}}' = \sum_{i=0}^{S^2} \sum_{j=0}^{B} (1 - L_{\alpha-IoU}) \tag{7}$$

where S^2 represents the number of image grids, and B is the number of every grid anchor box in object detection.

On this basis, after the method in Section 3.1 is used to judge whether the target leaves the shot, the object detection is conducted at the shot boundary, and the detection range function is at_bdy mentioned in Section 3.1. Then, the results of object detection are shown as

$$det = \{d_1, \ldots, d_n\} \ \in \ at\_bdy(b_d) \tag{8}$$

which represents a collection of detection bounding boxes for objects at the shot boundary, and $b_d = (x, y, w, h)$ are the whole object detection bounding boxes.

3.3. Backtracking Identification Module

By taking advantage of the peculiarity of the SOT model in TRTrack that it transforms object tracking tasks into a feature extraction and matching problem between the template image and the search region image, we design the backtracking identification module, which is implemented on the object detection results $det = \{d_1, \ldots, d_n\}$.

For the objects detected at the boundary of the shot, TRTrack crops them out by setting the pixel values in the frame except for the region of objects as 0 or 255. That is, only the detected objects are retained in the current frame. The formula is as follows

$$crop = \sum_{k=1}^{n} d_k,\ d_k \in det \tag{9}$$

Then, the cropped image $crop$ is input back to the SOT model and replaces the current frame image as the new search region image for tracking. Then the confidence score S, corresponding to each current prediction tracking bounding box B, will be output from the head of the tracking model. The confidence score S can be used to judge whether the target returns to the shot. The formula for judging results is shown as

$$in_t = \begin{cases} 1, & S \geq \lambda \\ 0, & S < \lambda \end{cases} \tag{10}$$

where $in_t \in \{0, 1\}$ is the judgment variable of whether the target in the frame I_t has returned to the shot, and λ is the identification threshold parameter, which is set to 0.56 in this paper. Figure 5 shows some examples of using the backtracking identification module.

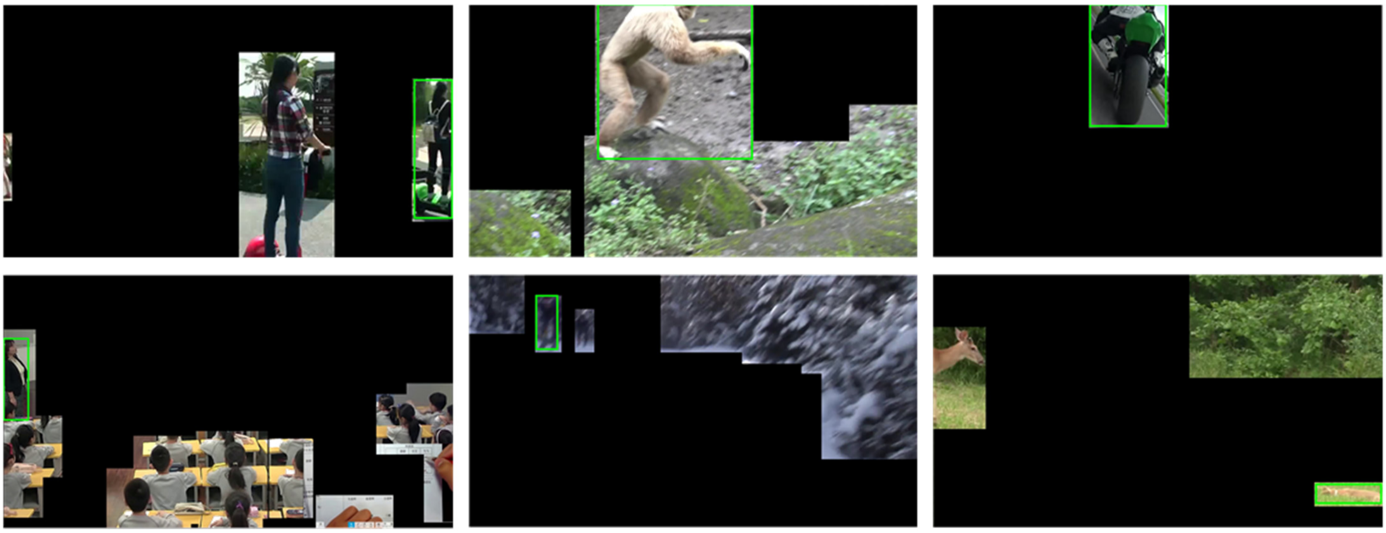


Figure 5. Effectiveness of the backtracking identification module. It shows the visual effect of using the object tracking model to track images of cropped shot boundary objects. It can be seen that the tracker will accurately track the target rather than other objects.

A flow-process diagram of the backtracking identification module is shown in Figure 6 to demonstrate its practical process better.

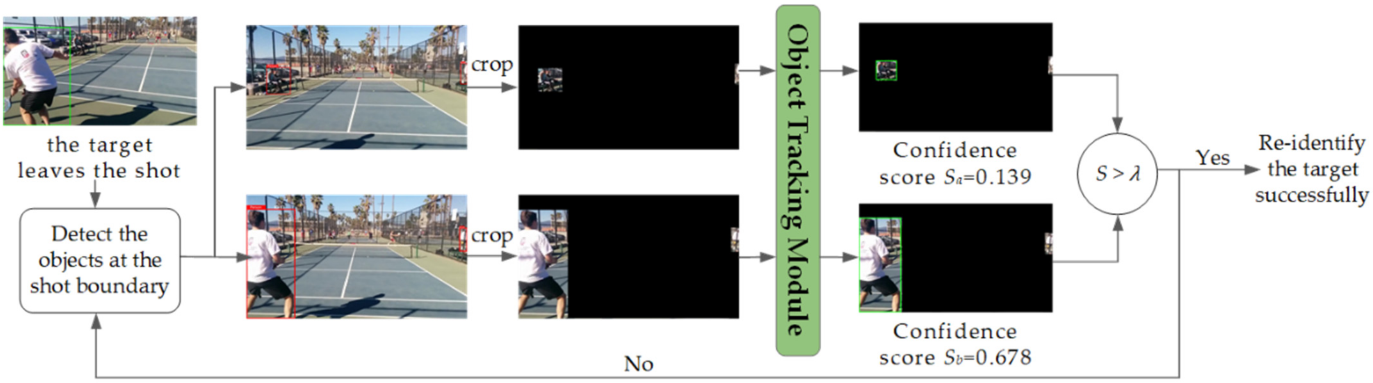

the target leaves the shot
Detect the objects at the shot boundary
crop
crop
Object Tracking Module
Confidence score S_a=0.139
Confidence score S_b=0.678
$S > \lambda$
Yes
Re-identify the target successfully
No

Figure 6. Flow-process diagram of the backtracking identification module. It can be seen that when the target is judged to have left the shot, the backtracking identification module will crop out the objects at the shot boundary and input them back into the object tracking model. The output of the tracking model includes the prediction bounding box and the corresponding confidence score. If the score is greater than the parameter λ, it indicates that the target is re-identified.

4. Experimental Results

The experimental platform is a Windows 11 64-bit system with two Nvidia GeForce RTX 4070Ti graphics cards, 64 GB running memory, Pytorch 1.10.0 deep learning framework, and Cuda 11.3 parallel computing platform, implemented on Python 3.8.

4.1. Experimental Setup

4.1.1. Datasets

We used two mainstream medium and long-term datasets of the SOT field, i.e., LaSOT [54] and GOT-10k [28], to evaluate the performance of the object tracking module in TRTrack. The LaSOT dataset consists of 1400 sequences, totaling over 3.5 million frames. Each sequence of it includes various challenges originating from the wild, such as target occlusion, deformation, etc. We also divided the training set and the test set according to the 4/1 ratio given by LaSOT. The GOT-10k dataset includes over 10,000 video segments with more than 1.5 million manually labeled bounding boxes. It contains a total of over 560 types of moving objects and 87 motion modes. We follow the split of GOT-10k, approximately 9.34 k of training data, and 420 test data in GOT-10k.

The benchmark dataset MS COCO 2017 was selected to evaluate our object detection model. MS COCO 2017 is an object detection dataset built by Microsoft, with a total of 80 object categories. The dataset contains objects of various sizes, some of which are noisy or obstructed, making it challenging. There are over 118 k and 5 k images in the training and testing sets, respectively. We follow this split in experiments.

For target re-identification, considering the target re-identification problem of objects in videos studied in this paper, we found that the classroom videos fit our research needs very well. In classroom videos, because the interaction between teacher and student is crucial in a lecture, the behavior of a teacher leaving the podium to communicate with students can easily lead to the teacher being out of view.

On this basis, we established a classroom video dataset CLV, which is selected from actual classroom videos with prominent classroom video features, such as targets' frequently moving in and out of the shots. It includes 16 main classroom video segments, with more than 6400 manually labeled bounding boxes when the target leaves the shot and reappears. The video information of the dataset is shown in Table 1.

Table 1. Basic information of classroom video dataset CLV.

Video ID	Video Duration	Video Size	Frame Rate	Number of Times the Target Leaves the Shot	Total Time of the Target Leaves the Shot
001	40:20	1920 × 1080	30 fps	11 times	79.674 s
002	50:35	1920 × 1080	30 fps	9 times	179.614 s
003	37:42	1920 × 1080	30 fps	11 times	106.721 s
004	33:51	1920 × 1080	30 fps	9 times	142.153 s
005	43:35	1920 × 1080	30 fps	24 times	332.490 s
006	45:14	1920 × 1080	30 fps	15 times	163.069 s
007	32:04	1920 × 1080	30 fps	3 times	110.833 s
008	41:40	1280 × 960	30 fps	8 times	58.689 s
009	38:26	1920 × 1080	30 fps	7 times	174.110 s
010	32:11	1920 × 1080	30 fps	14 times	88.457 s
011	36:58	1280 × 960	30 fps	6 times	101.776 s
012	40:46	1920 × 1080	30 fps	19 times	254.330 s
013	30:09	1280 × 960	30 fps	4 times	76.541 s
014	31:32	1920 × 1080	30 fps	13 times	94.879 s
015	35:27	1920 × 1080	30 fps	16 times	143.556 s
016	38:43	1920 × 1080	30 fps	12 times	167.123 s

4.1.2. Evaluation Metrics

We tested our preliminary tracking module using the evaluation metrics proposed by GOT-10k and LaSOT. For GOT-10k, including mean average overlap (mAO) and mean success rate (mSR). For LaSOT, including area under the curve (AUC), normalized precision (P_{Norm}), and precision (P). Three average precision (AP) and two mean average precision (mAP) in MS COCO 2017 were used to evaluate our object detection model.

For the self-built dataset CLV, we defined three metrics to evaluate the performance of target re-identification:

- ET

The unit is "times", which represents the number of tracking errors using the SOT model in a video for tracking the target, mainly reflecting the stability of the model.

$$ET = \sum_{k=1}^{n} 1, \quad Te_k \in TE \tag{11}$$

where TE represents the total moments when tracking errors occur, and Te_k means the moment when a tracking error occurs during the k-th tracking error process.

- ED

The unit is "second", which represents the total duration of tracking errors by using the SOT model in a video for tracking the target, mainly reflecting the accuracy of the model.

$$ED = \sum_{k=1}^{n} (Tc_k - Te_k) \tag{12}$$

where Tc_k represents the moment when the tracking error was corrected during the k-th tracking error process.

- TD

The unit is "second", which represents the total duration of the time interval between the target's reappearance after leaving the shot and being tracked again by using the SOT model in a video, mainly reflecting the ability of target re-identification of the model.

$$TD = \sum_{k=1}^{n} (Tr_k - Ta_k) \tag{13}$$

where Tr_k and Ta_k represent the moment when the tracker re-identifies the target and the moment the target reappears in the shot during the k-th target re-identification process, respectively.

4.1.3. Implementation Details

In this study, the OSTrack-384 [19] is selected as the main component of the preliminary object tracking module. The input sizes of templates and search regions are 192 × 192 pixels and 384 × 384 pixels, the initial learning rate was set to 4×10^{-5}, the batch size was set to 128, and the training epoch was defined as 300.

The bipartite matching model in TRTrack based on NeighborTrack [30] is used to optimize the tracking results. For SoftNMS in it, the IoU threshold was set to 0.25, and the time period τ of backtracking tracklets was set to 9.

For YOLOv5-DeepSORT in the boundary object detection module of TRTrack, the YOLOv5n network with balanced performance is chosen as the basic module of object detection, which is introduced in the alpha-IoU loss function. Following the universal approach, the training batch size was valued at 32, the decay rate was set to 5×10^{-4}, and the learning rate was defined as 1×10^{-2}. In particular, the value of parameter α of the alpha-IoU loss function is 3.

4.2. Results and Analysis

4.2.1. Results of Preliminary Object Tracking

As we mentioned in Section 4.1.1, two mainstream medium and long-term SOT benchmark datasets, LaSOT [54] and GOT-10k [28], were selected to evaluate the performance of our preliminary tracking module. LaSOT is a challenging long-term tracking benchmark. GOT-10k adopts a one-shot tracking rule, which requires only training the tracker on its training split, and the object classes between the train and test split do not overlap. We follow this rule to train our model. The comparison of our preliminary tracking module and state-of-the-art models tracking results on two SOT benchmarks, LaSOT and GOT-10k, are concluded in Table 2.

It can be seen that, for LaSOT, compared with the best version of NeighborTrack, namely, NeighboTrack-OSTrack, the metric AUC and P of the preliminary object tracking module in TRTrack are better, which are 0.731 and 0.787, respectively. A similar situation also occurs in the results of GOT-10k, in which the metric mAO and mSR_{75} of ours are higher than other state-of-the-art models, which are 0.763 and 0.739, respectively.

Table 2. Comparison with state-of-the-art models on two SOT benchmarks: LaSOT and GOT-10k.

Model	LaSOT			GOT-10k		
	AUC	**P_{Norm}**	**P**	**mAO**	**mSR_{50}**	**mSR_{75}**
AiATrack	0.690	0.794	0.738	0.696	0.800	0.632
SwinTrack-B-384	0.702	0.784	0.753	0.724	0.805	0.678
MixFormer-L	0.701	0.799	0.763	0.756	**0.8573** *	0.728
OSTrack-384	0.711	0.811	0.776	0.737	0.832	0.708
NeighborTrack-OSTrack	0.722	**0.818**	0.780	0.757	0.8572	0.733
Ours	**0.731**	0.814	**0.787**	**0.763**	0.854	**0.739**

* The bold numbers in the table represents the optimal value in the evaluation metrics.

4.2.2. Results of Object Detection

For our object detection model, the benchmark dataset MS COCO 2017 was selected to test. The object detection results of different loss functions on MS COCO 2017 are demonstrated in Table 3.

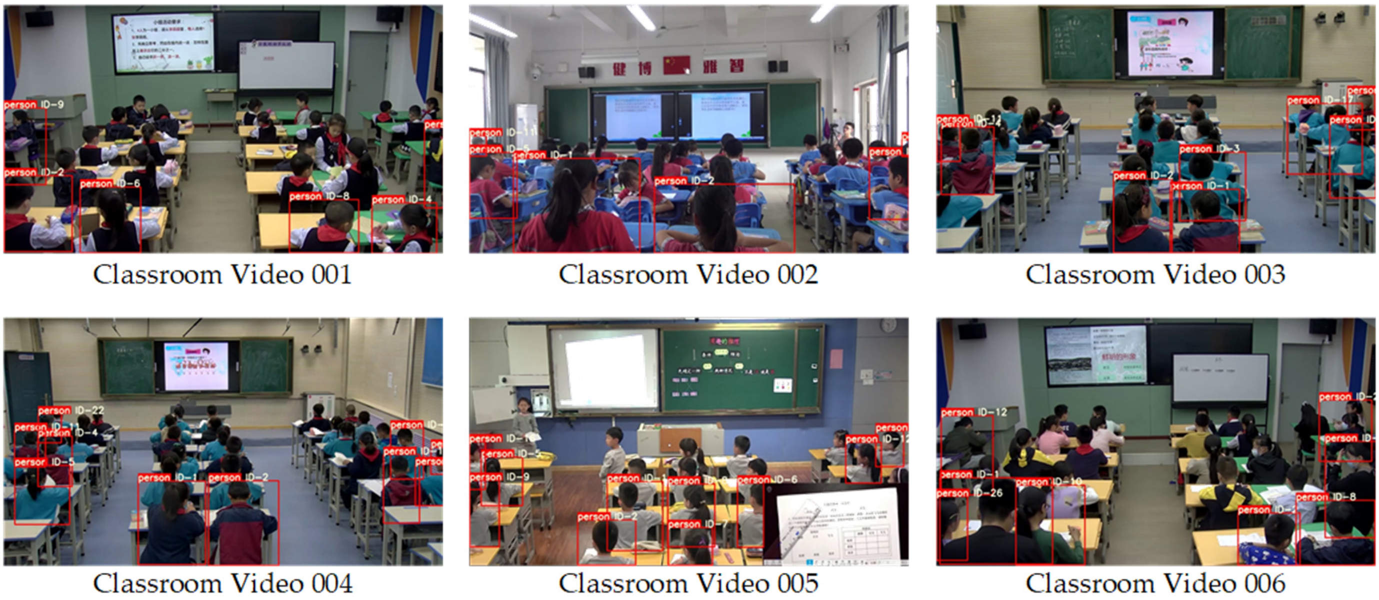
Table 3. Detection results of YOLOv5-DeepSORT in our method with different loss functions.

Loss Function	AP	AP_{50}	AP_{75}	mAP	$mAP_{75:95}$
L_{IoU}	47.61%	67.52%	53.48%	48.76%	34.51%
L_{GIoU}	50.10%	69.49%	55.58%	51.22%	36.04%
L_{CIoU}	49.38%	68.84%	54.69%	50.59%	35.58%
Ours	**52.97%** *	**72.19%**	**57.84%**	**53.51%**	**38.62%**

* The bold numbers in the table represents the optimal value in the evaluation metrics.

Compared with IoU, GioU, and CIoU, YOLOv5-DeepSORT, with the alpha-IoU loss function in our method, achieves impressive results. The metric AP, AP_{50}, AP_{75}, mAP, and $mAP_{75:95}$ are improved by 2.87%, 2.70%, 2.26%, 2.29%, and 2.58%, respectively. Such a high object detection accuracy of the model is easily estimated to be appropriate for the detection task of the shot boundary objects.

Figure 7 reveals the results of using our boundary object detection module to detect objects at the shot boundary in some classroom videos of the self-built dataset CLV.



Classroom Video 001

Classroom Video 002

Classroom Video 003

Classroom Video 004

Classroom Video 005

Classroom Video 006

Figure 7. Detection results of objects at the shot boundary. It shows the results of object detection for objects at the shot boundary in some classroom videos after the target leaves the shot. It can be seen that objects within a certain range of the shot boundary have been detected.

4.2.3. Results of Target Re-Identification

The experimental content of this part was mainly using the proposed model TRTrack, a target re-identification method based on shot boundary object detection for SOT, and state-of-the-art models, including OSTrack-384 [19], MixFormer-L [18], and NeighborTrack-OSTrack [30], to conduct comparative experiments through the classroom video dataset CLV established in this paper.

For these SOT models, OSTrack-384 (proposed by Ye et al.) connects the template and the search region to bidirectional information flows to combine feature learning with interaction. MixFormer-L, developed by Cui et al., introduced a mixed-attention module to build an information interaction channel between the template–search image pairs. NeighborTrack-OSTrack uses the confidence score to automatically utilize the information of neighbor regions that are not obscured. The above three models all achieve good performance in SOT tasks.

The experimental results can reflect the performance of TRTrack for target re-identification, which is shown in Table 4.

Table 4. Target re-identification results of the CLV dataset.

Video ID	OSTrack-384			MixFormer-L			NeighborTrack-OSTrack			TRTrack (Ours)		
	ET (Times)	**ED (s)**	**TD (s)**	**ET (Times)**	**ED (s)**	**TD (s)**	**ET (Times)**	**ED (s)**	**TD (s)**	**ET (Times)**	**ED (s)**	**TD (s)**
001	20	44.62	13.98	26	49.83	18.65	9	25.42	24.49	**2 ***	**2.72**	**2.64**
002	15	27.93	8.72	24	32.11	14.56	5	430.76	416.88	**2**	**2.71**	**2.71**
003	16	34.77	14.21	13	24.89	17.55	6	29.56	27.14	**4**	**5.27**	**5.27**
004	19	41.10	17.44	19	29.74	18.33	8	24.11	23.98	**4**	**2.94**	**2.94**
005	22	137.73	95.56	26	113.86	106.49	5	38.66	36.54	**3**	**3.01**	**3.01**
006	8	146.29	136.61	12	90.01	74.79	4	9.12	7.55	**2**	**2.01**	**1.89**
007	17	24.80	8.56	8	17.13	6.59	**0**	**0.00**	**0.00**	**0**	**0.00**	**0.00**
008	9	13.11	6.78	6	14.89	4.23	1	1.29	1.29	**0**	**0.00**	**0.00**
009	14	47.88	24.96	12	39.47	26.55	11	26.54	24.48	**3**	**5.42**	**4.89**
010	7	16.22	14.56	9	21.54	17.77	4	5.89	5.43	**2**	**1.69**	**1.43**
011	16	42.98	36.66	17	54.32	47.62	**1**	**4.32**	**4.16**	2	5.67	4.93
012	24	145.96	115.43	22	121.11	103.67	13	254.55	241.69	**7**	**9.43**	**9.12**
013	9	54.22	38.14	10	43.12	34.56	6	16.55	14.29	**3**	**8.76**	**7.84**
014	12	15.77	12.97	11	18.56	12.13	3	3.98	3.68	**0**	**0.00**	**0.00**
015	14	66.35	38.54	16	64.59	42.84	8	54.51	48.67	**2**	**7.56**	**7.07**
016	15	101.56	87.52	18	121.76	105.43	7	43.56	41.70	**4**	**11.56**	**8.79**
Avg	14.81	60.08	41.92	15.56	53.56	40.74	5.69	60.55	57.62	**2.50**	**4.30**	**3.91**
Sum	237	961.29	670.64	249	856.93	651.76	91	968.82	921.97	**40**	**68.75**	**62.53**

* The bold numbers in the table represents the optimal value in the evaluation metrics.

The table shows that compared with OSTack-384, MixFormer-L, and NeighborTrack-OSTrack, the TRTrack model proposed in this paper has an excellent performance in the classroom video dataset CLV, and the metric ET, ED, and TD are almost optimal in 16 videos.

The average value of ET is 2.5 times, and the sum value of it is 40 times, which is less than 3.19 times and 51 times of the second-best model NeighborTrack-OSTrack, indicating that TRTrack possesses good tracking stability and is not prone to tracking errors. The average value of ED is 4.3 s, and the sum value is 68.75 s, which is an order of magnitude less than other popular algorithms, indicating that TRTrack has a high tracking accuracy. The metric TD contains the average and sum value of the time interval between the target's reappearance after leaving the shot and being tracked again in a video. For other tracking algorithm models, TD has the order of tens of seconds and hundreds of seconds, accounting for most of the tracking error duration, namely the metric ED. This is enough to show that the re-identification of the target after leaving the shot is a crucial object-tracking problem, and the poor ability in this aspect of current SOT models is the major source of tracking errors. The average value of the TD of TRTrack is 3.91 s, and the sum value is 62.53 s, indicating the effective target re-identification ability of our method.

The comparison between the target re-identification performance of TRTrack and other algorithms is shown in Figure 8.

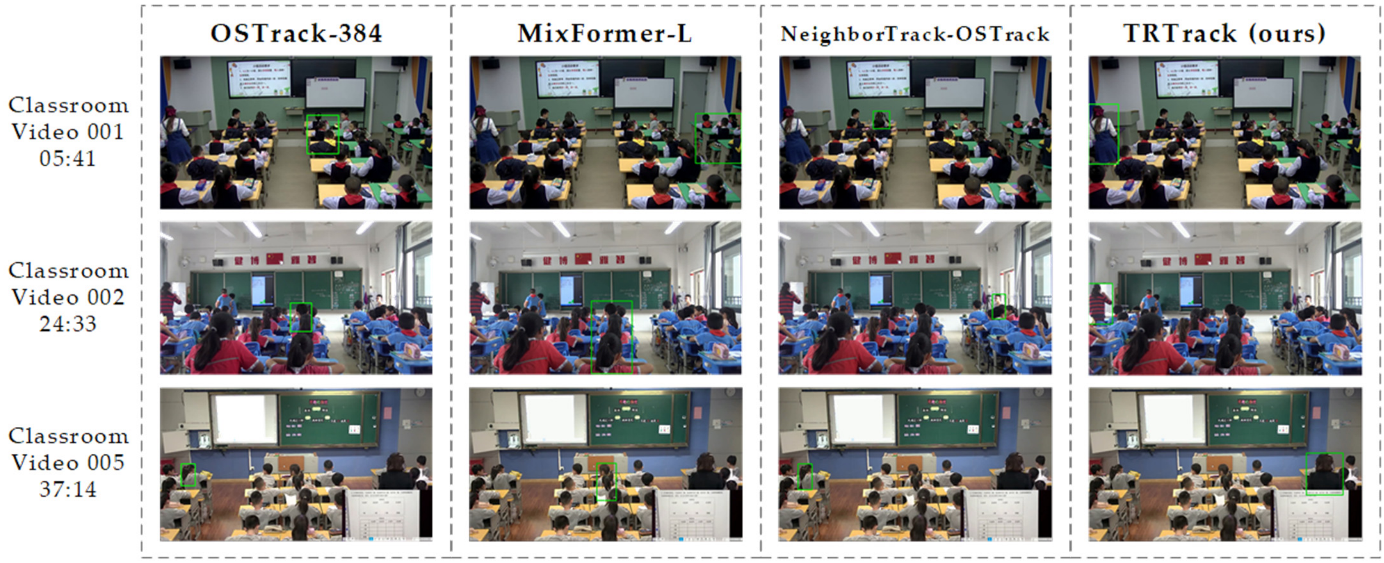

OSTrack-384
MixFormer-L
NeighborTrack-OSTrack
TRTrack (ours)
Classroom Video 001 05:41
Classroom Video 002 24:33
Classroom Video 005 37:14

Figure 8. Comparison of target re-identification results. It can be seen that when targets return to the shot, except for TRTrack, other models have tracking errors, and they are unable to quickly re-identify the target.

4.2.4. Ablation Study of Target Re-Identification

To reflect the optimization results of several modules in TRTrack, including the bipartite matching model in the preliminary object tracking module and improved YOLOv5-DeepSORT in the boundary object detection module, extensive ablation experiments were carried out.

- Effect of the bipartite matching model of the preliminary object tracking module

We used the original NeighborTrack and the bipartite matching model in TRTrack to conduct comparative experiments in our own classroom video dataset CLV to demonstrate the effect of optimizing the NeighborTrack. A part of the experimental results is shown in Table 5.

Table 5. Results of the bipartite matching model of the preliminary object tracking module.

Video ID	**TRTrack with Original NeighborTrack**			**TRTrack with Bipartite Matching Model (Ours)**		
	ET (Times)	**ED (s)**	**TD (s)**	**ET (Times)**	**ED (s)**	**TD (s)**
001	4	8.74	6.79	**2** *	**2.72**	**2.64**
002	5	9.26	7.42	**2**	**2.71**	**2.71**
003	**4**	6.38	**5.16**	**4**	**5.27**	5.27
004	5	7.29	6.94	**4**	**2.94**	**2.94**
005	7	12.16	10.07	**3**	**3.01**	**3.01**
006	6	6.11	4.83	**2**	**2.01**	**1.89**
007	**0**	**0.00**	**0.00**	**0**	**0.00**	**0.00**
008	1	1.65	0.54	**0**	**0.00**	**0.00**

* The bold numbers in the table represents the optimal value in the evaluation metrics.

Table 5 shows that the bipartite matching model can enhance the ability of target re-identification of TRTrack. In most videos of the CLV dataset, the metric ET, ED, and TD of TRTrack with the bipartite matching model are superior to TRTrack with the original NeighborTrack.

- Effect of improved YOLOv5-DeepSORT of boundary object detection module

Apart from comparing the effects of different loss functions on object tracking, the original YOLOv5-DeepSORT and our improved YOLOv5-DeepSORT are used to conduct a comparative experiment, which is also based on the classroom video dataset CLV to demonstrate the optimized effect. A part of the experimental results is shown in Table 6.

Table 6. Results of improved YOLOv5-DeepSOT of boundary object detection module.

Video ID	**TRTrack with Original YOLOv5-DeepSORT**			**TRTrack with Improved YOLOv5-DeepSORT (Ours)**		
	ET (Times)	**ED (s)**	**TD (s)**	**ET (Times)**	**ED (s)**	**TD (s)**
001	7	17.53	15.44	**2** *	**2.72**	**2.64**
002	8	16.74	13.10	**2**	**2.71**	**2.71**
003	7	11.98	10.56	**4**	**5.27**	**5.27**
004	9	19.24	15.13	**4**	**2.94**	**2.94**
005	14	22.87	19.06	**3**	**3.01**	**3.01**
006	7	14.88	12.65	**2**	**2.01**	**1.89**
007	4	5.43	3.29	**0**	**0.00**	**0.00**
008	5	8.47	7.08	**0**	**0.00**	**0.00**

* The bold numbers in the table represents the optimal value in the evaluation metrics.

The effect of improving the loss function is obvious. Table 6 shows that the target re-identification ability of TRTrack based on YOLOv5-DeepSORT with the alpha-IoU loss function is significantly better than TRTrack with the original YOLOv5-DeepSORT. The metric ET, ED, and TD have apparent optimization, indicating that different loss functions have different impacts on the effectiveness of models.

4.2.5. More Visualization of Target Re-Identification

Apart from classroom videos, our target re-identification model, TRTrack, also has excellent target re-identification results on publicly available SOT benchmark datasets. Figure 9 shows the target re-identification effect of TRTrack when some tracking objects leave the shots and reappear in the GOT-10k dataset, which reflects the target re-identification ability of TRTrack. It is worth noting that our model differs from other tracking algorithms in that when the target is identified as leaving the shot, tracking is no longer performed, and no prediction tracking bounding box is provided.

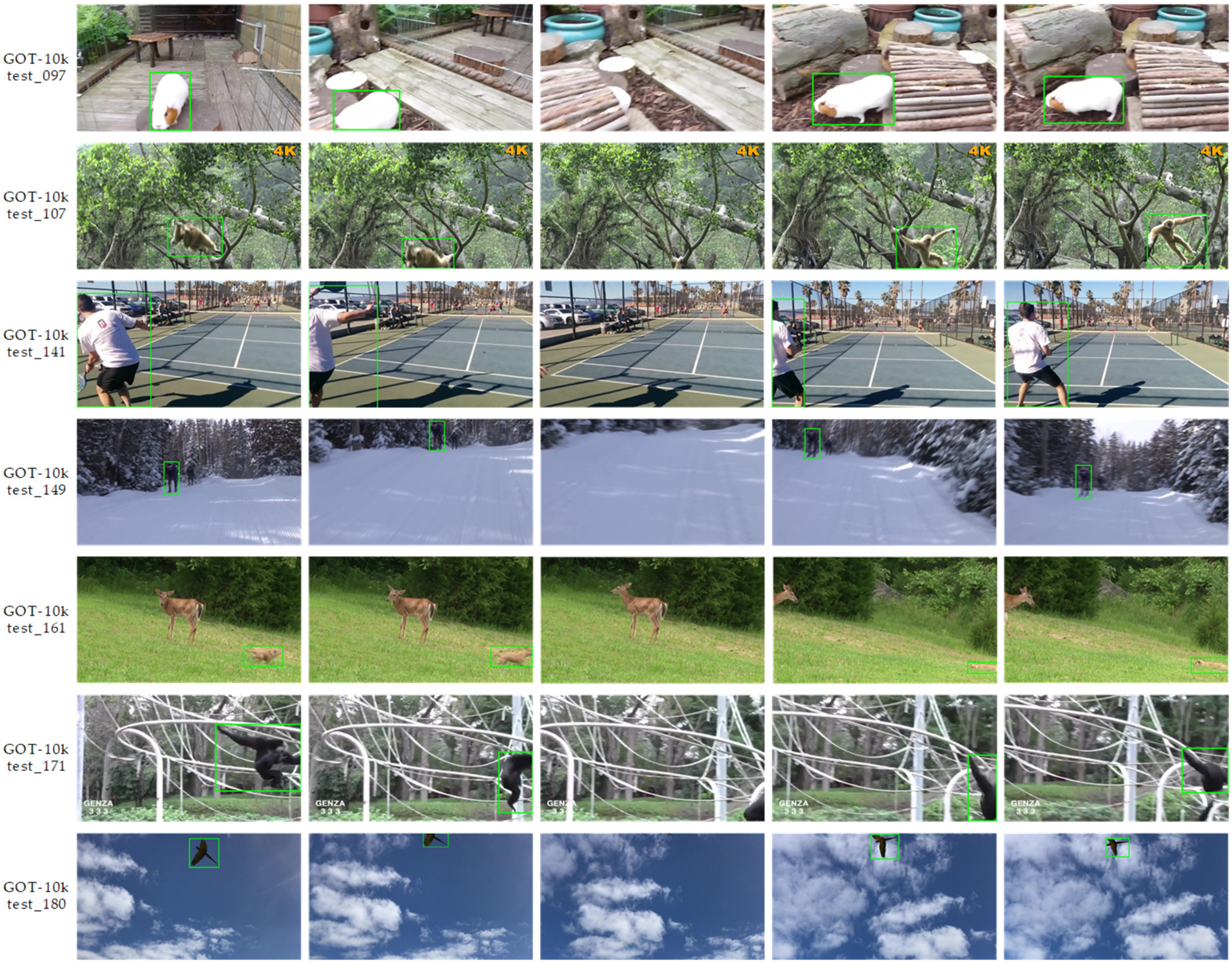


Figure 9. Target re-identification results of some videos in the GOT-10k dataset. In these videos, there are also cases of targets out-of-view, and our model can re-identify the target in a short time.

5. Conclusions and Future Work

5.1. Conclusions

In this paper, a target re-identification method called TRTrack based on shot boundary object detection is proposed to solve the issue of poor single object tracking due to target out-of-view. First, we build a bipartite matching model of candidate tracklets and neighbor tracklets optimized by the Hopcroft–Karp algorithm, which is used for preliminary tracking and judging the target leaves the shot. If the target is in the shot, the tracking will continue. Otherwise, the boundary object detection module based on improved YOLOv5-DeepSORT with the alpha-IoU loss function will detect the shot boundary objects in the video. Finally, the backtracking identification module in TRTrack will crop the detected objects out, replace the current frame image with them as the search region image and input them back into the object tracking model. According to the confidence score of tracking results, the target can be re-identified in a very short time.

The experimental results show that the preliminary object tracking model in our method achieves 73.1% AUC on LaSOT and 76.3% AO on GOT-10k. The YOLOv5-DeepSORT of the boundary object detection module obtains 38.62% $\text{mAP}_{75:95}$ on MS COCO 2017. The proposed target re-identification method TRTrack demonstrates its target re-identification ability at the self-built dataset CLV, which is superior to most state-of-the-art models and has practical application value.

5.2. Future Work

Although our target re-identification method TRTrack has achieved good results, some problems remain. For example, even though we used OSTrack-384 [19] and YOLOv5 with a good performance-speed balance in our model, the real-time performance of our approach is still not good enough. The current frame rate is about 10 fps, which can still be further improved. In future studies, we are expected to further optimize our model by introducing network pruning, model lightweight and other methods so as to improve the rate of target re-identification while maintaining accuracy. We look forward to better development of target re-identification for object tracking in the future.

Author Contributions: Conceptualization, B.M. and Z.C.; methodology, B.M. and Z.C.; validation, B.M. and A.Z.; formal analysis, B.M., Z.C. and H.L.; investigation, B.M, Z.C. and H.L.; resources, B.M. and Z.C.; data curation, B.M., Z.C. and H.L.; writing—original draft preparation, B.M.; writing—review and editing, B.M., Z.C. and H.L.; visualization, B.M.; supervision, Z.C. and H.L.; project administration, B.M. and A.Z.; funding acquisition, Z.C., H.L. and A.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (grant No. 62077022) and the Research Project of National Collaborative Innovation Experimental Base for Teacher Development of Central China Normal University (grant No. CCNUTEIII 2021-21).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, P.; Wang, D.; Wang, L.; Lu, H. Deep visual tracking: Review and experimental comparison. *Pattern Recognit.* **2018**, *76*, 323–338. [CrossRef]
2. Subaweh, M.; Wibowo, E. Implementation of Pixel Based Adaptive Segmenter method for tracking and counting vehicles in visual surveillance. In Proceedings of the 2016 International Conference on Informatics and Computing (ICIC), Mataram, Indonesia, 28–29 October 2016; pp. 1–5.
3. Li, H.; Wu, C.; Chu, D.; Lu, L.; Cheng, K. Combined Trajectory Planning and Tracking for Autonomous Vehicle Considering Driving Styles. *IEEE Access* **2021**, *9*, 9453–9463. [CrossRef]
4. Yi, J.; Liu, J.; Zhang, C.; Lu, X. Magnetic Motion Tracking for Natural Human Computer Interaction: A Review. *IEEE Sens. J.* **2022**, *22*, 22356–22367. [CrossRef]
5. Liu, H.; Fang, S.; Zhang, Z.; Li, D.; Lin, K.; Wang, J. MFDNet: Collaborative Poses Perception and Matrix Fisher Distribution for Head Pose Estimation. *IEEE Trans. Multimed.* **2022**, *24*, 2449–2460. [CrossRef]
6. Marvasti-Zadeh, S.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep Learning for Visual Tracking: A Comprehensive Survey. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 3943–3968. [CrossRef]
7. Henriques, J.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
8. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
9. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939.
10. Galoogahi, H.; Sim, T.; Lucey, S. Correlation filters with limited boundaries. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 4630–4638.
11. Javed, S.; Danelljan, M.; Khan, F.; Khan, M.; Felsberg, M.; Matas, J. Visual Object Tracking with Discriminative Filters and Siamese Networks: A Survey and Outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 6552–6574. [CrossRef]
12. Bertinetto, L.; Valmadre, J.; Henriques, J.; Vedaldi, A.; Torr, P. Fully-Convolutional Siamese Networks for Object Tracking. *arXiv* **2016**, arXiv:1606.09549.

13. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware Siamese Networks for Visual Object Tracking. *arXiv* **2018**, arXiv:1808.06048.
14. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M. Deformable Siamese Attention Networks for Visual Object Tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6727–6736.
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
16. Huang, Y.; Li, X.; Lu, R.; Qi, N. RGB-T object tracking via sparse response-consistency discriminative correlation filters. *Infrared Phys. Technol.* **2023**, *128*, 104509. [CrossRef]
17. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning Spatio-Temporal Transformer for Visual Tracking. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10428–10437.
18. Cui, Y.; Jiang, C.; Wu, G.; Wang, L. MixFormer: End-to-End Tracking with Iterative Mixed Attention. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 13598–13608.
19. Ye, B.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework. In *Computer Vision—ECCV 2022*; ECCV 2022. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13682.
20. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
21. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
22. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
23. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
24. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; ECCV 2016. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9905.
25. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
26. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *Computer Vision—ECCV 2020*; ECCV 2020. Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12346, pp. 213–229.
27. Oksuz, K.; Cam, B.; Kalkan, S.; Akbas, E. Imbalance Problems in Object Detection: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3388–3415. [CrossRef]
28. Huang, L.; Zhao, X.; Huang, K. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1562–1577. [CrossRef] [PubMed]
29. Thangavel, J.; Kokul, T.; Ramanan, A.; Fernando, S. Transformers in Single Object Tracking: An Experimental Survey. *arXiv* **2023**, arXiv:2302.11867.
30. Chen, Y.; Wang, C.-Y.; Yang, C.-Y.; Chang, H.-S.; Lin, Y.-L.; Chuang, Y.-Y.; Mark Liao, H.-Y. NeighborTrack: Improving Single Object Tracking by Bipartite Matching with Neighbor Tracklets. *arXiv* **2022**, arXiv:2211.06663.
31. Wang, Z.; Arabnia, H.; Taha, T. Review of Person Re-identification Methods. In Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2017; pp. 541–546.
32. Zheng, L.; Yang, Y.; Hauptmann, A. Person Re-identification: Past, Present and Future. *arXiv* **2016**, arXiv:1610.02984.
33. Liu, T.; Wang, J.; Yang, B.; Wang, X. NGDNet: Nonuniform Gaussian-label distribution learning for infrared head pose estimation and on-task behavior understanding in the classroom. *Neurocomputing* **2021**, *436*, 210–220. [CrossRef]
34. Xue, W.; Wang, A.; Zhao, L. FLFuse-Net: A fast and lightweight infrared and visible image fusion network via feature flow and edge compensation for salient information. *Infrared Phys. Technol.* **2022**, *127*, 104383. [CrossRef]
35. Liu, T.; Liu, H.; Li, Y.; Zhang, Z.; Liu, S. Efficient Blind Signal Reconstruction with Wavelet Transforms Regularization for Educational Robot Infrared Vision Sensing. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 384–394. [CrossRef]
36. Wang, M.; Liu, Y.; Huang, Z. Large Margin Object Tracking with Circulant Feature Maps. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4800–4808.
37. Liu, Y.; Cheng, L.; Tan, R.; Sui, X. Object Tracking Using Spatio-Temporal Networks for Future Prediction Location. In *Computer Vision—ECCV 2020*; ECCV 2020. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12367.
38. Voigtlaender, P.; Luiten, J.; Torr, P.; Leibe, B. Siam R-CNN: Visual Tracking by Re-Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6577–6587.
39. Ahn, W.-J.; Ko, K.-S.; Lim, M.-T.; Pae, D.-S.; Kang, T.-K. Multiple Object Tracking Using Re-Identification Model with Attention Module. *Appl. Sci.* **2023**, *13*, 4298. [CrossRef]

40. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the 2021 International Conference on Learning Representations (ICLR), Vienna, Austria, 3–7 May 2021.
41. Liu, T.; Liu, H.; Yang, B.; Zhang, Z. LDCNet: Limb Direction Cues-aware Network for Flexible Human Pose Estimation in Industrial Behavioral Biometrics Systems. *IEEE Trans. Ind. Inform.* **2023**. [CrossRef]
42. Liu, H.; Nie, H.; Zhang, Z.; Li, Y. Anisotropic angle distribution learning for head pose estimation and attention understanding in human-computer interaction. *Neurocomputing* **2021**, *433*, 310–322. [CrossRef]
43. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8122–8131.
44. Lin, L.; Fan, H.; Xu, Y.; Ling, H. SwinTrack: A Simple and Strong Baseline for Transformer Tracking. In Proceedings of the 2022 Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 28 November–9 December 2022.
45. Chen, B.; Li, P.; Bai, L.; Qiao, L.; Shen, Q.; Li, B.; Gan, W.; Wu, W.; Ouyang, W. Backbone is All Your Need: A Simplified Architecture for Visual Object Tracking. In *Computer Vision—ECCV 2022*; ECCV 2022. Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13682.
46. Ling, L.; Tao, J.; Wu, G. Pedestrian Detection and Feedback Application Based on YOLOv5s and DeepSORT. In Proceedings of the 2022 34th Chinese Control and Decision Conference (CCDC), Hefei, China, 15–17 August 2022; pp. 5716–5721.
47. Dai, X. Hybridnet: A fast vehicle detection system for autonomous driving. *Signal Process. Image Commun.* **2019**, *70*, 79–88. [CrossRef]
48. Shen, R.; Zhen, T.; Li, Z. YOLOv5-Based Model Integrating Separable Convolutions for Detection of Wheat Head Images. *IEEE Access* **2023**, *11*, 12059–12074. [CrossRef]
49. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16), Barcelona, Spain, 5–10 December 2016; Curran Associates Inc.: Red Hook, NY, USA, 2016; pp. 379–387.
50. Lin, T.; Goyal, P.; Girshick, R.; He, K.; DollárFocal, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef]
51. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
52. He, J.; Erfani, S.; Ma, X.; Bailey, J.; Chi, Y.; Hua, X. Alpha-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression. *arXiv* **2021**, arXiv:2110.13675v2.
53. Chang, Y.; Li, D.; Gao, Y.; Su, Y.; Jia, X. An Improved YOLO Model for UAV Fuzzy Small Target Image Detection. *Appl. Sci.* **2023**, *13*, 5409. [CrossRef]
54. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5369–5378.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.