IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 July 2024, accepted 17 August 2024, date of publication 26 August 2024, date of current version 10 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3450370

RESEARCH ARTICLE

A Multi Object Tracking Framework Based on YOLOv8s and Bytetrack Algorithm

YINGYUN WANG[1,2] AND VLADIMIR Y. MARIANO[1], (Member, IEEE)
[1]College of Computing and Information Technologies, National University, Manila 1008, Philippines
[2]Big Data and Artificial Intelligence College, Anhui Xinhua University, Hefei 230088, China

Corresponding author: Yingyun Wang (wangyingyun@axhu.edu.cn)

This work was supported by the Quality Engineering Project of Anhui Province under Grant 2020xsxxkc214 and Grant 2016sjjd041.

ABSTRACT In recent years, the YOLOv8 series algorithms have become a research hotspot in many fields, and they can perform excellently in different computer vision tasks. However, YOLOv8 still has room for improvement in multi-target tracking. We integrated it with the Symmetric Positive Definite Convolution (SPD-Conv) module and proposed the YOLOv8s SPD detector, which enhances its detection ability for small targets. The values of mAP@0.5 and mAP@.5:95 have both been increased compared to YOLOv8s. Subsequently, the detector was combined with the ByteTrack tracking algorithm, and the IoU and loss function were optimized to achieve superior performance. We refer to this tracking framework as YBTrack. YBTrack was tested on the Multiple Object Tracking (MOT) Challenge MOT17 and MOT 20 datasets, and achieved MOTA metrics of 74.0% and 66.8%, respectively. Compared with existing tracking frameworks with built-in detectors, our tracking framework has better performance.

INDEX TERMS YOLOv8s, Bytetrack, SPD Conv, MOT, computer vision.

I. INTRODUCTION

Target tracking, which has many applications in automatic driving, video surveillance, and other industries, is a significant area of study in computer vision. It is further classified into two categories: single target tracking (SOT) and multi-target tracking (MOT). Different from SOT, MOT not only has to face the problems of target occlusion, scale change, rapid motion, etc., but also needs to consider the identity label (ID) matching of multiple targets in the front and back frames, which is more difficult and has wider applicability, and has attracted the attention of many scholars at home and abroad. Early MOT algorithms [1], [2], [3], [4], [5], [6] mainly used human-designed features to conduct inter-frame target association, and the available target information was limited. Subsequently, machine learning and deep learning were gradually applied to the field of MOT, and breakthroughs were made [7], [8], [9]. Among these, the deep learning-based MOT method substitutes the initial artificial features with deep features that are extracted by the neural network, significantly enhancing tracking accuracy and resilience. Based on the joint detection and embedding (JDE) paradigm and the tracking-by-detection (TBD) paradigm, the model structure basically divides the MOT algorithm into two ways.

The TBD paradigm cascades detection and tracking tasks, matching detection and tracking boxes through data association, ultimately achieving the goal of ID association. Typical algorithms include SORT [10], DeepSORT [11], and ByteTrack [12]. Detector, motion estimation and data association constitute the framework of SORT. To finish the motion estimate, the target frame is predicted using the Kalman filter [13]. In data association, the Hungarian algorithm [14] is used for matching, and the Intersection over Union (IoU) is employed to calculate the distance between the detection box and the prediction box. DeepSORT proposes a cascaded matching based data association method based on SORT, which adds Mahalanobis distance metric [15] (motion matching) and cosine distance metric (feature matching) before IoU distance metric, improving the long-term association ability of the tracking model. ByteTrack believes that in the detection stage, there are blocked targets in the detection boxes with

The associate editor coordinating the review of this manuscript and approving it for publication was Sunil Karamchandani.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

low scores. If the low score box is removed directly based on the threshold, the target is missing and the missed detection is increased. In order to preserve the veiled target in the low score detection frame, ByteTrack keeps the low score detection frame and matches it with the tracking track twice. The outcomes demonstrate that this approach lowers the missing rate and raises the model's tracking consistency.

Wang et al [16] believe that the two-stage TBD paradigm has efficiency issues, and they proposed the JDE paradigm by studying the shared structure between detection and tracking tasks. The JDE paradigm integrates object detection and Re identification (Re ID) networks into one network, and then performs subsequent matching and tracking based on the network output, but its essence is still a two-stage pattern. After studying existing one shot trackers, Zhang et al. [17] pointed out that prior anchor boxes may deviate from the actual target area, and there is not a one-to-one correspondence between anchor boxes and targets, which seriously affects tracking performance. They proposed an Anchor free JDE method called FairMOT, which locates the center point of the target through a heatmap and uses low dimensional Re ID feature vectors to reduce overfitting risk and improve model robustness. CTracker [18] and CenterTrack [19] further integrated the data association matching process on the basis of the two-stage JDE paradigm, achieving end-to-end training of the tracking model, making it a true single-stage model.

In addition, some scholars put forward a new paradigm based on tracking-by-attention (TBA) from the Query-Key mechanism. For example, TransTrack [20], TrackFormer [21] and MOTR [22] are all built based on Transformer [23] architecture, and use Query to represent the target and complete the detection and tracking of the target through the codec structure. TBA paradigm implicitly realizes data association and improves tracking efficiency to a certain extent, but its model structure is fixed, which limits the flexibility of application.

Although MOT research has made significant progress, multi-target tracking still faces significant challenges: 1) the problem of difficult identification of appearance features for small targets in tracking targets. On the one hand, when different targets interact frequently, the tracker finds it difficult to distinguish them based on their appearance features and location information. On the other hand, when the target is obstructed by obstacles in the background, it will disappear briefly or for a long time, affecting the accuracy of target detection. 2) The problem of target loss caused by motion estimation deviation caused by irregular target motion. Most of the existing algorithms consider the motion of the target rule. If the target suddenly accelerates or turns, it will make the motion estimate inaccurate, resulting in an IoU of 0, making the target lose the chance of matching.

In response to the above issues, considering the simplicity and ease of implementation of engineering applications, as the detectors and trackers of JDE and TBA paradigms are not designed in a separate cascading manner, it is not possible to flexibly replace detectors. Therefore, this article chooses to design a tracking model using the TBD paradigm, which has high application value and is flexible to replace. The well-performing YOLOv8s [24] detector and the SPD Conv [25] module were combined to enhance small target detection performance. Concurrently, a multi-target tracking system based on matching for Expansion Intersection over Union (EIoU) is suggested. Firstly, for the first matching of trajectory and high score detection, a measure index based on small expansion intersection ratio is designed to improve the performance of high score detection box direct matching. secondly, a measure index based on large expansion intersection ratio is designed to match the activation trajectory and low score detection twice, which improves the tracking performance of low score detection frame. In this paper, MOT17 [26] and MOT20 [27] data are selected as experimental data sets, and the experimental results demonstrate that the suggested framework has higher robustness and better precision when compared to the current multi-target tracking system.

II. YBTrack TRACKER FRAMEWORK

The detection and tracking framework based on TBD design requires first inputting each frame in the video into the detector to generate detection boxes, and then linking the detection boxes to the tracking trajectory. In this method, the effectiveness of marker detection is particularly important because it determines the number and type of targets to track. This article first improves the object detector. Then, the optimized YOLOv8s fusion BYTE data association method is used to achieve target recognition and tracking.

A. YOLOv8 TARGET DETECTION

YOLOv8 was introduced by the YOLOv5 team in January 2023 and is a continuation of the YOLO [28] series. It can swiftly finish multi-image processing jobs such object detection, instance segmentation, picture classification, and key point detection. It is compatible with a wide range of image processing tasks. To accommodate various scenarios, each of these processing tasks has five distinct parameter models: n, s, m, l, and x.

Input, Backbone and Head form the overall network structure of YOLOv8. In the Input, YOLOv8 closes the Mosiac enhancement operation in the last 10 iteration cycles of the data enhancement section, which can effectively improve accuracy. In the Backbone section, the convolution in YOLOv8 uses Conv blocks, namely 2D convolution Conv2d, 2D batch normalization BatchNorm2d, and activation function SiLU. The last layer of Backbone is the SPPF module, which consists of two Conv modules before and after, three MaxPooling modules in series in the middle, and one connection layer. The input feature map passes through a Conv module, then undergoes three max pooling operations, and finally passes through a Conv module. The connection layer connects four feature maps of Conv, Conv and 1 max pool, Conv and 2 max pools, Conv and 3 max pools to realize feature fusion at different scales. YOLOv8 replaces all C3

modules in YOLOv5 with C2f modules to get richer gradient flow information. In the Head section, YOLOv5's original Anchor Based has been replaced with Anchor Free.

B. SPD Conv MODULE

A space-to-depth layer and a non-strided convolution layer make up the space-to-depth layer and non-strided convolution layer (SPD-Conv) [25].

In the spatial-depth layer, the original feature map X with any size $S \times S \times C$ *i*s segmented by the *scale* factor, and two feature subgraphs of *scale*, $f_{x,y}$, and the dimensions are $(\frac{S}{scale}, \frac{S}{scale}, C)$, and the *scale* multiple downsampling of the original feature map X is realized. Then, the feature submap is connected along the channel dimension to obtain the middle layer feature map $X'(\frac{S}{scale}, \frac{S}{scale}, scale^2C)$, which preserves every bit of data in the channel dimension. Equation 1 depicts the calculating procedure.

$$
\begin{aligned}
&f_{0,0} = X\,[0:S:scale, 0:S:scale]\,,\\
&f_{1,0} = X\,[1:S:scale, 0:S:scale]\,,\ldots,\\
&f_{scale-1,0} = X\,[scale-1:S:scale, 0:S:scale]\,;\\
&f_{0,1} = X\,[0:S:scale, 1:S:scale]\,,\\
&f_{1,1} = X\,[1:S:scale, 1:S:scale]\,,\ldots,\\
&f_{scale-1,1} = X\,[scale-1:S:scale, 1:S:scale]\,;\ldots\\
&f_{0,scale-1} = X\,[0:S:scale, scale-1:S:scale]\,,\\
&f_{1,scale-1},\ldots,\\
&f_{scale-1,scale-1} = X\,[scale-1:S:scale, scale-1:S:scale]\,.
\end{aligned}
\tag{1}
$$

Figure 1 takes $scale = 2$ as an example, and the original feature map $X[S, S, C]$ is split to obtain four feature subgraphs, $f_{0,0}, f_{0,1}, f_{1,0}, f_{1,1}$, all of which are $(\frac{S}{2}, \frac{S}{2}, C)$, and 2 times the downsampling of X is realized. Then, the feature sub-maps are connected to obtain the middle layer feature map $X'(\frac{S}{2}, \frac{S}{2}, 4C)$. The original X's length and width are cut in half, while the channel dimension increases to four times its original size.

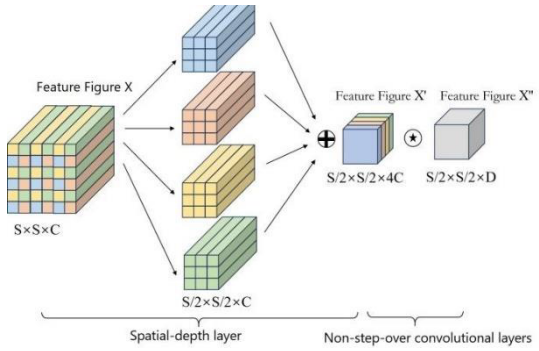

FIGURE 1. Space depth conversion module.

The non-step-over convolutional layer, the D-filter is used to further transform the feature map X' of the middle layer to obtain the feature map $X''(\frac{S}{2}, \frac{S}{2}, D)$. Since the non-step-over convolution retains the feature information of X' to the greatest extent, the SPD-Conv module realizes downsampling while retaining the feature information as much as possible.

C. YOLOv8s SPD ALGORITHM

Small target detection is a very challenging task, because of the low resolution of small target, when it coexists with large target, the feature learning process is often dominated by large target, which is easy to cause small target missing detection. The YOLOv8 backbone network's step convolution module uses downsampling to increase the sensitivity field and decrease parameter calculation, which solves the problem of large amount of redundant pixel information in scenes with high image resolution and moderate object size. However, it inadvertently loses fine-grained information, which greatly reduces the feature learning ability. In turn, YOLOv8s has a low accuracy in detecting small targets, so it is not suitable for practical applications. This study suggests the YOLOv8s-SPD approach to address these issues by optimizing the step convolution and pooling layer using the SPD-Conv module to achieve downsampling without sacrificing learnable features.

In order to reduce complex background interference, YOLOv8s-SPD algorithm adds SPD-Conv module to the Conv module of backbone network and head network, so that the network can focus on small target features. Specifically, the improvement lies in the addition of SPD Conv modules to replace the 3rd, 6th, 9th, and 12th layers of the YOLOv8s backbone network, as well as the initial downsampling modules found in the Head network's layers 22 and 28. The suggested algorithm's network structure diagram is displayed in Figure 2.

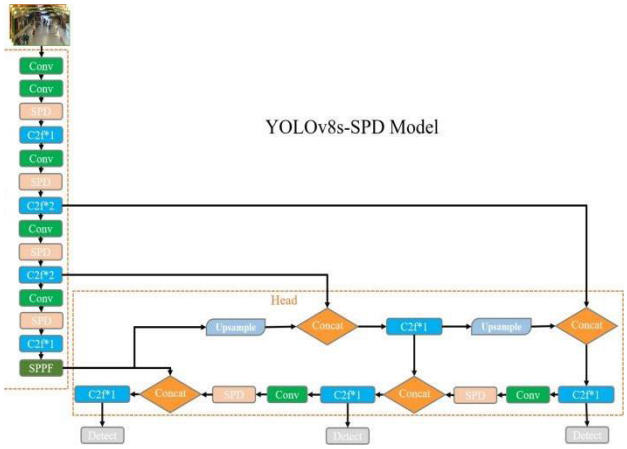

FIGURE 2. YOLOv8s SPD network structure diagram.

D. ByteTrack MULTI OBJECT TRACKING ALGORITHM

Based on the TBD paradigm, ByteTrack is a multi-target tracking technique that uses the detector to obtain the detection frame for track tracking [12]. In the detection phase,

most TBD algorithms discarded the low score detection box directly, but this may cause the occluded real target to lose its tracking trajectory. In order to improve such problems, ByteTrack uses the data association method BYTE to establish trajectories for high score detection boxes while performing secondary matching between low score detection boxes and trajectories, mining real targets and maintaining trajectory coherence. The specific operation process is as follows:

First, The high frame threshold is set according to the YOLOv8s-SPD detection results. If the confidence of a detection box exceeds the confidence of the box, it is added to the D_{high} set, the set of high-scoring detection boxes. It is also necessary to determine the low frame threshold. If the confidence of a detection box falls between the low score threshold and the high score threshold, it is added to the D_{low} set (low score detection box set).

Second, D_{high} is matched with an existing trajectory for the first time. The IoU distance matrix of the D_{high}. high-resolution frame and the trajectory set is calculated and matched by the Hungarian algorithm. For successfully matched tracks, its Kalman filter is updated and placed in the current frame track collection. The trajectories that failed to be successfully matched are placed into the set of trajectories that failed to be matched in the first association, and in T_{remain}, the high-scoring detection frames that failed to be successfully matched are placed in the set of detection frames that failed to be matched in the first association, D_{remain}.

Third, the second IoU association match was performed with the D_{low} and T_{remain} trajectories. The IoU distance matrix of the D_{low} and T_{remain} trajectory sets is calculated, and the Hungarian algorithm is used to match. Trajectories that fail to be successfully matched are put into the lost trajectories collection, and the low-score detection boxes that fail to be successfully matched are directly deleted in T_{lost}. For successfully matched tracks, its Kalman filter is updated and placed in the current frame track collection.

Finally, Track creation, deletion, and merge. For the detection box in D_{remain}, if the confidence value is greater than the tracking score threshold, a new trajectory is created for it and merged into the current frame trajectory set, otherwise it is not processed. For tracks that remain in T_{lost}, 30 frames are retained, and when they reappear, they are matched, and if 30 frames do not reappear, they are deleted. The policy process flow chart is displayed in Figure 3.

E. GREEDY MATCHING ALGORITHM BASED ON EIoU

In multi-target tracking tasks, due to changes in states such as occlusion, target interaction, and irregular motion, it is difficult to obtain appearance and motion features, i.e. IoU=0, resulting in trajectory matching failure. To address the issue of traditional IoU metric failure, this paper designs a metric based on the Expansive Intersection Union Ratio (EIoU) region, which constructs spatiotemporal similarity between the initial non overlapping detection area and the trajectory. Without changing the original position center point, aspect ratio, and shape, the matching space between the two is expanded.

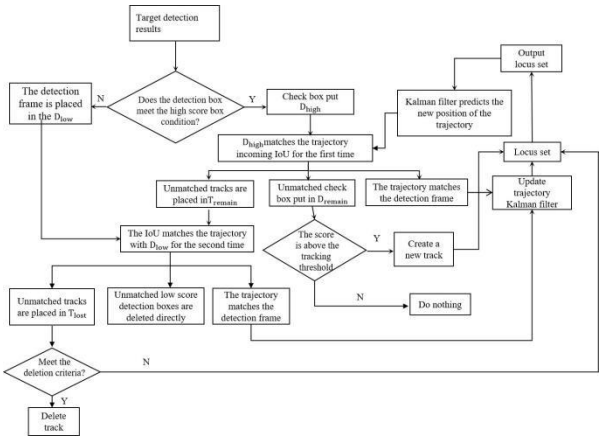

FIGURE 3. ByteTrack algorithm flowchart.

The IoU metric is shown in Figure 4, with the specific equation of:

$$IoU = \frac{A \bigcap B}{A \bigcup B} \tag{2}$$

Among them, A and B represent two original boxes, the numerator is composed of the intersection between boxes A and B, the denominator is composed of the union between boxes A and B, and the IoU indicator is obtained by the ratio between the two.

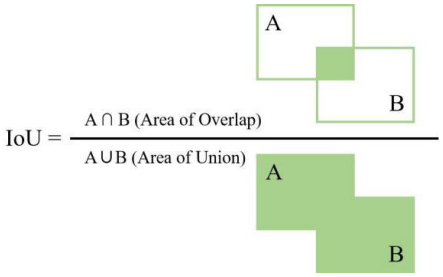

FIGURE 4. Schematic diagram of traditional IoU matching.

The calculation method based on the expansion intersection to union ratio region EIoU designed in this article involves adding a proportional expansion region around the original box, as shown in Figure 5. The specific equation is:

$$EIoU = \frac{C \bigcap D}{C \bigcup D} \tag{3}$$

Among them, C and D represent two expansion boxes based on the original boxes A and B, respectively. The molecule is composed of the intersection between C and D expansion boxes, and the denominator is composed of the union between C and D boxes. The EIoU index is obtained by the ratio between the two.

The expansion area is shown in Figure 6, and the expansion box's center coordinates match those of the original box.

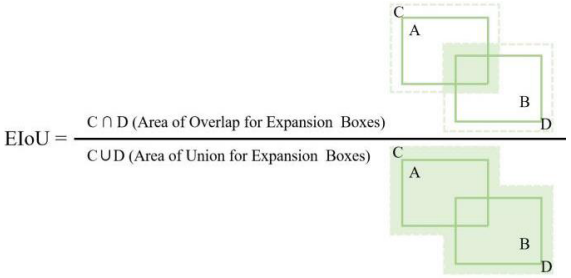

FIGURE 5. EIoU matching diagram.

The specific equation is:

$$e = \frac{E_w - w}{w} = \frac{E_h - h}{h} \tag{4}$$

where, e stands for the proportional parameter of the expansion area, E_W stands for the width of the expansion box, the expansion box's height is denoted by E_h, the original box's height by $\hbar$, and the width of the original box by W.

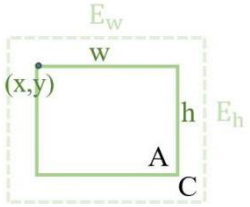

FIGURE 6. Schematic diagram of expansion area.

Assuming the original match is represented as:

$$A = (x, y, w, h) \tag{5}$$

Of them, the upper left corner of the detection box's horizontal coordinate is represented by x, while its vertical coordinate is represented by y. According to the expansion area ratio parameter e, the detection of the expansion area is expressed as:

$$C = \left(\left(x - \frac{ew}{2}\right), \left(y + \frac{eh}{2}\right), w\,(1+e), h\,(1+e)\right) \tag{6}$$

The method in this paper uses a two-stage association cascade matching strategy, with a small expansion region proportion hyperparameter e1 and a large expansion region proportion hyperparameter e2, and e1 < e2. The combination of e1 and e2 is searched in the range of $0.2 \sim 0.7$, and the best values of e1 and e2 are determined through grid search.

F. OVERALL PROCESS

With the deepening of the detection model network layer, the convolution and pooling operations increase, which is easy to cause the loss of small target features. Therefore, using the SPC-Conv module to optimize the shallow convolutional and pooling layers of the network, in order to maximize the preservation of small target feature information. The tracking part adopts ByteTrack algorithm. Select high and low confidence thresholds of 0.6 and 0.1 for two matches. If both matches fail, match them with inactive trajectories in the target tracking pool. If the detection result does not match the current tracking pool, save 30 frames.

Coordinate loss, object confidence loss and object classification loss constitute the loss function of common object detectors. In this paper, since we only target people, target confidence loss and coordinate loss make up the two halves of the loss function. The target confidence loss uses BCEWithLogitsLoss, a variant of Binary Cross-Entropy Loss (BCELoss), which combines the BCELoss and sigmoid functions and is numerically more stable than using BCELoss and sigmoid alone. The coordinate loss is based on CIoU loss, which takes into account the distance between the center point of the boundary frame and the aspect ratio, and improves the recognition ability of occlusion interference to a certain extent. Equation 7 illustrates the BCEWithLogitsLoss calculation process.

$$l = -w\left[y log\sigma\,(x) + (1-y)\log\left(1-\sigma\,(x)\right)\right] \tag{7}$$

where $\sigma\,(x) = sigmoid(x)$, x represents the prediction output, y represents the confidence label, and w is a fixed parameter, which is set when the label is unbalanced, and is set to 1 under normal circumstances.

The calculation method for CIoU loss is shown in Equations 8-11.

$$L_{CIoU} = 1 - EIoU + \frac{\rho^2\left(p, p^{gt}\right)}{c^2} + \alpha v \tag{8}$$

$$EIoU = \frac{|C \bigcap D|}{|C \bigcup D|} \tag{9}$$

$$\alpha = \frac{v}{(1 - EIoU) + v} \tag{10}$$

$$v = \frac{4}{\pi^2}(arctan\frac{E_w^{gt}}{E_h^{gt}} - arctan\frac{E_w}{E_h})^2 \tag{11}$$

where D is the expanded prediction frame, C is the expanded dimension frame, ρ is the Euclidean distance, p is the center point of the prediction frame, p^{gt} is the center point of the dimension frame, c is the diagonal distance of the smallest outer rectangular frame between boxes. α is a weight parameter, v is a parameter to measure the consistency of the aspect ratio, E_w and E_h are the width and height of the prediction box after expansion, ${E_w}^{gt}$ and E_h^{gt} are the width and height of the label box after expansion.

III. EXPERIMENTAL DESIGN

A. EXPERIMENTAL DATASET

The experiments in this paper are based on two publicly available MOT datasets.

The MOT17 dataset, one of the most widely used publicly available datasets for multi-target tracking, contains 28 video sequences, 14 of which are training sets and 14 of which are test sets. All of the sequences are provided as video frames; the total number of frames in the 28 videos is 11,235, and the

number of frames used for training and testing is 5,316 and 5,919, respectively.

MOT20 data set focuses on extremely dense crowd scenes, and its video can reach up to 246 people per frame, which can well verify the tracking effect of the algorithm in frequent occlusion and interaction scenes. MOT20 data set contains a total of 8 videos, among which 01, 02, 03 and 05 are training sets and 04, 06, 07 and 08 are test sets. All videos are provided in the form of video frames. The 8 videos contain a total of 13,410 frames, and the number of video frames for training and test are 8931 and 4479 frames respectively.

In the MOT17 and MOT20 datasets, the first half of each video is used for training and the second half is used for validation in this paper.

B. EXPERIMENTAL ENVIRONMENT AND PARAMETERS

The hardware configuration of this experiment is as follows: the CPU is Intel (R) Core (TM) i5-13400F, the main frequency is 2.50GHz, and the GPU is NVIDIA GeForce RTX4070. The experimental software environment is Win11, CUDA11.8, and Python 2.1.2. We use YOLOv8s SPD as the detector for the tracking framework. Train and evaluate on the MOT17 and MOT20 datasets without loading any pre-trained weights. In data augmentation, the image input size is set to 640 × 640, the network learning rate is set to 0.0001, the epoch is set to 100, and standard color hue, saturation, rotation angle modification are used along with concatenation and confusion techniques. In the tracking algorithm, the number of unmatched tracking frames is set to 30; the low confidence threshold is set at 0.1 while the high confidence threshold is set at 0.6.

C. EVALUATION INDICATORS

Precision (P), recall (R), and average precision (mAP) are used as performance evaluation indicators for the detector [29]. The integral method computes the precision and recall curves and the area around the axis.This is called one-class precision (AP). The mAP value can be obtained by summing the AP values of the individual categories and dividing by the total number of categories. In this paper, the value of mAP is calculated with IoU=0.5, that is, mAP@0.5 [30]. The specific equation is shown as follows:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} * 100\% \tag{12}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} * 100\% \tag{13}$$

$$\mathrm{AP} = \int_0^1 P\textcircled{R} dr * 100\% \tag{14}$$

$$\mathrm{mAP} = \frac{\sum_{i=1}^{k} AP_i}{k} * 100\% \tag{15}$$

The integration procedure in Equation 14 is to determine the area occupied by the smoothed curve $P\textcircled{R}$, which represents the smoothed precision and recall curves. In Equation 15, i is an ordinal number, AP_i represents the precision of the ith category, and k is the number of categories. In this study, k=1.

Using MOTA, IDF1, MT, ML, IDs, and FPS as performance evaluation indicators for the tracking framework. MOTA (Multiple Object Tracking Accuracy) [31] is a measure of the overall performance of a tracker for multiple object tracking, defined as follows:

$$MOTA = 1 - \frac{\sum_t (FP_t + FN_t + IDs_t)}{\sum_t GT_t} \tag{16}$$

Among them, FP_t, FN_t, IDs_t, and GT_t. respectively have the number of false detections, the number of missed detections, the number of trajectory ID conversions, and the number of true values at time t, respectively. MOTA takes into account tracking stability, accuracy, and completeness to measure the overall performance of trackers. Since FP_t, FN_t, and IDs_t. have no upper limit, the MOTA range is $(-\infty,1)$.

IDF1 (ID F1 Score) [32] takes into account both ID accuracy IDP and ID recall IDR, and has a high sensitivity to ID information during the tracking process. Its definition is as follows:

$$\begin{cases} IDF1 = \dfrac{2 \times IDP \times IDR}{IDP + IDR} \\ IDP = \dfrac{IDTP}{IDTP + IDFP} \\ IDR = \dfrac{IDTP}{IDTP + IDFP} \end{cases} \tag{17}$$

Among them, IDP and IDR represent the correct rate and recall rate of identification, while IDTP, IDFP, and IDFN represent the number of true positive IDs, false positive IDs, and false negative IDs.

MT (Mostly Tracked Targets) [33] is a high integrity tracking, which refers to the proportion of tracks with a tracking length of over 80%.

ML (Most Lost Targets) [29] is a high degree of missing tracking, which refers to the proportion of trajectories with tracking length less than 20%.

IDs (ID switches) [33] are the total number of ID transitions during the tracking process.

FPS (Frames Per Second) is the number of frames processed by the tracking framework per second.

D. YOLOv8 DETECTION RESULTS

Table 1 shows the detection results of YOLOv8 on the MOT17 and MOT20 datasets. It can be seen from this that on the MOT17 dataset, the accuracy, recall, and mAP@0.5 mAP@.5. 95 is lower than the highest values of 1.1, 0.2, 0.6, and 1.6 percentage points, respectively. On the MOT20 dataset, the accuracy, recall, and mAP@0.5 mAP@.5. 95 is lower than the highest values of 2.1, 2.2, 1.2, and 4.5 percentage points respectively, but its performance is significantly improved compared to the n model. However, from the perspective of model memory usage and GFLOPs values, the size and complexity of the s model are much smaller than those of the m, l, and x models.

TABLE 1. Performance of five parameter models in YOLOv8.

dataset	model	P/%	R/%	mAP@0.5/%	mAP@.5:.95/%	Model size/MB	GFLOPs
MOT17	YOLOv8n	80.1	76.0	85.7	59.3	6.0	8.9
	YOLOv8s	81.5	75.5	86.7	61.0	21.4	28.8
	YOLOv8m	82.2	75.6	87.0	62.0	49.6	79.3
	YOLOv8l	**82.6**	**75.7**	87.2	62.3	83.5	165.7
	YOLOv8x	82.5	75.3	**87.3**	**62.6**	130.0	258.5
MOT20	YOLOv8n	85.6	88.9	94.3	76.8	6.0	8.9
	YOLOv8s	88.7	91.0	96.1	82.3	21.4	28.8
	YOLOv8m	90.6	92.2	97.0	85.7	49.6	79.3
	YOLOv8l	**90.8**	92.3	97.1	86.0	83.5	165.7
	YOLOv8x	90.5	**93.2**	**97.3**	**86.8**	130.0	258.5

Compared with the n model, the improvement in model size and complexity is not significant. Therefore, this article uses the YOLOv8s detection network to detect targets with high accuracy and fast detection speed, providing fast and accurate detection targets for subsequent ByteTrack multi-target tracking algorithms.

E. COMPARISON OF YOLOv8s SPD DETECTION RESULTS

We adopted the same data partitioning strategy and experimental Settings with existing target detection algorithms such as YOLOv5s and YOLOv8s, and trained them with the same experimental parameters for 100 rounds respectively to test their performance. Corresponding experimental results are shown in Table 2. It can be seen from the table that compared with the YOLOv8s algorithm, the improved YOLOv8s algorithm has improved in the values of P, R, mAP@0.5 and mAP@.5:.95, and its mAP@0.5 value on MOT17 and MOT20 datasets has increased by 1% and 0.6% respectively. The experimental results show that the YOLOv8s-SPD algorithm with SPD-Conv module can realize the subsampling while retaining the target feature information as much as possible.

F. ABLATION EXPERIMENT

In this section, we take a close look at all the above methods and conduct rigorous experiments on the BCEWithLogitsLoss and EIoU methods in YBTrack based on the YOLOv8s SPD model. We conducted ablation experiments without the assistance of a pre-trained model in order to ensure fair and objective results. Of the MOT17 and MOT20 datasets, half were used for training and the other half for evaluation. Table 3 displays the results of the ablation test. Based on loss function and IoU optimization, the experimental findings show that MOTA and IDF1 on MOT17 and MOT20 data

TABLE 2. YOLOv8s SPD detection results.

data set	model	P/%	R/%	mAP@0.5/%	mAP@.5 :.95%	Model size/MB	GFLOPs
MOT17	YOLOv5s	81.9	75.3	86.5	60.5	18.5	16.5
	YOLOv8s	81.5	75.5	86.7	61.0	21.4	28.8
	YOLOv8s-SPD	**82.3**	**75.5**	**87.7**	**63.2**	25.7	184.1
MOT20	YOLOv5s	87.9	90.7	95.7	80.8	18.5	16.5
	YOLOv8s	88.7	91.0	96.1	82.3	21.4	28.8
	YOLOv8s-SPD	**89.6**	**92.0**	**96.7**	**84.6**	25.7	184.1

are, respectively, 2.8 and 2.7, 2.6 and 2.3 percentage points higher than those of prior frameworks. BCEWithLogitsLoss and EIoU methods can effectively improve the performance of YBTrack tracking framework.

TABLE 3. Ablation experiment.

data set	BCEWithLogitsLoss	EIoU	MOTA/%	IDF1/%
MOT17	√	√	**74.0**	**75.1**
	×	√	72.9	73.1
	√	×	72.7	72.8
	×	×	71.2	72.4
MOT20	√	√	**66.8**	**75.5**
	×	√	65.1	74.2
	√	×	64.9	73.9
	×	×	64.2	73.2

G. COMPARISON WITH OTHER TRACKING FRAMEWORKS

In order to make the results fair and impartial, after training the YOLOv8s-SPD model, to load the best round of pre-training weights into the tracking system, we used the same training and test data. Table 4 displays the outcomes of the experiment. Our model performs exceptionally well on the MOT17 dataset. With the exception of the fact that the tracking framework's total ID changes exceed those of the Relation Track [34] method, other performance indicators of the tracking framework are better than that of the Relation Track method. The values of MOTA and IDF1 are respectively 0.2 and 0.4 percentage points higher than those of Relation Track method, and the values of FPS are 0.4 points higher than those of Relation Track method.

In the dataset MOT20, all of our metrics outperform the LCC tracking framework compared to the RekTCL [35] and LCC [36] methods, only the total number of ID changes is slightly higher than the LCC. On the other hand, although the

RekTCL approach is faster than our approach, our framework has higher MOTA and IDF1 values, and its total target ID changes are 1.8 times higher than ours. After comparison, it can be seen that our model speed is not the highest, but combining model speed, accuracy and other important indicators data, our framework has advantages, especially IDF1 value is 5.4 percentage points higher than RekTCL method.

TABLE 4. Performance comparison with other tracking frameworks.

data set	Method	MOTA ↑ /%	IDF1 ↑ /%	MT ↑ /%	ML ↓ /%	IDs ↓	FPS ↑
MOT17	STPP[36]	52.4	-	22.4	40.0	2224	-
	TPM[37]	52.4	-	22.4	40.0	2215	-
	JDE[16]	63.0	59.5	35.7	17.3	6171	18.8
	TubeTK[38]	63.0	58.6	31.2	19.9	4137	3.0
	CTracker[18]	66.6	57.4	32.2	24.2	5529	34.4
	CenterTrack[19]	67.8	64.7	34.6	24.6	3039	22.0
	XJTU-priv[16]	68.8	64.6	40.4	23.7	2190	-
	LMOT-Tracker[39]	72.0	70.3	**45.4**	**17.3**	3071	28.6
	Relation Track[34]	73.8	74.7	41.7	23.2	**1374**	9.8
	Method of this paper	**74.0**	**75.1**	42.3	23.1	2214	10.2
MOT20	TransCenter[40]	57.1	46.7	35.7	18.0	4940	1.0
	LMOT-Tracker[39]	59.1	61.1	25.1	23.0	**1398**	22.4
	FairMOT[17]	61.8	67.3	**68.8**	**7.6**	5243	13.2
	XJTU-priv[16]	64.3	66.6	50.4	14.0	3379	-
	RekTCL[34]	65.2	70.1	61.3	10.5	4139	**22.4**
	LCC[35]	66.0	67.1	56.3	13.3	2237	-
	Method of this paper	**66.8**	**75.5**	57.1	12.0	2337	21.3

IV. CONCLUSION

As can be seen from Table 1, on MOT17 and MOT20 datasets, the values of mAP@0.5 and mAP@.5:.95 of the improved Yolov8S-SPD model increase by 1%, 2.2%, 0.9% and 1%, respectively, with higher accuracy than the original YOLOv8s model. As can be seen from Table 4, on the MOT Challenge MOT17 dataset of tracking framework.

YBTrack, the values of MOTA and IDF1 are 0.2 and 0.4 percentage points higher than those of Relation Track method respectively, and the values of FPS are 0.4 points higher than those of Relation Track method. The MOTA and IDF1 values of YBTrack on the MOT20 data set were 75.5% and 66.8%, respectively, which were 8.4 and 0.8 percentage points greater than those of the LCC method. It can be seen that YBTrack has better performance than existing tracking frameworks with built-in detectors.

V. DISCUSSIONS

In this work, we integrated SPD-Conv module into the YOLOv8s model to alleviate the phenomenon of missing detection of small targets, and conducted ablation experiments. The outcomes of the experiment demonstrated that SPD-Conv retained more feature information of targets in the process of downsampling. Considering that the tracking target may have occlusion, target interaction, irregular motion and other changing states, we optimized the traditional IoU to design EioU. The results show that the intersection ratio after expansion effectively reduces the probability of trajectory matching failure. After that, we use BCEWithLogitsLoss for loss calculation, and propose a YBTrack tracking framework with high performance, which has achieved excellent performance on both MOT20 and MOT17 datasets.

In the process of track deletion in ByteTrack algorithm, we adopt a fixed retention frame number of 30. Whether this value is the best, and whether it is more reasonable and effective to design this retention frame value as a dynamic value? In the future, we will continue to conduct more in-depth research to explore the optimal retention frame to further optimize our approach.

REFERENCES

[1] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1144–1152.
[2] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1349–1358.
[3] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
[4] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.
[5] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Computer Vision—ECCV*. Zurich, Switzerland: Springer, 2015, pp. 254–265.
[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy. Berlin, Germany: Springer Heidelberg, 2012, pp. 702–715.
[7] D. Y. Kim and M. Jeon, "Data fusion of radar and image measurements for multi-object tracking via Kalman filtering," *Inf. Sci.*, vol. 278, pp. 641–652, Sep. 2014.
[8] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 1–8.
[9] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13708–13715.
[10] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.
[11] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.
[12] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 1–21.
[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.
[14] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.
[15] P. C. Mahalanobis, "On the generalized distance in statistics," *Proc. Nat. Inst. Sci. India*, vol. 2, no. 1, pp. 49–55, 1936.
[16] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 107–122.

[17] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, Nov. 2021.
[18] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Proc. 16th Eur. Conf.*, Glasgow, U.K. Cham, Switzerland: Springer, 2020, pp. 145–161.
[19] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 474–490.
[20] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "TransTrack: Multiple object tracking with transformer," 2020, *arXiv:2012.15460*.
[21] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8834–8844.
[22] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "MOTR: End-to-end multiple-object tracking with transformer," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 659–675.
[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
[24] (Jan. 10, 2024). *Ultralytics YOLOv8*. [Online]. Available: https://github.com/ultralytics/ultralytics
[25] R. Sunkara and T. Luo, "No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2022, pp. 443–459.
[26] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.
[27] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," 2020, *arXiv:2003.09003*.
[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[29] W.-Y. Hsu and W.-Y. Lin, "Adaptive fusion of multi-scale YOLO for pedestrian detection," *IEEE Access*, vol. 9, pp. 110063–110073, 2021.
[30] Y. Liu, B. H. Lu, J. Peng, and Z. Zhang, "Research on the use of YOLOv5 object detection algorithm in mask wearing recognition," *World Sci. Res. J.*, vol. 6, no. 11, pp. 276–284, 2020.
[31] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Jan. 2008.
[32] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 17–35.
[33] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2953–2960.
[34] E. Yu, Z. Li, S. Han, and H. Wang, "RelationTrack: Relation-aware multiple object tracking with decoupled representation," *IEEE Trans. Multimedia*, vol. 25, pp. 2686–2697, 2022.
[35] W. Li, Y. Xiong, S. Yang, M. Xu, Y. Wang, and W. Xia, "Semi-TCL: Semi-supervised track contrastive representation learning," 2021, *arXiv:2107.02396*.
[36] Z. Zou, J. Huang, and P. Luo, "Compensation tracker: Reprocessing lost object for multi-object tracking," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, p. 2673.
[37] T. Wang, K. Chen, W. Lin, J. See, Z. Zhang, Q. Xu, and X. Jia, "Spatio-temporal point process for multiple object tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 1777–1788, Apr. 2023.
[38] J. Peng, T. Wang, W. Lin, J. Wang, J. See, S. Wen, and E. Ding, "TPM: Multiple object tracking with tracklet-plane matching," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107480.
[39] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "TubeTK: Adopting tubes to track multi-object in a one-step training model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6307–6317.
[40] X. Wan, S. Zhou, J. Wang, and R. Meng, "Multiple object tracking by trajectory map regression with temporal priors embedding," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1377–1386.
[41] R. Mostafa, H. Baraka, and A. Bayoumi, "LMOT: Efficient lightweight detection and tracking in crowds," *IEEE Access*, vol. 10, pp. 83085–83095, 2022.
[42] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "TransCenter: Transformers with dense queries for multiple-object tracking," 2021, *arXiv:2103.15145v4*.

YINGYUN WANG was born in Tongling County, Anhui Province. She received the master's degree in computer technology from Anhui University, in July 2012. She is currently pursuing the Ph.D. degree in computer science with National University, Philippines. She is currently working as an Associate Professor with Anhui Xinhua University. Her research interests include machine learning, artificial intelligence, and network security.

VLADIMIR Y. MARIANO (Member, IEEE) received the B.S. degree in statistics and the M.S. degree in computer science from the University of the Philippines Los Banos and the Ph.D. degree in computer science and engineering from The Pennsylvania State University. He is currently working as a Professor with the Faculty of Computing and Information Technologies, National University, Philippines. His research interests include computer vision, digital image processing, and machine learning.