



RetinaMOT: rethinking anchor-free YOLOv5 for online multiple object tracking

Jie Cao¹ · Jianxun Zhang¹ · Bowen Li¹ · Linfeng Gao¹ · Jie Zhang¹

Received: 30 November 2022 / Accepted: 17 February 2023 / Published online: 8 March 2023
© The Author(s) 2023

Abstract

In recent years, YOLOv5 networks have become a research focus in many fields because they are capable of outperforming state-of-the-art (SOTA) approaches in different computer vision tasks. Nevertheless, there is still room for improvement in YOLOv5 in terms of target tracking. We modified YOLOv5 according to the anchor-free paradigm to be on par with other state-of-the-art tracking paradigms and modified the network backbone to design an efficient module, thus proposing the RetinaYOLO detector, which, after combining state-of-the-art tracking algorithms, achieves state-of-the-art performance: we call it RetinaMOT. To the best of our knowledge, RetinaMOT is the first such approach. The anchor-free paradigm SOTA method for the YOLOv5 architecture and RetinaYOLO outperforms all lightweight YOLO architecture methods on the MS COCO dataset. In this paper, we show the details of the RetinaYOLO backbone, embedding Kalman filtering and the Hungarian algorithm into the network, with one framework used to accomplish two tasks. Our RetinaMOT shows that MOTA metrics reach 74.8, 74.1, and 66.8 on MOT Challenge MOT16, 17, and 20 test datasets, and our method is at the top of the list when compared with state-of-the-art methods.

Keywords Multiple object tracking · Attention mechanism · YOLOv5 · Scene analysis · Pooling pyramid

Introduce

With the in-depth development of multiple target tracking technology, target tracking technology has been widely used in all aspects of life [45]. Along with the rapid development of communication technology and unmanned technology, target tracking based on video streaming and live streaming will be the main development direction in the future [47,67]. This places demands on the accuracy and speed of target tracking.

Various technical problems also arise, such as missed detection, false detection, and tracking target extinction, which can be jointly optimized or partially optimized in the detection model and tracking algorithm [97].

Multiple object tracking algorithms are mainly divided into SDE (separate detection and embedding) and TBD (tracking by detection) approaches. SDE first locates a target using a detection network, extracts features, and finally associates the target by a data association algorithm. Bewley et al. [6] associated the target based on the Kalman filter and a faster R-CNN [93]. modeled the target motion pattern and interaction based on a CNN. Girbau et al. [25] and Shan et al. [61] predicted the target motion state based on graph convolution and recurrent neural networks, respectively. In 2017, Wojke et al. [76] predicted the position based on the Kalman filter, which computes the position information with the appearance features and calculates the closeness between the predicted and detected targets. To reduce detection interference and redundant detection, Chen et al. [8] designed a scoring mechanism to evaluate the tracking process, which was used to remove low threshold detection and unreliable candidate trajectories for secondary matching and finally match the target appearance based on the remaining targets

✉ Jianxun Zhang
zjx@cqut.edu.cn

Jie Cao
cj308861679@163.com

Bowen Li
lbw@stu.cqut.edu.cn

Linfeng Gao
gaojjfyd@2020.cqut.edu.cn

Jie Zhang
52210313143@stu.cqut.edu.cn

¹ Department of Computer Science and Engineering,
Chongqing University of Technology, Chongqing 400054,
China

with the Kalman filter. In 2021, Lit et al. [40] designed a self-correcting Kalman filter for predicting target locations, and they evaluated the similarity between targets by assessing the affinity between targets through a recursive neural network.

A method based on target motion characteristics can combat the interference of short-time occlusion, reducing interference from objects with high similarity in appearance to the tracking model [52]. However, the greatest need is for appearance features, which leads to the poor performance on densely crowded datasets, and such scenarios often have continuously changing target scales [96]. The tracking performance of the method based on target motion features decreases dramatically. Tracking algorithms that combine motion and appearance features have higher robustness. In denser dataset scenarios, where the targets are dense and scale changes are significant, the network complexity is high, which can improve the tracking metrics. However, the disadvantage is that the real-time performance could be better.

In recent years, most researchers have adopted a strategy based on tracking detection (TBD), which can solve many multitarget tracking problems. Wojke et al. [76] proposed the DeepSort algorithm to address the shortcomings of SORT. In another effort, a separately trained pedestrian reidentification module (person reidentification, ReID) model [3] was used to extract deep epistemic features, which can track the obscured target again. However, training through multiple deep learning branches dramatically increases a neural network's computational power, resulting in the poor real-time performance of the algorithm. Most of the target tracking methods in the last two years have continued to improve on DeepSort [17,59]. For example, Zhang et al. [89] performed tracking using only the box obtained from target tracking. The tracking algorithm uses a Kalman filter to predict the bounding box, and a Hungarian algorithm matches the target and the track after predicting the target location. For the excellent use of low-scoring boxes, the authors argued that low-scoring detection frames may be generated when objects are occluded and that discarding low-scoring detection frames directly affect performance. Therefore, the authors used the low-scoring frame for the second matching of the tracking algorithm, which optimizes the problem of switching IDs due to occlusion during the tracking process. Aharon et al. [1] was based on a feature tracker with camera motion compensation and a suitable Kalman-filtered state vector, and this method achieved better bounding-box localization. The camera compensation approach significantly improves the performance of the tracking algorithm, and a simple and effective method of cosine distance fusion of IoU and ReID was proposed. The tracking algorithm can establish a stronger association between prediction and matching.

In our method, we first use YOLOv5 as the target feature extractor, while based on the target information matrix extracted by the feature zero, while predicting the neigh-

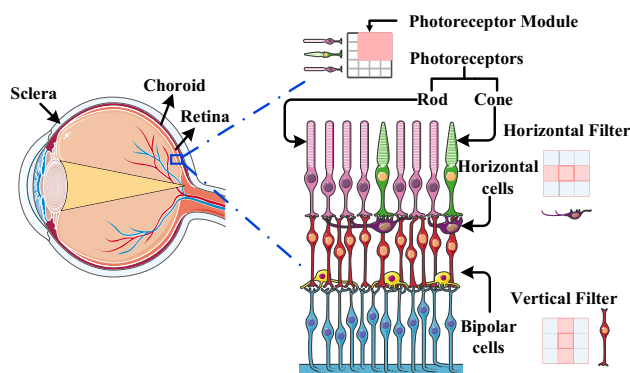


Fig. 1 On the left is a diagram of the structure of the human eye, and on the right is a diagram of the cellular composition of the retina. We consider bipolar cells as vertical filters, horizontal cells as horizontal filters, and rod cells and cone cells as members of photoreceptors representing photoreceptors filters

boring frame positions of the target by Kalman filtering, the Hungarian algorithm associates all predicted positions with a target for matching correlation to get the final tracking result [2].

To balance the accuracy and real-time of target tracking with a one-stage network structure. Xiao et al. [79] first proposed to accomplish two tasks simultaneously within an end-to-end framework, dealing with feature extraction of pedestrian targets and pedestrian re-identification tasks at the same time [42,65]. One-stage detection algorithms in recent years are the most popular when the YOLO series of networks [4,7,24]. His detection performance and real-time performance is unanimously recognized in the field. Therefore, In our work, The YOLOv5 network is fused into a multi-target tracking network, using YOLOv5 as a feature extractor for pedestrian targets to achieve better tracking results.

Before embedding the YOLOv5 feature extractor into the tracker, we focus on the detailed design for feature extraction. We take inspiration from the human retina, similar to a photoreceptor in the human eye. The image of an object falls on the retina through the refractive system. This motivates whether convolution can be simulated to the cellular structure of the retina, and a high-performance convolution capable of extracting deep semantic features can be designed to help multi-target tracking.

The retina is mainly composed of photoreceptors, bipolar cells, horizontal cells, and anaglyph cells, of which the photoreceptors are, in turn, composed of optic sensory cells and optic cone cells. This is shown in Fig. 1.

When uniform light hits the entire retina (i.e., when all photoreceptors are equally excited by the incident light), horizontal cells enhance the contrast at the edges of the visual image. The contour lines of the scene are highlighted by a central-peripheral negative response [57]. Bipolar cells do not simply react to light but begin to analyze image infor-

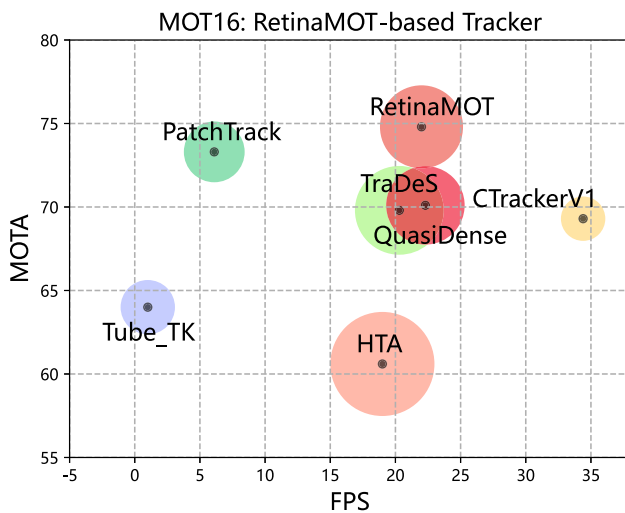


Fig. 2 The competitive nature of RetinaMOT on the MOT16 dataset. The x -axis is the model inference speed, the y -axis is the MOTA metric, and the size of the circle is represented as the number of IDs. The lower the number of IDs is, the larger the radius of the circle. RetinaMOT achieves the highest MOTA metric on the MOT16 dataset and balances the model speed

mation. The bipolar cell's signal conveys information about small dots of light with dark surroundings or small dark dots of light surroundings [34]. The function of the optic rod cells is mainly to perceive changes in light and darkness, but they are poor at discriminating light and can only perceive the general color and shape of objects. The function of cone cells is mainly to distinguish colors and details of objects in detail, but they are not sensitive to light [46].

We constructed two filters and a photoreceptor module based on the roles of the above cells. A powerful feature extraction module was constructed by expressing each cell in a neural network according to its characteristics. The other modules proposed in this paper (Cross-CPC Attention, DeSFPP) are built based on retinal convolution, and all the modules designed significantly improve tracking accuracy. Our contributions in this work are as follows:

- We propose that retinal convolution has strong robustness and feature extraction capability and can improve the performance of other tasks.
- We propose adaptive cascade pyramids to align features while compensating for the shortcomings of pyramid sets.
- We propose a simple and effective method for attention. The importance of the interaction of cross-latitude information and location information in attention is verified (Fig. 2).

Related work

Detectors

We used the state-of-the-art target detection framework YOLOv5 and modified it to an anchor-free detection mode following the method of CenterNet [95], YOLOv5 is a work proposed by [31]. Moreover, as of November 22, their work still has not stopped, the repository is still kept up to date, the algorithm is structured by YOLOv4 [7], using a series of improved methods to bring the speed and accuracy of target detection to new heights, while the YOLOv5 wave has set off, there are also many variations based on YOLOv5 variants: YOLOX [24], YOLOv6 [35], YOLOZ [5], there are currently many trackers that work in conjunction with the YOLOX framework, and it can be said that this has become a mainstream, But we bucked the trend using the YOLOv5 architecture, and we were able to achieve excellent tracking performance by simply modifying the model skeleton and anchoring strategy.

Tracking algorithms

For the motion model estimation, we use the Kalman filter algorithm [75]. The position of motion information in video frames is estimated by Kalman filtering. When the feature extraction network cannot extract the object information, the Kalman filter can also predict the tracked object's position. The correlation matching between the ID and the object is then done by the Hungarian algorithm [63].

Our algorithm's Kalman filtering algorithm is the most basic without optimization. There is no treatment of nonlinear interference or exogenous interference between matched and unmatched targets. Our work focuses on constructing an excellent feature extraction framework. Constructing robust nonlinear predictive filtering systems is our next goal. We carefully studied some excellent algorithms for solving real-world problems, such as [84] constructing an intelligent controller based on disturbance suppression based on constrained nonlinear systems, which fully considers internal and external disturbances and various constraints and compensates for system perturbations in the absence of accurate system modeling. The disturbances are compensated by tracking errors and state estimation errors, and then the network weights are updated, which improves functional performance in cases where the system modeling is not accurate. Moreover, excellent results were achieved in the experiments. Yang [82] proposed two integral robust control algorithms. They ensured tracking performance by cleverly introducing filtering errors and asymptotic filters for the same class of simultaneous matched and unmatched disturbance effects. They used three methods to integrate over the inversion

framework and iterate the filtered signal inversion, extremely and cleverly ensuring the asymptotic tracking performance. Yang et al. [83] used a new inversion method without adding any additional information about the internal dynamics and other physical parameters, integrating a robust nonlinear controller based on a continuous control method with disturbance compensation. High-performance tracking control of nonlinear systems with matched and unmatched modeling uncertainties was obtained. Their excellent work gives us ideas for the subsequent filter optimization.

The performance of the tracker and detector is most tested in exceptionally crowded scenes. Many trackers add the ReID model to improve the reidentification ability of the detector [17,37,91] so that the network can be trained simultaneously. Inference can extract features and targets at the same time. However, the ReID network is not used in our model. ByteTrack’s study found that the feature extractor can replace ReID to handle long-term associations between objects as long as the feature detector progress is high enough. It was also found that adding ReID did not improve the tracking results [89].

Methods

Model structure

The advantage of YOLOv5 is its one-stage structure, which can extract pedestrian target features quickly and accurately. Therefore, we cancel the anchor profile and change the neck configuration in the network to three upsampling and three C3 operations, combined into one and output to the head for the prediction of heatmap, width, height, and offset, which is consistent with CenterNet.

As seen in Fig. 3, the Focus module is used at the beginning of the network, which downsamples the image utilizing array slicing by four array slices. firstly, by four array slices, respectively, The top left, top right, bottom left, and bottom right corners of the image are cut to preserve the feature information, fully preserve the semantic information, and expand the perceptual field. The number of channels is quadrupled by stitching the four feature maps according to the dimension, and finally, the number of channels is restored using convolution.

After the Focus module, the network alternately uses the Cross-CPC Attention (described in detail in the “Cross-dimensional coordinate perception cross-attention” section)

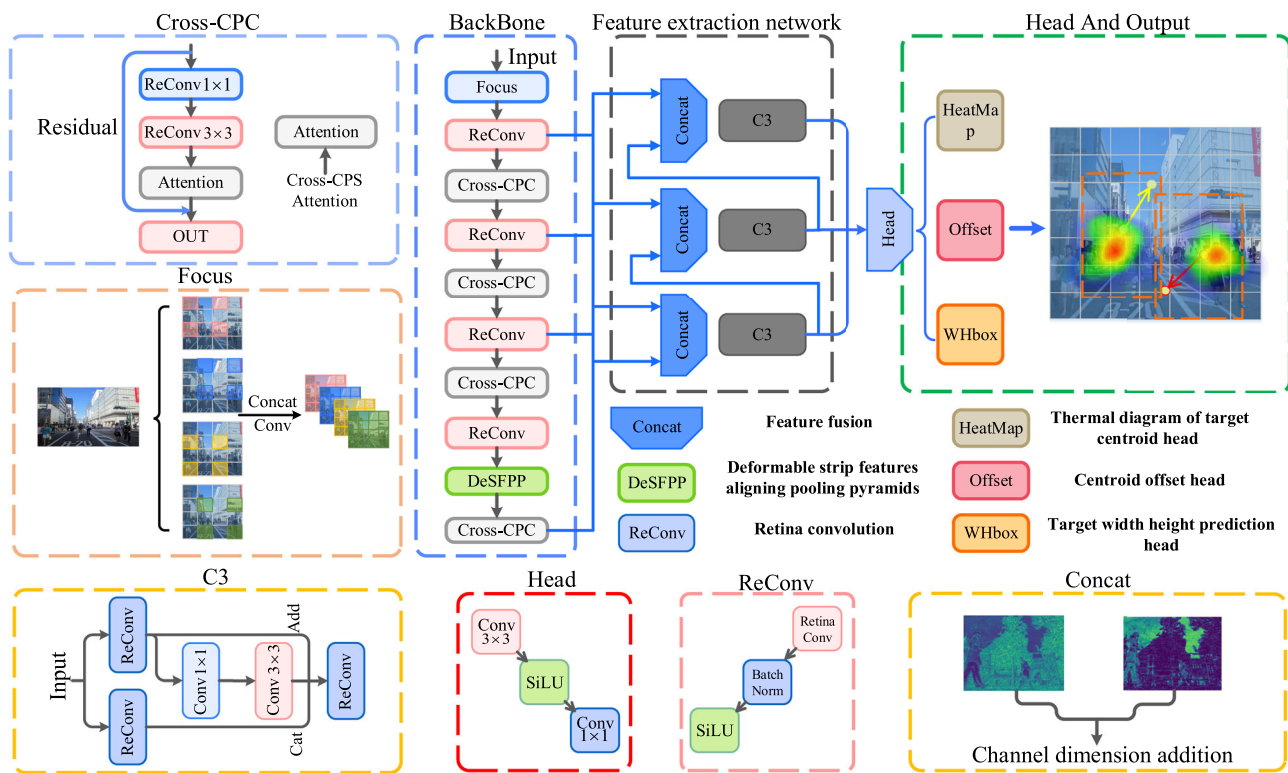


Fig. 3 The structure diagram of the detector network we designed, where Cross-CPC Attention, Retina Conv, and DeSFPP are detailed in the “Cross-dimensional coordinate perception cross-attention”, “Retina convolution”, and “Adaptive cascade feature alignment pooling pyramid” modules, respectively

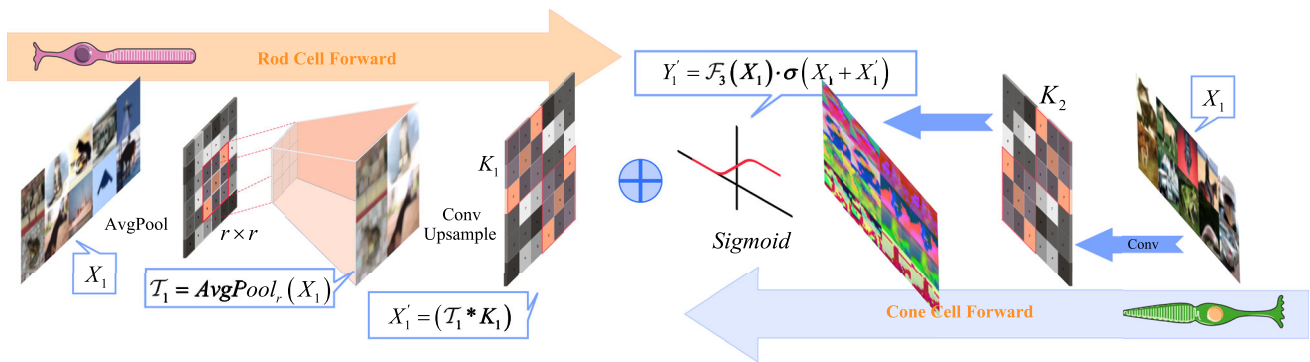


Fig. 4 Detailed structure diagram of the modules represented by rod cell and cone cell

and DeSFPP modules (described in detail in the “Adaptive cascade feature alignment pooling pyramid” section), allowing the Backbone to expand the field of sensation and fully extract feature location information. In the last part of Backbone, DeSFPP is used for upsampling features from different global pools, thus enabling the network to adjust the sampling position of the network. The neural network learns the offset of the occluded pedestrian to locate the pedestrian’s moving position accurately.

The Backbone information is output to the feature pyramid to extract the multi-scale pedestrian information. Lower-level semantic features are integrated with higher-level semantic features through top-down horizontal connections so that each layer has equally rich semantics [71]. By establishing the high-level semantics for the occluded pedestrian and the near and far pedestrian targets, each scale feature can be extracted effectively to improve the tracking accuracy and reduce the number of target identity switching. Finally, the fused features are sent to Head, which outputs an array containing HeatMap information, coordinate offset information, and target width and height prediction information.

It can be seen in Fig. 2. The RetinaMOT method proposed in this paper has promising results on the mainstream tracking methods, achieving the best tracking accuracy with limited and surpassing the mainstream methods at a reasonably fast inference speed.

Retina convolution

In traditional 2D convolution, feature extraction is often performed using a symmetric rectangular window. A standard traditional convolution kernel for feature extraction is uneven in spatial distribution, especially in height occlusion and object pixel deformation, with greater weight at the center intersection of spatially high-resolution images and high similarity in the way height occlusion targets are learned [14]. To enhance the diversity of features and avoid the interference of image noise and uniform weight distribution, we extract uniform weight features using horizontal and vertical convo-

lutional kernels of different shapes. After the convolutional superposition operation is applied to the correction module, it can effectively capture the global information of pedestrian spatial location. The diversity features extracted by the horizontal and vertical convolutional kernels are fused with the weighted features [66], thus allowing the sample features to expand the convolutional perception domain and establish long-term spatial dependence around the occluded target.

At the beginning of the design, we referred to a series of sampling features and working cells of the human retina. The first is the photoreceptor in the retina, which has two components: 1. Rod cells. 2. Cone cells.

Rod cells in the retina are a type of cell in the retina that is comparable to the cone cells, mainly distributed around the center of the retina, and more sensitive to light than the cone cells; its function is mainly to perceive changes in light and dark, but the ability to distinguish light is poor, and it can only perceive the general color and shape of objects for distinguishing contours [58]. According to the function of Rod cells. We use averaging pooling in the entry part of the Photoreceptor Module for averaging only the feature points in the neighborhood, filtering their foreground information, and keeping the background information. The gradient is averaged according to the neighborhood size and then passed to the index position [41] according to the cell’s characteristics. With an input image X_1 in the network, using an average pooling filter size of $r \times r$ and a step size of r , we get $T_1 = \text{AvgPool}_r(X_1)$, after which the features are extracted using 3×3 convolution to get K_1 , and then the pooled features T_1 and the convolution are feature mapped, $X'_1 = \text{Up}(\mathcal{F}_2(T_1)) = \text{Upsample}(T_1 * K_1)$, $\text{Up}(\cdot)$ represents the linear interpolation function that maps the small-scale space features to the original feature space. The design is as shown in Fig. 4.

The role of cone cells in the human retina is to recognize color and object details. The cone cells have conical ends with pigments that filter incident light, thus giving them different response curves [18,20]. Better performance in better visual conditions (when the light is good and the line of sight is

sufficient). Based on these cellular properties, the we first input the image X_1 to the convolution of 3×3 to obtain K_2 , denoted as After this series of operations, it can be expressed as $Y'_1 = \mathcal{F}_3(X_1) \cdot \sigma(X_1 + X'_1)$ where $\mathcal{F}_3(X_1) = X_1 * K_2$, where σ is used to denote the Sigmoid function, and then the dot product operation is performed. The convolution output of the feature input 3×3 after the dot product operation is K_3 , X'_1 is the residual established to establish the weight of the input features and the two parts of the output features, cone cell module is designed in the way as Fig. 4. Finally the whole Photoreceptor Module output is expressed as $Y_1 = \mathcal{F}_4(Y'_1) = Y'_1 * K_3$.

Secondly, there are two essential cells in retinal cells: 1. Horizontal cell, 2. Bipolar cell. The role of the horizontal cell is to adjust the luminance of the visual signal output from the photoreceptor cells and to achieve luminance adaptation of vision.

Moreover, enhance the contrast of the edge of the visual image and highlight the contour lines of the scene [32]. Bipolar cells (bipolar cells) receive signal input from photoreceptors and transmit it to anaglyph cells and ganglion cells after integration. In retinal signaling, bipolar cells show an essential function: shunting visual signals into ON and OFF signals [23,85]. We designed two filters named Horizontal Filter and Vertical Filter according to the characteristics of the horizontal cell and bipolar cell; respectively, we refer to [15] their work as shown in 5 in which they are the convolution kernels of $N \times 1$ and $1 \times N$, respectively. Given the horizontal and vertical filters, the vertical and horizontal striped regions allow the encoding of features. They are establishing dependence on the discrete distribution of the area around the pedestrian target.

We take advantage of the superimposability of convolution [43] (the same input features are sampled at the same downsampling rate using a compatible 2D convolution kernel. Generate feature maps of the same size.) as in the Eq. (1). Add the rich features extracted by the Horizontal Filter and Vertical Filter to the Photoreceptor. Module, a new equivalent convolution kernel can be added, that is, the additivity of the two-dimensional convolution in the Horizontal Filter, Vertical Filter, and Photoreceptor. The additivity of the two-dimensional convolution holds for the convolutional kernels of the Horizontal Filter, Vertical Filter, and Photoreceptor Module. We can reuse the smaller Horizontal and Vertical Filter features for the Photoreceptor Module. Formally, this transformation is feasible on layers p and q . If the bar convolution satisfies the formula (2), it is possible to obtain an equivalent convolution with the same output size as the Photoreceptor Module with the same output size.

$$I * K^{(1)} + I * K^{(2)} = I * (K^{(1)} \oplus K^{(2)}) \quad (1)$$

$$M^{(p)} = M^{(q)}, H_p \leq H_q, W_p \leq W_q, D_p = D_q \quad (2)$$

Table 1 Performance of classical classification networks and classification networks using Retina convolution on Cifar10

Model	Base Top-1	Retina Top-1	Top-1↑
MobileNetV2	92.35%	92.63%	0.28%
Shake-ResNet	91.22%	91.55%	0.33%
DenseNet	85.57%	85.91%	0.34%
WRN-28-4	92.68%	93.22%	0.54%
Resnet18	92.07%	93.16%	1.09%

In Eq. (1), I is a matrix, $K^{(1)}$ and $K^{(2)}$ are two convolution kernels with appropriate sizes, \oplus denotes the summation operation, and in Eq. (2), $M^{(p)}$ denotes the Photoreceptor Module's output graph, $M^{(q)}$ denotes the output graph of two bar convolutions, and H, W, D denotes a convolution of size $H \times W$ with D number of channels. Retina-Conv is a feature summation operation for the three branches according to the convolutional feature summation. In this way, we can implement feature reuse. Enhance deep semantic information of crowded pedestrian targets to greatly improve tracking accuracy.

To fully validate the effectiveness of Retina-Conv, we did not use Retina-Conv convolution directly on the trace dataset when we designed it because we were thinking of easily embedding Retina-Conv into other modules of the whole network. We analyzed its theoretical feasibility and experimented on two classic classification datasets, Cifar10 and Cifar100. Our experiments were compared without any tricks (such as hyperparameter adjustment, learning rate adjustment, data augmentation, etc.), using an Nvidia RTX 3070 8G GPU; the learning rate was always kept at 0.01, Batch-Size was always kept at 12, the optimizer selected random gradient descent, and the loss function used the cross-entropy loss function. The specific experimental data are shown in Table 1.

The RetinaConv designed in this paper can improve the feature extraction ability for various classic networks and can be easily embedded into each network, laying the foundation for improving the tracking accuracy later.

Adaptive cascade feature alignment pooling pyramid

For intensive pedestrian prediction tasks, locating individual pedestrian targets for accurate spatial localization requires rich details of spatial features. When dealing with intensive pedestrian detection tasks, one of the key obstacles to overcome is effectively utilizing feature layers at different scales.

Initially, the spatial pooling pyramid (SPP) was proposed by [26]. The spatial pooling pyramid maximizes the pooling of input feature maps at different scales using different pooling kernels. The advantage of this is that the images are

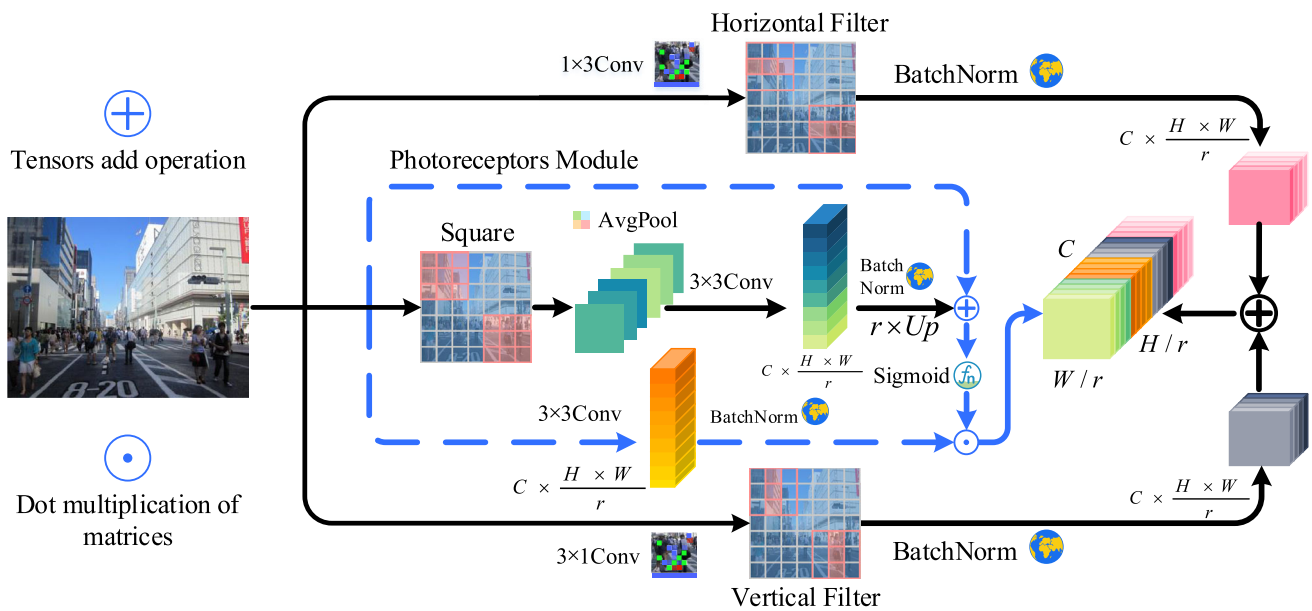


Fig. 5 The detailed design of retinal convolution, the photoreceptors module in the middle, flanked by horizontal and vertical filters, finally converges into the four color feature layers in the figure to represent the output

pooled into a deeper network, and the robustness of the detection model is improved. Similar methods were proposed by [9], proposed atrous spatial pyramid pooling (ASPP). They used hole convolution to obtain a larger perceptual field and make the image feature map size without losing too much edge detail, increasing the perceptual ability of feature maps at different scales.

At present, SPP is mainly used, and the native backbone of YOLOv5 also uses SPP in the final stage of the network to make pooling kernels of size 5, 9, and 13 in each grid of the feature map. However, the size of the feature map is the same after the maximum pooling of different pooling kernels. However, the more significant the pooling kernel is after maximum pooling, the smaller the feature map is, and the size is the same. However, many areas are filled using this approach, which is equivalent to the most specific upsampling feature, where the feature map is directly summed with the local feature residuals, as shown in Fig. 2, where SPP uses the feature map of a smaller pooling kernel and directly catches the feature map of a larger pooling kernel, leading to the problem of contextual misalignment after feature mapping. Indirectly, the location localization fails in the dense pedestrian dataset, making the width-height prediction of the target biased [48]. In addition, the ability of SPP to extract deep high-level semantic information is powerful. However, it mainly relies on stacking the giant pooling kernel in a convolution operation. Hence, his perceptual field is minimal, and there is much room for improvement in scenarios with complex target-tracking datasets (Figs. 5, 6).

Based on the above limitations, we propose a shape-shifting feature alignment pooling pyramid with a detailed

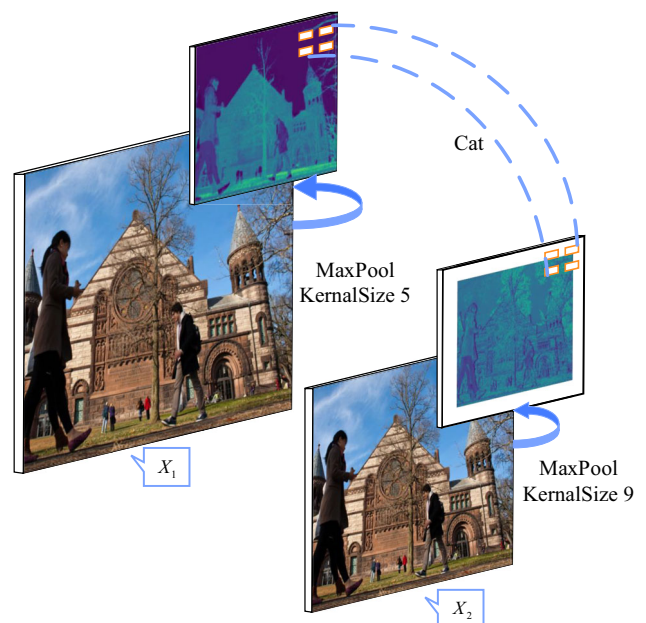


Fig. 6 The SPP module in the original framework uses different maximum pooling kernels, which pool the feature maps into different sizes. Complete zeros are used to keep the smaller and larger feature maps of the same size, but it is not scientific to take a direct tensor summation for the feature maps

design as shown in Fig. 8, and extensive experimental results show that the pooling operation loses most of the information in the image while pooling, which degrades the performance of the whole network. To minimize the loss of information during the pooling operation [64], we first compress the feature maps into multiple pooled feature maps of $N \times N$

Table 2 Experimental data for several variants of DeSFPP have been designed, and the best results have been indicated by bold

Model	MOTA↑	IDF1↑	IDs↓
Base	57.3%	62.6%	336
SPP-A	58.0%	62.0%	314
SPP-B	58.0%	61.1%	357
SPP-C	58.5%	62.3%	344
SPP-D	58.8%	62.0%	331
SPP-E	59.2%	61.0%	345
DeSFPP	59.7%	63.6%	336

size using adaptive averaging pooling and then fuse the feature maps in both spatial directions with semantic information, capturing their information-dependent associations on a particular space. This preserves the location information of dense pedestrians, which helps the detection network to better localize the occluded targets.

We designed six structures to perform the SPP analysis step by step. In the data experiment on the MOT17-half dataset, A indicates that the maximum pooling of SPP is directly replaced with adaptive average pooling (pooling kernel sizes are 1, 2, 3, 6); B indicates that the maximum pooling of SPP is retained and adaptive average pooling is added after the final merge (pooling kernel sizes are 1, 2, 3, 6); C indicates that each maximum pooling of SPP followed by C denotes adding an adaptive average pooling (pooling kernel size 1, 2, 3, respectively) after each maximum pooling of SPP; D denotes replacing the maximum pooling of SPP directly with adaptive average pooling (pooling kernel size 1, 2, 3, 6, respectively) and adding the hole convolution for merging; E denotes replacing the maximum pooling of SPP directly with adaptive average pooling (pooling kernel size refer to the maximum pooling size of SPP: 5, 9, 13).

After our extensive experiments, we proved that the SPP-A module in Table 2, for dense pedestrian prediction replaces the maximum pooling with adaptive averaging pooling directly, which can improve the tracking accuracy slightly, and the number of identity switches is reduced by 22 times. Compared to SPP-B, the tracking accuracy has no change, and the IDs of these metrics are decreased, while in the SPP-C structure, the features after maximum pooling are then connected to an adaptive average pooling to improve the tracking accuracy, indicating that the maximum pooling in SPP has a negative impact on the tracking task. We remove the maximum pooling along this line of thought and only use adaptive average pooling to compress the feature map to $N \times N$, and add in SPP-E, we design the size of the feature map output from adaptive averaging pooling according to the size of the maximum pooling kernel of SPP, removed the hole convolution (multiple hole convolutions increase

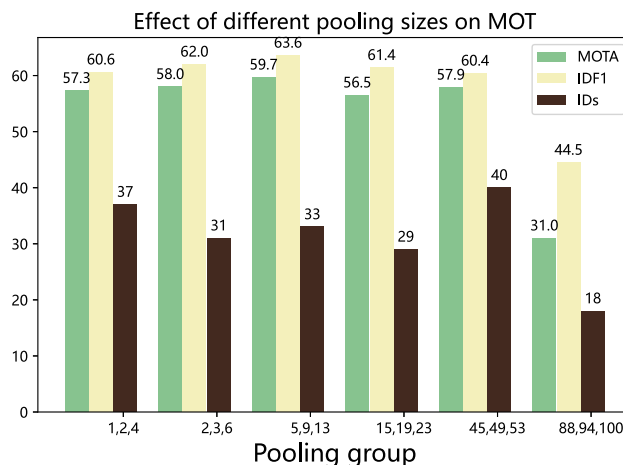


Fig. 7 A large number of experiments were performed for each set of pooling groups, and finally, the optimal pooling group was found in [5, 9, 13]

the inference cost in experiments), and used only the simplest adaptive averaging pooling kernel to make the tracking metrics based on the SPP-E, we introduce the feature pyramid design and design the DeSFPP structure by merging the top-down compressed feature maps with adaptive average pooling outputs of 5, 9, and 13, using retinal convolution to extract features before using adaptive average pooling and then merging all layers afterward (Fig. 7).

When designing DeSFPP, the design of the feature pyramid refers to the FaPN feature aggregation module proposed by [30]. When the input image enters the module, features are first extracted, and channel reduction is performed using a retinal convolution of 3×3 . Then multiple adaptive pools are performed using a cascaded fusion of FaPN, a step we call cascaded AdaFPN. And then feed an adaptive averaging pooling block with different output sizes to output feature map sizes of $C_5 \in \mathbb{R}^{C \times 5 \times 5}$, $C_9 \in \mathbb{R}^{C \times 9 \times 9}$, $C_{13} \in \mathbb{R}^{C \times 13 \times 13}$, arranging them from the bottom up, and each layer of the feature map in into the feature pyramid. It is first used in the feature selection module to extract the target details on the high-resolution spatial map. It is mapped to the location information of the feature map to extract meaningful information and self-calibrate. Equation (4) shows the feature selection module process.

For the pooling layer size in DeSFPP, we did many experiments to show why we chose [5, 9, 13] as our final choice. In Fig. 7. (we scaled all the IDs by a factor of 10 to keep the metrics in the same magnitude), we can see that the larger the pooling layer is, the more significant the impact on the tracking metrics is, When the input pooling layer size in the pooling pyramid is [88, 94, 100], the indicator has dropped very low. Hen the size is smaller than the [5, 9, 13] group, the impact is small, but all things considered, the [5, 9, 13] group is the optimal choice.

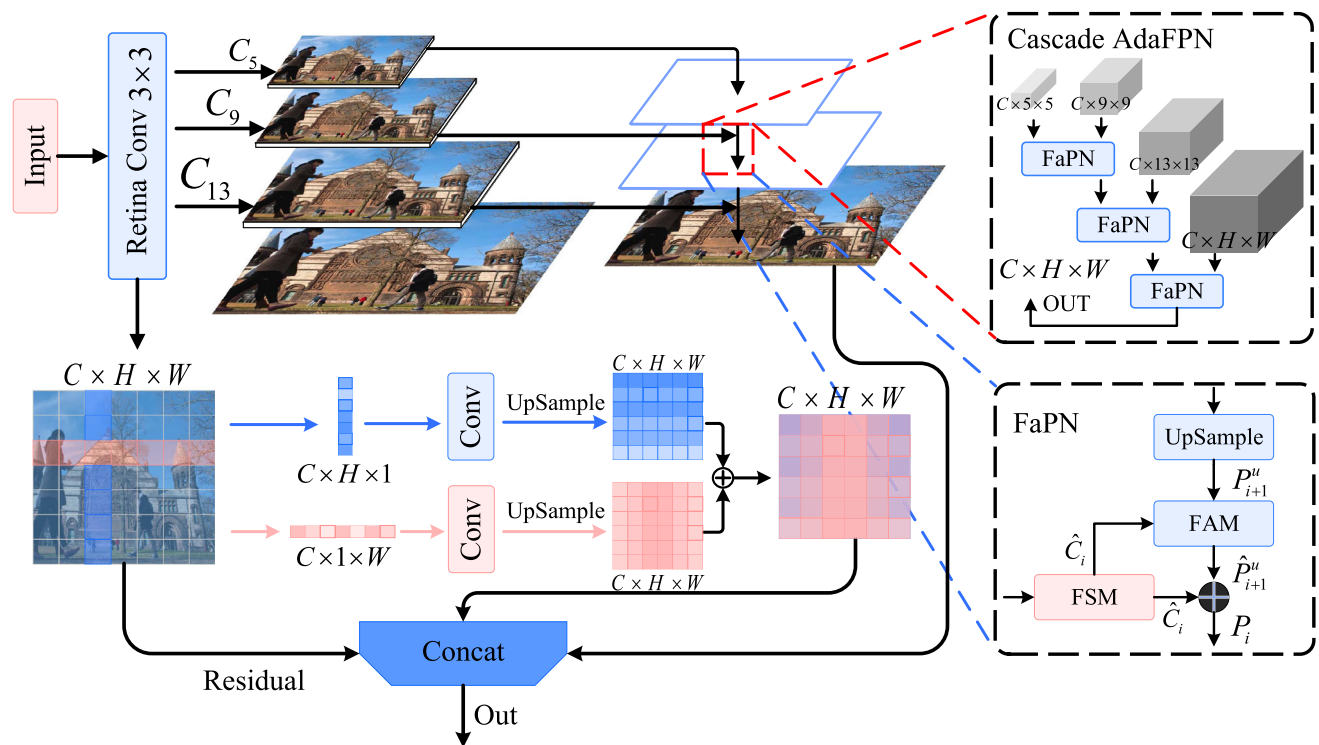


Fig. 8 In the complete DeSFPP, the feature pyramid aggregation part is Cascade AdaFPN, which in turn contains FaPN, which in turn contains FSM and FAM modules, and the specific design of FaPN is in Fig. 9

$$\hat{C}_i = f_s(C_i + \mathbf{u} * C_i)$$

$$\mathbf{u} = f_m(\mathbf{z}) \tag{3}$$

$f_s(\cdot)$ denotes the significant feature selection operation (using 1×1 convolution kernel a BatchNorm), representing the input features as $C_i = [c_1, c_2, \dots, c_D]$, \mathbf{D} means it is the i th channel of the input feature, $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_D]$ denotes the feature weights. The more important features are weighted more. $f_m(\cdot)$ denotes the significant feature construction layer (using 1×1 convolution kernel connected to a Sigmoid), \hat{C}_i denotes the output feature, $z = \text{AdaPooling}(C_i)$ [78], z is computed in Eq. (4)

$$z = \frac{1}{H_i \times W_i} \sum_{h=1}^{H_i} \sum_{w=1}^{W_i} C_i(h, w) \tag{4}$$

The salient features processed by the feature selection module are passed into the FAM module. Due to the frequent use of adaptive averaging pooling, the feature maps are all compressed tiny. At this point, the feature map sizes after different pooling are presented bottom-up, and the feature maps of different sizes are aggregated using FAM. As shown in Eq. (5).

$$\hat{P}_i^u = f_a(P_i^u, \Delta_i)$$

$$\Delta_i = f_o([\hat{C}_{i-1}, P_i^u]) \tag{5}$$

where $f_o(\cdot)$ denotes the feature space difference function, $f_a(\cdot)$ denotes the learning offset information function from the feature, P_i^u is obtained from C_{i-1} after upsampling, $[\hat{C}_{i-1}, P_i^u]$ is concatenated to obtain spatial difference information Δ_i , and Δ_i with P_i^u using deformable convolution [11] via bias learning, which is used to align the top-down feature map features, as specified in Figs. 8 and 9.

After going through the cascade AdaFPN module, the multi-scale features of feature maps with different pooling sizes have been fused. In the other branch, the feature map with the input $C \times H \times W$, the position information in space is extracted separately on each channel to get the weight information of features in wide dimension and high dimension, corresponding to the feature position importance, and then the current position and adjacent features are adjusted by 3×1 and 1×3 convolution, and then upsampled to $C \times H \times W$ size. After 1×1 convolution and sigmoid to obtain the weights, the output feature map is obtained. Each location in the output tensor is related to the pedestrian location information in the input tensor. By re-fusing with cascade AdaFPN, the long-term dependency of the whole scene can be constructed. It not only captures rich scene information but also helps greatly in tracking accuracy cues.

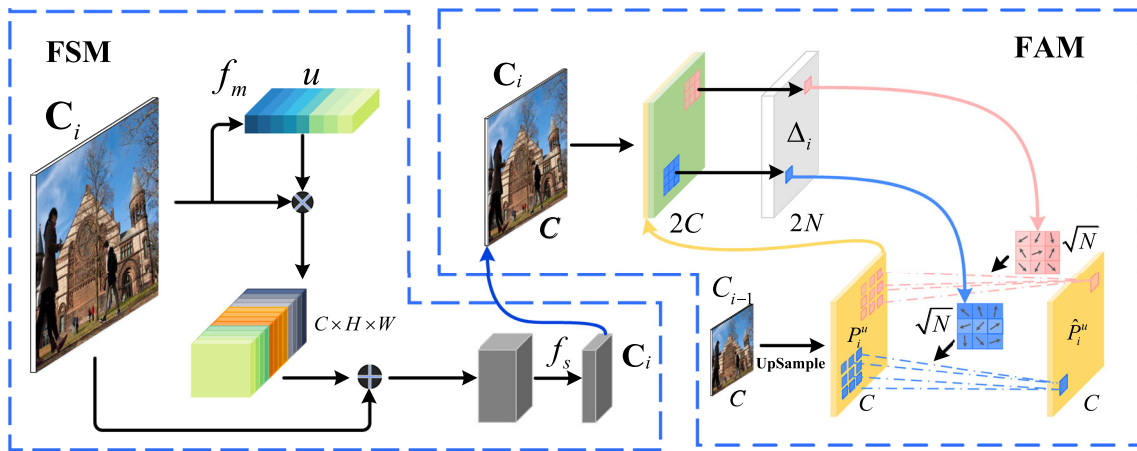


Fig. 9 The specific design of FaPN, which contains both FSM and FAM modules, with FAM using a deformable convolutional implementation

Cross-dimensional coordinate perception cross-attention

In recent years, attention mechanisms have been receiving attention in the field of vision. However, universal channel attention such as SE Net [29] and ECANet [69] only consider channel information modeling to weigh the importance of each channel of the feature without considering the intersection of space, channel, and location, which results in limited feature semantic information being generated. Therefore, a large amount of rich information is ignored in the target tracking process, and as a result, a large number of tracked targets perish. Based on the above research, we propose cross-dimensional coordinate-aware cross-attention. We are inspired by the research of [50] and [28], who designed three branches to capture the cross dimension. However, in the third branch, we obtain the tensor (C, H) , (C, W) , and (H, W) dimension dependencies between them and use the Z-Pool module to concatenate the average pooling and complete pooling information. This approach is challenging to map the global spatial information into the channel features, resulting in the location information not being sufficiently mapped with other dimensional information.

We design three branches based on the above characteristics and defects, passing the tensor $X_1 \in R^{C \times H \times W}$ into the three branches simultaneously. The first branch transforms the tensor X into \hat{X}_1 by doing the rotation operation along H direction. Its shape is $\hat{X}_1 \in R^{W \times H \times C}$, the output tensor $\hat{X}_1^* \in R^{2 \times H \times C}$ after class pooling by the Z-Pool module. Then the Retina-Conv using 7×7 , BatchNorm provides an intermediate output and passes in the SiLU activation function. A linear correction is performed. Finally, \hat{X}_1^* is recalibrated to $X \in R^{C \times H \times W}$ using the Sigmoid activation function to generate the attention weights after the inverse

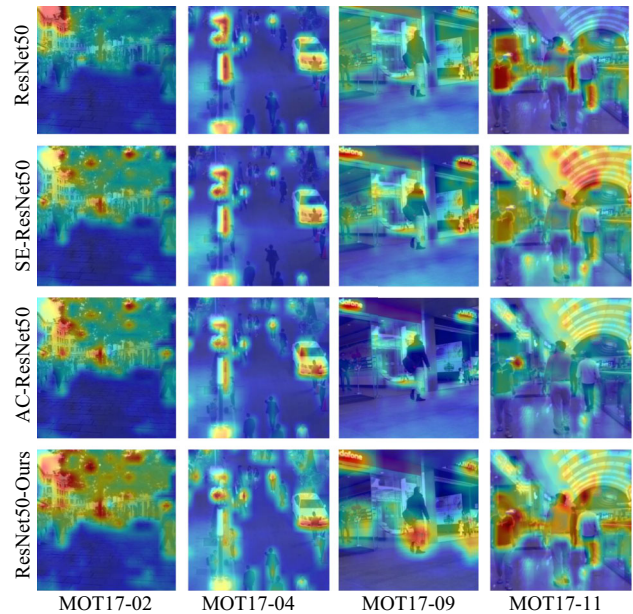


Fig. 10 Visualization of activation maps by Grad-CAM features on ResNet50 using different attention mechanisms

rotation. Eventually, an interaction is established between the H and C dimensions. The detailed design is shown in Figs. 10 and 11.

Similarly, the tensor $X_2 \in R^{C \times H \times W}$ is passed to the second branch to transform the tensor X into \hat{X}_2 by doing the rotation operation along W direction. Its shape is $\hat{X}_2 \in R^{H \times C \times W}$, which is processed by the Z-Pool module for The output tensor $\hat{X}_2^* \in R^{2 \times C \times W}$ after the class pooling process, then the Retina-Conv using 7×7 , then the BatchNorm provides an intermediate output which is passed into the SiLU activation function [19] for linear correction. Finally, the \hat{X}_2^* is recalibrated to $X \in R^{C \times H \times W}$ using the Sigmoid activation function to generate attention weights and then inverse

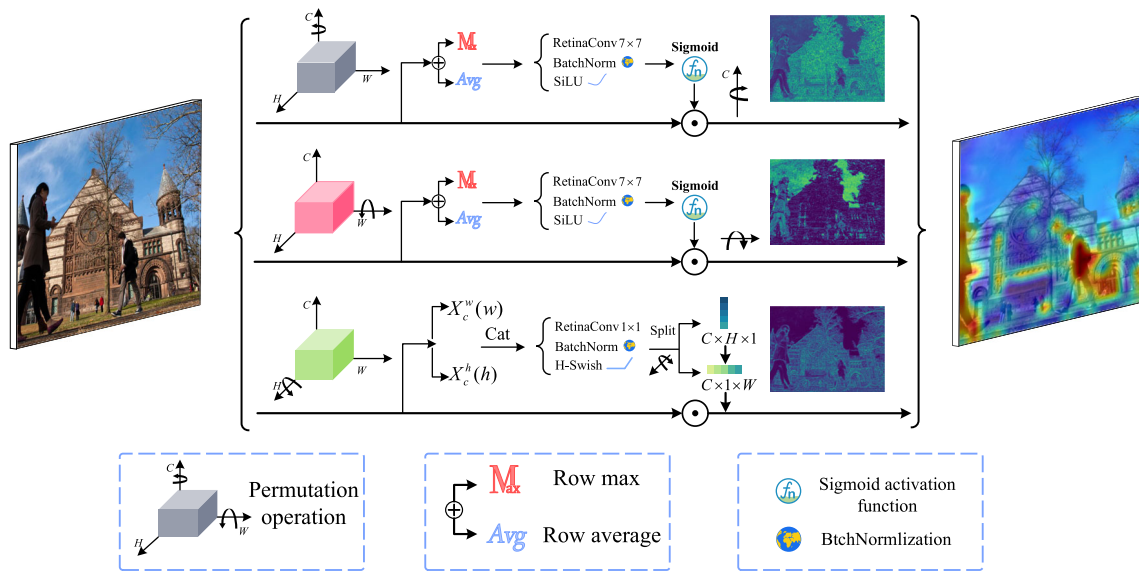


Fig. 11 In the detailed structure diagram of cross-dimensional coordinate perception cross-attention, all components and operations are expressed graphically; the diagram shows that the final output diagram

is a heat map, but in fact, the output is still a feature map because our attention is to improve the focus of the network on the target, we directly visualized the results using Gram-CAM

rotation. Eventually, an interaction is established between the C and W dimensions.

Finally, the third branch, passed into the tensor $X_3 \in R^{C \times H \times W}$, is first encoded for each channel using horizontal and vertical bar pooling, and after vertical pooling, the output is.

$$X_c^h(h) = \frac{1}{W} \sum_{0 \leq i < W} X_c(h, i) \tag{6}$$

Then perform the same operation for W and channel C .

$$X_c^w(w) = \frac{1}{H} \sum_{0 \leq j < H} X_c(j, w) \tag{7}$$

Aggregating the features in both spatial directions greatly helps the stability of target tracking. Convolution and BatchNorm and H-Swith linear correction operations [33] are performed afterward.

$$F = H_{\text{Swish}} \left(F_1 \left(\left[X_c^h, X_c^w \right] \right) \right) \tag{8}$$

After the subsequent channel reduction and dimensional decomposition, the final output is obtained as $\hat{X}_3 \in R^{W \times H \times C} = F(X_3)$. After getting three components, $\hat{X}_1, \hat{X}_2, \hat{X}_3$ the three feature components are converged together, and then a weight balance is performed, and finally, the feature information across dimensions is interacted with features to get deep semantic features and significant atten-

tion.

$$y = \frac{1}{3} \left(\hat{X}_1 \sigma \left(F_1 \left(\hat{X}_1^* \right) \right) + \hat{X}_2 \sigma \left(F_2 \left(\hat{X}_2^* \right) \right) + X \sigma \left(F_3 \left(\hat{X}_3 \right) \right) \right) \tag{9}$$

In our experiments, the good results of cross-dimensional coordinate aware cross-attention are because the detection model considers the intersectionality of space, channel, and location simultaneously, so it captures more features for complex and dense scenes, allowing the target tracking network to perform better. In Fig. 10, we used the Grad-CAM [60] technique on ResNet50 to visualize it, this technique converts the gradient of the input image into a color image, and the darker the color indicates the greater attention to the region. The visualized image can help the understanding of cross-dimensional coordinate aware cross-attention. We use ResNet50 [27] because we want to see how the network focuses region changes after adding our attention, and the simple and practical structure of ResNet50 facilitates our observation and training. Our proposed attention can be seen in Fig. 10, compared to SENet and coordinate attention cross-dimensional coordinate aware cross-attention can focus a larger area, especially for the MOT series pedestrian dataset, where the area to be focused is scattered and requires complete map focus for this particular case.

Experiments

Datasets and evaluation metrics

DataSets Our ablation experiments are trained and evaluated on MOT17-half, MIX-MOT17-half dataset, MOT17-half is the MOT17 [49] dataset half divided into training set half divided into test set, MIX-MOT17-half indicates that the training set is composed of half of the MOT17 sequence and Caltech [16], CityPersons [88], CUHK-SYSU [79], PRW [92], ETH [21] datasets, the comparison experiment used MOT16 [49], MOT20 [12], CrowdHuman [62] datasets The MOT17, and MOT20 datasets are composed of various street or mall scenes, consisting of many sequences with different lighting, pedestrian density, static and dynamic camera shot videos, and different styles of people [42]. The test set results must pass the official Multiple Object Tracking Benchmark provided by mot challenge, the website provides the corresponding evaluation videos, and various strategies are used to avoid overfitting.

Metrics When evaluating the dataset, the following metrics are of primary interest, starting with MOTA (Multi-Objective Tracking Accuracy), which has been used as the primary evaluation metric in the field of multi-objective tracking and can be obtained from the Eq. (10).

$$\text{MOTA} = 1 - \frac{\text{FN} + \text{FP} + \text{IDS}}{\text{GT}} \quad (10)$$

where GT denotes the number of accurate labels in the dataset labels, IDS denotes the number of target identity switches, FP and FN denote the negative samples predicted by the model as positive class FN: positive samples predicted by the model as negative class [22], respectively.

The next is the IDF1 (identification accuracy) metric, as in Eq. (11), where IDFP, IDFN, and IDTP are identifying false positive matches, false negative matches, and true positive matches.

$$\text{IDF}_1 = \frac{2\text{IDTP}}{2\text{IDTP} + \text{IDFP} + \text{IDFN}} \quad (11)$$

Finally, the IDS indicator represents the number of identity switches of the target during the tracking process; the lower the number, the better the effect.

Experimental details

We used YOLOv5 as our backbone, and after all the modification work, We added all the above modules to the native Yolov5 framework. We retrain the COCO dataset in Anchorbase [38] to obtain pre-trained weights for the backbone. All hyperparameters are used in the default Yolov5 parameters except Batch-size, which is set to the explicit memory

limit, and 300 epochs are trained on the COCO dataset. To prevent the use of pre-training weights from causing early convergence of the network resulting in inaccurate accuracy, no pre-training weights are used in the ablation experiments, and pre-training weights are used in the comparison experiments. The MIX dataset (including MOT16, MOT17, MOT20, Caltech, CityPersons, CUHK-SYSU, PRW, ETH, and CrowdHuman) is then uploaded to the mot challenge private evaluator. For the ablation experiments, we used Nvidia RTX 3060 12 G graphics cards, and for the comparison experiments, we used two Nvidia RTX A6000 48 G graphics cards for training.

We used the designed RetinaYOLO as our feature extraction backbone. It was trained on the COCO dataset without loading any pretraining weights to obtain new pretraining weights. The data enhancement used typical hue, saturation, rotation angle adjustment as well as mosaic and mixup, the image input size was set to 640×640 , the initial learning rate after training the COCO dataset was used to obtain the pretraining weights of RetinaYOLO for the backbone, and the MIX dataset was used for training. The input dataset size was 1088×608 during training, and the network learning rate was 0.0001. The inference parameters of the tracking algorithm are set as follows: the tracking confidence threshold is 0.4, the detection threshold is 0.3, the nonmaximum suppression threshold is 0.4, the number of unmatched tracking frames is initially 30, and it is set to 100 frames to slightly improve the IDF1 index. To identify the lost target again in more subsequent video frames, if the detection frame and tracking target frame IOU are less than 0.2, the match is rejected, and the tracking accuracy is unchanged if it is set to 0.1. However, the number of target identity switches will increase. After training the MIX dataset to fine-tune the MOT series dataset, the learning rate step is set to 15 epochs, adjusted once to reduce by ten times, and trained again for 20 epochs. It is found that MOT15 and MOT20 both have a slight increase, but there is no change in MOT17 and MOT16.

Ablative studies

In this section, we scrutinize all the methods mentioned above and conduct rigorous experiments on four valid methods in RetinaMOT, including RetinaConv, Cross-CPC, DeSFPP, and ByteTrack (Table 4).

Componentwise analysis In this subsection, we explore the effectiveness of individual components in the network. To conduct the experiments fairly and objectively, we do not use pretrained models in the ablation experiments. However, we use half of the MOT17 dataset for training and half for evaluation. The 30th calendar hour of training is used as the evaluation criterion. The baseline component ablation experiments are shown in the upper part of Table 3. It can be

Table 3 For ablation experiments on the MOT17 half dataset

NUM	RetinaConv	Cross-CPC	DeSFPP	ByteTrack	MOTA↑	IDF1↑	IDs↓
①	–	–	–	–	50.1%	54.7%	590
②	✓	–	–	–	53.7%	60.9%	578
③	–	✓	–	–	52.6%	58.7%	372
④	–	–	✓	–	53.3%	60.1%	385
⑤	–	–	–	✓	52.1%	57.3%	405
⑥	✓	✓	–	–	56.4%	61.7%	513
⑦	✓	–	✓	–	54.8%	60.5%	390
⑧	✓	–	–	✓	56.7%	61.1%	337
⑨	–	✓	✓	–	51.9%	56.4%	410
⑩	–	✓	–	✓	54.6%	59.1%	349
⑪	–	✓	✓	✓	53.7%	57.5%	386
⑫	–	–	✓	✓	55.6%	60.1%	371
⑬	✓	✓	✓	–	57.1%	61.8%	527
⑭	✓	✓	✓	✓	59.7% (9.6%↑)	63.6% (8.9%↑)	336 (254↓)

↑ means higher results are needed, ↓ means lower results are needed, and the best results are shown in bold

Table 4 For ablation experiments on the MIX MOT17 half dataset

NUM	RetinaConv	Cross-CPC	DeSFPP	ByteTrack	MOTA↑	IDF1↑	IDs↓
①	–	–	–	–	63.7%	69.4%	490
②	✓	–	–	–	65.0%	71.8%	445
③	✓	✓	–	–	65.5%	67.1%	390
④	✓	✓	✓	–	66.7%	69.8%	373
⑤	✓	✓	✓	✓	68.8% (5.1%↑)	71.1% (1.7%↑)	233 (257↓)

↑ means higher results are needed, ↓ means lower results are needed, and the best results are shown in bold

seen that individual components are embedded directly into the baseline without using any tricks to obtain performance gains. Adding RetinaConv in ② and adding DeSFPP in ④ both can get more than 3.0 MOTA value gains. In ④, the direct embedding of DeSFPP we propose in ④ can significantly reduce the value of IDs and significantly improve the IDF1 metric. Our proposed Cross-CPC attention in ③ can directly increase MOTA and IDF1 metrics and reduce the number of IDs by 220. In summary, all our proposed single modules can improve the tracking series metrics exceptionally well.

Self-relation and cross-relation analysis To verify the effectiveness of component self-relations and component cross-relations in our tracking framework, we used several experiments involving cross-components with each other to explore the changes in their tracking metrics. In ⑭, a mixed cross-evaluation of all components can obtain metric gains of 9.6% for MOTA, 8.9% for IDF1, and 245 reductions in IDs based on the baseline; while compared to single components it can be found that some components are limited individually and cross-evaluated. For example, in ④, adding the DeSFPP component alone can achieve a MOTA gain of 3, but in ⑪,

the three components together can only achieve a MOTA gain of 3. Therefore, they must work together in RetinaConv to achieve maximum performance. After we ported ByteTrack to our framework, we found that in both the component self-relations and component cross-relations experiments, we can see that directly adding ByteTrack gives approximately 2% MOTA gain and approximately 2% ID1 gain and results in a significant reduction in IDs.

because we only use the 30th epoch as the standard epoch for evaluation. Therefore, some components do not converge. Therefore, we increase the training dataset to obtain fuller and richer features, and we use MIX-MOT17-half. In the results, we can see that the improvement of ② and ③ is slight. Only a 0.5% MOTA gain is obtained. However, this leads to a reduction in IDs, which indicates that the joint action of these two components makes pedestrian tracking more stable. It can better discriminate highly occluded targets and retrack them, but the MOTA gain of the component that gains more feature weights is limited under the joint action. Experiments in the ④ and ⑤ groups showed that the continued addition of other components again led to an improvement in the MOTA

index, resulting in a significant reduction in the number of pedestrian identity transactions.

We found that our components can improve performance in self-relational verification and multicomponent cross-validation from ablation experiments.

Comparisons with state-of-the-art trackers

For comparison experiments, we embed all the proposed modules into the YOLOv5 framework and train and validate the performance of our model on the MS COCO dataset, with training parameters following the default parameters of YOLOv5. For fairness and objectivity of the comparison, no pretraining weights are used for training, and all 300 epochs are retrained. In Fig. 12, the change of metric information during our training is shown. We use the YOLOv5-S model as our Baseline and compare all the state-of-the-art lightweight frameworks in the last two years, as seen in Table 5. Our RetinaYOLO adds only 1.7M parameters and 1.2G FLOPs to the YOLOv5-S model but increases the $mAP_{0.5:0.95}^{test}$ metric by 6.5% and the $mAP_{0.5:0.95}^{val}$ metric by 5%. If we use the YOLOv6 -T model for comparison, we have 6.1 M fewer RetinaYOLO parameters and more than two times less computation. But the $mAP_{0.5:0.95}^{val}$ metric is going to gain 1.4% more. It can be seen in table 5 that we outperform all the lightweight frameworks of the YOLO family in terms of performance, and we are in the middle of the pack in terms of several parameters and complexity, balancing the speed and accuracy paradoxes extremely well.

After training the MS COCO dataset, we load the best round of pretrained weights in the tracking framework. To make the results fair and just, we use the same training data and evaluator to train the complete MIX dataset, which

contains the training sequences of MOT20, MOT17, and MOT16. To make the controller results in the comparison experiments fair, the algorithm results of Table 6 are derived from the online evaluator of MOT Challenge, the most famous and unbiased evaluation site in the field of target tracking, where target tracking researchers around the world are eager to see new records. The algorithms we compared were all developed by MOT Challenge. The algorithms we compare are all published by the MOT Challenge private detector and are trained using an additional dataset (Crowd Human + MIX dataset). In the detector part of all compared algorithms, various large pretrained datasets are used, such as ImageNet 1K and ImageNet21K [13]), while our detector is trained only on the COCO dataset, which is much smaller in size than the other detectors, and the target tracking paradigm is based on the TBD and JDE paradigms. However, in terms of actual performance, we can compete with them with SOTA performance.

As seen in Table 6, we stand out among all state-of-the-art tracking methods on the MOT16 dataset. Our tracking accuracy is slightly higher than that of GSDTv2 [73], but more than 13 times faster, with 125 fewer IDs and a 5.1% higher IDF1 metric. Similarly, in the current latest publicly available dataset MOT20, compared to the [36] and [98] methods, all of our metrics are higher than the LCC tracking method. The ReKTCL method, on the other hand, is faster than ours and has the highest IDF1 metric among all compared methods with a 2.3 times higher total number of target ID changes. After comparison, we found that we are not the highest in model speed, but the balance of model speed, accuracy, and other vital metrics shows that we should be the best.

Qualitative results

We visualized our tracking results and analyzed the typical situation, as shown in Fig. 13. We analyzed MOT20-01, MOT20-02, and MOT20-03 in the MOT20 dataset, and the three images from top to bottom represent a segment in a video sequence. You can see that in the MOT20-01 sequence, our target of interest can be seen walking side by side with other targets at the beginning of the video. At this time, our target interest number is 141. In the middle segment, the target is obscured and loses its number at this time. After displacement, the target reappears and can correctly match number 141 again.

In the MOT20-02 sequence, the target we focus on at the beginning of the video clip is number 129, a partially obscured target that does not have complete pedestrian characteristics and has been in a semi-obscured state. In the middle part of the clip, the target is more or less displaced and more obscured, and at the end of the clip, although a long time has passed, the tracker can still accurately identify and number this target.

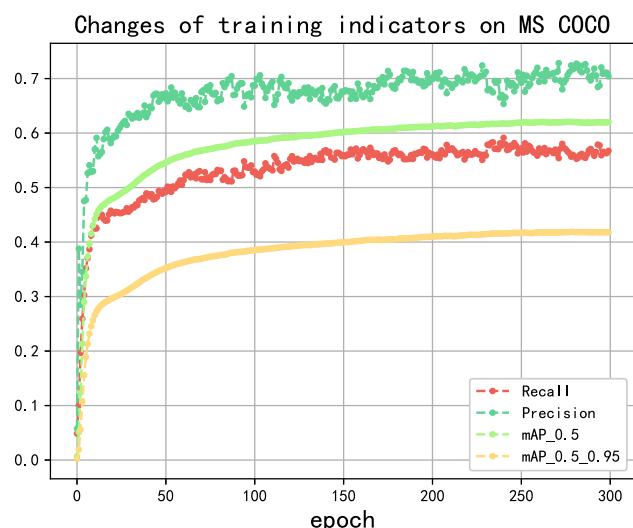


Fig. 12 The metrics change during training without using any pretraining weights and restarting the training

Table 5 The results of post-training evaluation on MS COCO are compared with the most advanced methods and the best results are shown in bold

Model	Size	mAP _{0.5:0.95} ^{test}	mAP _{0.5:0.95} ^{val}	mAP _{0.5} ^{test}	mAP _{0.5} ^{val}	Params (M)	FLOPs (G)
YOLOv5-S [31]	640	36.7%	36.7%	–	55.4%	7.2	17.0
YOLOX-S [24]	640	40.5%	40.5%	–	59.3%	9.0	26.8
YOLOF [10]	640	–	37.7%	–	56.9%	44.0	86.0
YOLOv6-T [35]	640	–	40.3%	–	56.6%	15.0	36.7
PP-PicoDet-L [87]	640	–	40.9%	–	57.6%	3.0	8.91
PP-YoloE-S [80]	640	43.1%	43.1%	60.5%	59.6%	7.9	17.4
YOLOv7-Tiny-SiLU [70]	640	38.7%	38.7%	56.7%	–	6.2	13.8
RetinaYOLO (Ours*)	640	43.2%	41.7%	62.4%	61.7%	8.9	18.2

Table 6 Under the MOT challenge private detector protocol

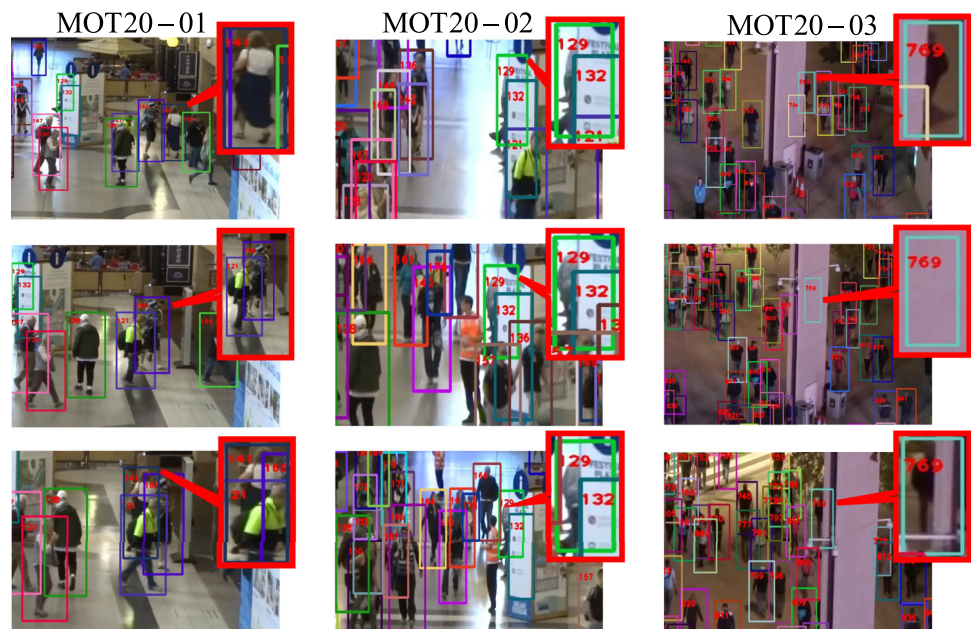
Dataset	Method	Published	MOTA↑	IDF1↑	MT↑	ML↓	IDs↓	FPS↑
MOT16	STPP [72]	TNNLS2020	50.5%	–	19.6%	39.4%	638	–
	TPM [56]	PR2020	50.9%	–	19.4%	39.4%	619	–
	HTA [39]	PRL2021	62.5%	54.3%	33.1%	24.0%	660	19.7
	TubeTK-POI [53]	CVPR2020	66.9%	62.2%	39.0%	16.1%	1236	1.0
	CTrackerV1 [55]	ECCV2020	67.6%	57.2%	32.9%	23.1%	1897	34.4
	QuasiDense [54]	CVPR2021	69.8%	67.1%	41.6%	19.8%	1097	20.3#
	TraDeS [77]	CVPR2021	70.1%	64.7%	37.3%	20.0%	1144	22.3#
	OUnTrack-fm [44]	Neurocomputing2022	74.2%	71.1%	44.8%	13.8%	1328	25.4#
	GSDTv2 [73]	ICRA2021	74.5%	68.1%	41.2%	17.3%	1229	1.6#
Ours*	Ours	74.8%	73.2%	41.5%	18.3%	1104	22.2	
MOT17	STPP [72]	TNNLS2020	52.4%	–	22.4%	40.0%	2224	–
	TPM [56]	PR2020	52.4%	–	22.4%	40.0%	2215	–
	JDE [74]	CVPR2020	63.0%	59.5%	35.7%	17.3%	6171	18.8
	TubeTK [53]	CVPR2020	63.0%	58.6%	31.2%	19.9%	4137	3.0
	CTrackerV1 [55]	ECCV2020	66.6%	57.4%	32.2%	24.2%	5529	34.4
	CenterTrack [94]	ECCV2020	67.8%	64.7%	34.6%	24.6%	3039	22.0
	QuasiDense [54]	CVPR2021	68.7%	66.3%	40.6%	21.9%	3071	20.3#
	XJTU-priv [68]	ACM MM2021	68.8%	64.6%	40.4%	23.7%	2190	0#
	LMOT-Tracker [51]	IEEE Access2022	72.0%	70.3%	45.4%	17.3%	3071	28.6#
RelationTrack [86]	IEEE T MULTIMEDIA2022	73.8%	74.7%	41.7%	23.2%	1374	9.8	
Ours*	Ours	74.1%	70.9%	41.1%	19.4%	3786	22.0	
MOT20	TransCenter [81]	arXiv2021	57.1%	46.7%	35.7%	18.0%	4940	1.0#
	LMOT-Tracker [51]	IEEE Access2022	59.1%	61.1%	25.1%	23.0%	1398	22.4#
	FairMOT [90]	ICCV2021	61.8%	67.3%	68.8%	7.6%	5243	13.2
	XJTU-priv [68]	ACM MM2021	64.3%	66.6%	50.4%	14.0%	3379	0#
	RekTCL [36]	arXiv2021	65.2%	70.1%	61.3%	10.5%	4139	22.4
	LCC [98]	WACV2022	66.0%	67.0%	56.3%	13.3%	2237	0.2#
Ours*	Ours	66.8%	67.5%	58.6%	15.1%	1739	18.6	

The most usual method among the state-of-the-art methods is selected. FPS represents the model's processing speed, our methods are marked with "**", and the best metric results are in bold. Regarding the FPS metrics, the # sign indicates that we did not find the specific metrics from the paper and obtained the results from the MOT challenge competition leaderboard

In the MOT20-03 sequence, the video clip has an obstacle blocking all the targets passing this obstacle; at the beginning of the clip, there is a pedestrian who is passing this obstacle; the number is 769 at this time, after being com-

pletely blocked, the excellent Kalman filter comes into play and accurately predicts the linear moving object, still retaining the tracking frame, at the end of the clip the pedestrian

Fig. 13 On the MOT20 dataset, different scenarios are visualized and three common situations are analyzed



reappears. The detector can identify again that this is the previous target. The number is 769.

From the above visualization analysis, our method can perform accurate identification for target identity, extract deep semantic information about the target, and accurately predict and track objects that are obscured for a long time.

Conclusion

In this work, we propose a tracking framework with three high-performance modules that can be migrated to different domains to achieve different results. At a time when the mainstream detectors in the target tracking domain are pointing to YOLO-X, we have given new thought to the performance of YOLOv5 on the RetinaYOLO detector we designed, obtaining excellent results on both MOT domain and classic target tracking domain datasets. In our subsequent research, we will continue to delve into the field of multiple target tracking, optimize our method, and explore it further.

Acknowledgements This work is partly supported by the National Natural Science Foundation of China [Grant number 61971078] and the Chongqing University of Technology Graduate Innovation Foundation [Grant number gzlxc20223221]. We are grateful to all the researchers in the field of artificial intelligence, on whose shoulders I stood to do such a great job. We sincerely thank the editors and every anonymous reviewer. You are the heroes behind every excellent paper because your professionalism and seriousness in your work improve the journal's quality and promote the development of the human scientific field without your efforts.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as

long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Aharon N, Orfaig R, Bobrovsky BZ (2022) Bot-sort: robust associations multi-pedestrian tracking. arXiv preprint [arXiv:2206.14651](https://arxiv.org/abs/2206.14651)
2. Ahmed M, Maher A, Bai X (2022) Aircraft tracking in aerial videos based on fused RetinaNet and low-score detection classification. IET Image Process
3. Almasawa MO, Elrefaei LA, Moria K (2019) A survey on deep learning-based person re-identification systems. IEEE Access 7:175228–175247
4. Azimjonov J, Özmen A (2021) A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways. Adv Eng Inform 50:101393
5. Benjumea A, Teeti I, Cuzzolin F, Bradley A (2021) YOLO-z: improving small object detection in YOLOv5 for autonomous vehicles. arXiv preprint [arXiv:2112.11798](https://arxiv.org/abs/2112.11798)
6. 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). IEEE, pp 3464–3468
7. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection. arXiv preprint [arXiv:2004.10934](https://arxiv.org/abs/2004.10934)
8. Chen L, Ai H, Zhuang Z, Shang C (2018) Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE international conference on multimedia and expo (ICME). IEEE, pp 1–6

9. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Pattern Anal Mach Intell* 40:834–848
10. Chen Q, Wang Y, Yang T, Zhang X, Cheng J, Sun J (2021) You only look one-level feature. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13039–13048
11. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y (2017) Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 764–773
12. Dendorfer P, Rezatofighi H, Milan A, Shi J, Cremers D, Reid I, Roth S, Schindler K, Leal-Taixé L (2020) MOT20: a benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*
13. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255
14. Ding X, Guo Y, Ding G, Han J (2019a) ACNet: strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1911–1920
15. Ding X, Guo Y, Ding G, Han J (2019b) ACNet: strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks. In: The IEEE international conference on computer vision (ICCV)
16. Dollár P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: a benchmark. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 304–311
17. Du Y, Song Y, Yang B, Zhao Y (2022) StrongSort: make DeepSort great again. *arXiv preprint arXiv:2202.13514*
18. Durmus D (2022) Correlated color temperature: use and limitations. *Light Res Technol* 54:363–375
19. Elfving S, Uchibe E, Doya K (2018) Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw* 107:3–11
20. Elsharif M, Salih AE, Yetisen AK, Butt H (2021) Contact lenses for color vision deficiency. *Adv Mater Technol* 6:2000797
21. Ess A, Leibe B, Schindler K, Van Gool L (2008) A mobile vision system for robust multi-person tracking. In: 2008 IEEE conference on computer vision and pattern recognition. IEEE, pp 1–8
22. Galor A, Orfaig R, Bobrovsky BZ (2022) Strong-transcenter: improved multi-object tracking based on transformers with dense representations. *arXiv preprint arXiv:2210.13570*
23. Gaynes JA, Budoff SA, Grybko MJ, Hunt JB, Poleg-Polsky A (2022) Classical center-surround receptive fields facilitate novel object detection in retinal bipolar cells. *Nat Commun* 13:1–17
24. Ge Z, Liu S, Wang F, Li Z, Sun J (2021) YOLOx: exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*
25. Girbau A, Giró-i Nieto X, Rius I, Marqués F (2021) Multiple object tracking with mixture density networks for trajectory estimation. *arXiv preprint arXiv:2106.10950*
26. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans Pattern Anal Mach Intell* 37:1904–1916
27. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
28. Hou Q, Zhou D, Feng J (2021) Coordinate attention for efficient mobile network design. In: CVPR
29. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
30. Huang S, Lu Z, Cheng R, He C (2021) FAPN: feature-aligned pyramid network for dense image prediction. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 864–873
31. Jocher G, Chaurasia A, Stoken A, Borovec J, NanoCode012 Kwon Y, TaoXie Fang J, imyhxy Michael K, Lorna V A, Montes D, Nadar J, Laughing tkianai yxNONG Skalski P, Wang Z, Hogan A, Fati C, Mamma L, AlexWang1900 Patel D, Yiwei D, You F, Hajek J, Diaconu L, Minh MT (2022) ultralytics/olov5: v6.1—TensorRT, TensorFlow edge TPU and OpenVINO export and inference. <https://doi.org/10.5281/zenodo.6222936>
32. Kawai F (2022) Certain retinal horizontal cells have a center-surround antagonistic organization. *J Neurophysiol*
33. Koonce B (2021) Mobilenetv3. In: Convolutional neural networks with swift for Tensorflow. Springer, pp 125–144
34. Lagali PS, Balya D, Awatramani GB, Münch TA, Kim DS, Busskamp V, Cepko CL, Roska B (2008) Light-activated channels targeted to on bipolar cells restore visual function in retinal degeneration. *Nat Neurosci* 11:667–675
35. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, Ke Z, Li Q, Cheng M, Nie W et al (2022) Yolov6: a single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*
36. Li W, Xiong Y, Yang S, Xu M, Wang Y, Xia W (2021a) Semi-TCL: semi-supervised track contrastive representation learning. *arXiv preprint arXiv:2107.02396*
37. Li Y, Yin G, Liu C, Yang X, Wang Z (2021) Triplet online instance matching loss for person re-identification. *Neurocomputing* 433:10–18
38. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, pp 740–755
39. Lin X, Li CT, Sanchez V, Maple C (2021) On the detection-to-track association for online multi-object tracking. *Pattern Recogn Lett* 146:200–207
40. Lit Z, Cai S, Wang X, Shao H, Niu L, Xue N (2021) Multiple object tracking with GRU association and Kalman prediction. In: 2021 international joint conference on neural networks (IJCNN). IEEE, pp 1–8
41. Liu C, Sun H, Katto J, Zeng X, Fan Y (2022) Qa-filter: a QP-adaptive convolutional neural network filter for video coding. *IEEE Trans Image Process* 31:3032–3045
42. Liu H, Xiao Z, Fan B, Zeng H, Zhang Y, Jiang G (2021) PrGCN: probability prediction with graph convolutional network for person re-identification. *Neurocomputing* 423:57–70
43. Liu J, Luo X, Huang Y (2022b) Facial expression recognition based on improved residual network. In: 2nd international conference on information technology and intelligent control (CITIC 2022). SPIE, pp 349–355
44. Liu Q, Chen D, Chu Q, Yuan L, Liu B, Zhang L, Yu N (2022) Online multi-object tracking with unsupervised re-identification learning and occlusion estimation. *Neurocomputing* 483:333–347
45. Liu S, Liu D, Srivastava G, Połap D, Woźniak M (2020) Overview of correlation filter based algorithms in object tracking. *Complex Intell Syst*
46. Livingstone M, Hubel D (1988) Segregation of form, color, movement, and depth: anatomy, physiology, and perception. *Science* 240:740–749
47. Luo W, Xing J, Milan A, Zhang X, Liu W, Kim TK (2021) Multiple object tracking: a literature review. *Artif Intell* 293:103448
48. Miao J, Wu Y, Liu P, Ding Y, Yang Y (2019) Pose-guided feature alignment for occluded person re-identification. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 542–551
49. Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K (2016) Mot16: a benchmark for multi-object tracking
50. Misra D, Nalamada T, Arasanipalaj AU, Hou Q (2021) Rotate to attend: Convolutional triplet attention module. In: Proceedings

- of the IEEE/CVF winter conference on applications of computer vision (WACV), pp 3139–3148
51. Mostafa R, Baraka H, Bayoumi A (2022) LMOT: efficient light-weight detection and tracking in crowds. *IEEE Access* 10:83085–83095
 52. Nie Y, Bian C, Li L (2022) Object tracking in satellite videos based on Siamese network with multidimensional information-aware and temporal motion compensation. *IEEE Geosci Remote Sens Lett* 19:1–5
 53. Pang B, Li Y, Zhang Y, Li M, Lu C (2020) Tubetk: adopting tubes to track multi-object in a one-step training model. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 6308–6318
 54. Pang J, Qiu L, Li X, Chen H, Li Q, Darrell T, Yu F (2021) Quasi-dense similarity learning for multiple object tracking. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 164–173
 55. Peng J, Wang C, Wan F, Wu Y, Wang Y, Tai Y, Wang C, Li J, Huang F, Fu Y (2020a) Chained-tracker: chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In: *European conference on computer vision*. Springer, pp 145–161
 56. Peng J, Wang T, Lin W, Wang J, See J, Wen S, Ding E (2020) TPM: multiple object tracking with tracklet-plane matching. *Pattern Recogn* 107:107480
 57. Piccolino M, Neyton J, Gerschenfeld H (1981) Center-surround antagonistic organization in small-field luminosity horizontal cells of turtle retina. *J Neurophysiol* 45:363–375
 58. Qiu Y, Zhao Z, Klindt D, Kautzky M, Szatko KP, Schaeffel F, Rifai K, Franke K, Busse L, Euler T (2021) Natural environment statistics in the upper and lower visual field are reflected in mouse retinal specializations. *Curr Biol* 31:3233–3247
 59. Quan H, Ablameyko S (2022) Multi-object tracking by using strong sort tracker and YOLOv7 network
 60. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*, pp 618–626
 61. Shan C, Wei C, Deng B, Huang J, Hua XS, Cheng X, Liang K (2020) Tracklets predicting based adaptive graph tracking. *arXiv preprint arXiv:2010.09015*
 62. Shao S, Zhao Z, Li B, Xiao T, Yu G, Zhang X, Sun J (2018) CrowdHuman: a benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*
 63. Shopov VK, Markova VD (2021) Application of Hungarian algorithm for assignment problem. In: *2021 international conference on information technologies (InfoTech)*. IEEE, pp 1–4
 64. Stergiou Alexandros PR, Grigorios K (2021) Refining activation downsampling with softpool. In: *International conference on computer vision (ICCV)*. IEEE, pp 10357–10366
 65. Sun J, Li Y, Chen H, Peng Y, Zhu X, Zhu J (2021) Visible-infrared cross-modality person re-identification based on whole-individual training. *Neurocomputing* 440:1–11
 66. Tian C, Xu Y, Zuo W, Lin CW, Zhang D (2021) Asymmetric CNN for image superresolution. *IEEE Trans Syst Man Cybern Syst* 52:3718–3730
 67. Tu Z, Zhou A, Gan C, Jiang B, Hussain A, Luo B (2021) A novel domain activation mapping-guided network (DA-GNT) for visual tracking. *Neurocomputing* 449:443–454
 68. Wan X, Zhou S, Wang J, Meng R (2021) Multiple object tracking by trajectory map regression with temporal priors embedding. In: *Proceedings of the 29th ACM international conference on multimedia*, pp 1377–1386
 69. Wang Q, Wu B, Z.P.L.P, Z W, Hu Q (2020) ECA-Net: efficient channel attention for deep convolutional neural networks. In: *The IEEE conference on computer vision and pattern recognition (CVPR)*
 70. Wang CY, Bochkovskiy A, Liao HYM (2022) Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*
 71. Wang J, Zhu C (2021) Semantically enhanced multi-scale feature pyramid fusion for pedestrian detection. In: *2021 13th international conference on machine learning and computing*, pp 423–431
 72. Wang T, Chen K, Lin W, See J, Zhang Z, Xu Q, Jia X (2020a) Spatio-temporal point process for multiple object tracking. *IEEE Trans Neural Netw Learn Syst*
 73. Wang Y, Kitani K, Weng X (2021) Joint object detection and multi-object tracking with graph neural networks. In: *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp 13708–13715
 74. Wang Z, Zheng L, Liu Y, Li Y, Wang S (2020b) Towards real-time multi-object tracking. In: *European conference on computer vision*. Springer, pp 107–122
 75. Welch GF (2020) Kalman filter. *Computer vision: a reference guide*, pp 1–3
 76. 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)*. IEEE, pp 3645–3649
 77. Wu J, Cao J, Song L, Wang Y, Yang M, Yuan J (2021) Track to detect and segment: an online multi-object tracker. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 12352–12361
 78. Xiang S, Liang Q, Hu Y, Tang P, Coppola G, Zhang D, Sun W (2019) AMC-Net: asymmetric and multi-scale convolutional neural network for multi-label HPA classification. *Comput Methods Progr Biomed* 178:275–287
 79. Xiao T, Li S, Wang B, Lin L, Wang X (2017) Joint detection and identification feature learning for person search. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3415–3424
 80. Xu S, Wang X, Lv W, Chang Q, Cui C, Deng K, Wang G, Dang Q, Wei S, Du Y et al (2022) PP-YOLOE: an evolved version of YOLO. *arXiv preprint arXiv:2203.16250*
 81. Xu Y, Ban Y, Delorme G, Gan C, Rus D, Alameda-Pineda X (2021) Transcenter: transformers with dense queries for multiple-object tracking. *arXiv preprint arXiv:2103.15145*
 82. Yang G (2022) Asymptotic tracking with novel integral robust schemes for mismatched uncertain nonlinear systems. *Int J Robust Nonlinear Control*
 83. Yang G, Wang H, Chen J (2021) Disturbance compensation based asymptotic tracking control for nonlinear systems with mismatched modeling uncertainties. *Int J Robust Nonlinear Control* 31:2993–3010
 84. Yang G, Yao J, Dong Z (2022) Neuroadaptive learning algorithm for constrained nonlinear systems with disturbance rejection. *Int J Robust Nonlinear Control*
 85. Young BK, Ramakrishnan C, Ganjawala T, Wang P, Deisseroth K, Tian N (2021) An uncommon neuronal class conveys visual signals from rods and cones to retinal ganglion cells. *Proc Natl Acad Sci* 118:e2104884118
 86. Yu E, Li Z, Han S, Wang H (2022) Relationtrack: relation-aware multiple object tracking with decoupled representation. *IEEE Trans Multimед*
 87. Yu G, Chang Q, Lv W, Xu C, Cui C, Ji W, Dang Q, Deng K, Wang G, Du Y et al (2021) PP-PicoDet: a better real-time object detector on mobile devices. *arXiv preprint arXiv:2111.00902*
 88. Zhang S, Benenson R, Schiele B (2017) CityPersons: a diverse dataset for pedestrian detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3213–3221
 89. Zhang Y, Sun P, Jiang Y, Yu D, Weng F, Yuan Z, Luo P, Liu W, Wang X (2022) ByteTrack: multi-object tracking by associating every detection box

90. Zhang Y, Wang C, Wang X, Zeng W, Liu W (2021) FairMOT: on the fairness of detection and re-identification in multiple object tracking. *Int J Comput Vis* 129:3069–3087
91. Zheng L, Tang M, Chen Y, Zhu G, Wang J, Lu H (2021) Improving multiple object tracking with single object tracking. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 2453–2462
92. Zheng L, Zhang H, Sun S, Chandraker M, Yang Y, Tian Q (2017) Person re-identification in the wild. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1367–1376
93. Zhou H, Ouyang W, Cheng J, Wang X, Li H (2018) Deep continuous conditional random fields with asymmetric inter-object constraints for online multi-object tracking. *IEEE Trans Circuits Syst Video Technol* 29:1011–1022
94. Zhou X, Koltun V, Krähenbühl P (2020) Tracking objects as points. In: *European conference on computer vision*. Springer, pp 474–490
95. Zhou X, Wang D, Krähenbühl P (2019) Objects as points. In: *arXiv preprint [arXiv:1904.07850](https://arxiv.org/abs/1904.07850)*
96. Zhu F, Yan H, Chen X, Li T, Zhang Z (2021) A multi-scale and multi-level feature aggregation network for crowd counting. *Neurocomputing* 423:46–56
97. Zhuo L, Liu B, Zhang H, Zhang S, Li J (2021) MultiRPN-DIDnet: multiple RPNs and distance-IoU discriminative network for real-time UAV target tracking. *Remote Sens* 13:2772
98. Zou Z, Huang J, Luo P (2022) Compensation tracker: reprocessing lost object for multi-object tracking. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp 307–317

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.