Feng X, Wu HM, Yin YH *et al.* CGTracker: Center graph network for one-stage multi-pedestrian-object detection and tracking. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 37(3): 626–640 May 2022. DOI 10.1007/s11390-022-2204-8

CGTracker: Center Graph Network for One-Stage Multi-Pedestrian-Object Detection and Tracking

Xin Feng (冯　欣), *Senior Member*, *CCF*, *Member*, *IEEE*, Hao-Ming Wu (吴浩铭), Yi-Hao Yin (殷一皓), and Li-Bin Lan (兰利彬), *Member*, *CCF*

College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China

E-mail: xfeng@cqut.edu.cn; 52190325121@2019.cqut.edu.cn; 51180324103@2018.cqut.edu.cn; lanlbn@cqut.edu.cn

Received February 3, 2022; accepted May 6, 2022.

Abstract Most current online multi-object tracking (MOT) methods include two steps: object detection and data association, where the data association step relies on both object feature extraction and affinity computation. This often leads to additional computation cost, and degrades the efficiency of MOT methods. In this paper, we combine the object detection and data association module in a unified framework, while getting rid of the extra feature extraction process, to achieve a better speed-accuracy trade-off for MOT. Considering that a pedestrian is the most common object category in real-world scenes and has particularity characteristics in objects relationship and motion pattern, we present a novel yet efficient one-stage pedestrian detection and tracking method, named CGTracker. In particular, CGTracker detects the pedestrian target as the center point of the object, and directly extracts the object features from the feature representation of the object center point, which is used to predict the axis-aligned bounding box. Meanwhile, the detected pedestrians are constructed as an object graph to facilitate the multi-object association process, where the semantic features, displacement information and relative position relationship of the targets between two adjacent frames are used to perform the reliable online tracking. CGTracker achieves the multiple object tracking accuracy (MOTA) of 69.3% and 65.3% at 9 FPS on MOT17 and MOT20, respectively. Extensive experimental results under widely-used evaluation metrics demonstrate that our method is one of the best techniques on the leader board for the MOT17 and MOT20 challenges at the time of submission of this work.

Keywords pedestrian detection and tracking, object center, object graph

1 Introduction

Online multi-object tracking (MOT) aims to take advantage of the object information contained in the previous and the current frame to match the objects across different frames in a video stream, and the motion trajectories of different objects can thus be derived according to the cross-frame matching results. Since only the information of the current and previous frames can be used, it is extremely challenging for online tracking methods to satisfy both the high tracking accuracy and the low time delay.

Currently, the MOT task is mainly solved by the tracking-by-detection framework [1–3]. In this framework, video frames are first inputted into an object detection module to recognize and locate objects frame by frame, and a data association module [3–5] is then used to associate the same object across different frames. Although a considerable progress has been made in the field of MOT in the past few years, the existing MOT methods still have two problems.

1) Data association often depends on the quality of object detection. Therefore, in order to obtain a good performance of data association, most tracking-by-detection methods use anchor-based object detec-

Regular Paper

Special Section of CVM 2022

This work is partially supported by Humanities and Social Sciences of Chinese Ministry of Education Planning under Grant No. 17YJCZH043, the Key Project of Chongqing Technology Innovation and Application Development under Grant No. cstc2021jscx-dxwtBX0018, and the Scientific Research Foundation of Chongqing University of Technology under Grant No. 0103210650.

tion methods [6–9], which greatly increases the time cost of the entire tracking solution. In addition, existing trackers often adopt a pre-trained feature embedding network to extract discriminative feature representation of detected objects for object association. However, this multi-stage pipeline network not only makes model more complex, but also reduces the tracking efficiency.

2) Most MOT methods focus on associating objects based on appearance features of the detected objects through Intersection over Union (IOU) [10]. This data association, however, does not consider the spatial relationships between different objects in the same frame and same objects in the consecutive frames.

The pedestrian is the most common and major object category in real-world scenes. Especially, pedestrian detection and tracking is the key and fundamental technique for many applications, such as auto-driving and video surveillance. As multiple pedestrian targets often appear in the visual scenes in company, pedestrian tracking is taken as one of the main problems of MOT. In order to realize highly efficient and accurate online multi-pedestrian tracking, we design a novel one-stage multi-object detection and tracking method to jointly optimize the pedestrian detection and tracking tasks in a unified framework, which we term as Center Graph Tracker (CGTracker). CGTracker takes two consecutive frames as input, and both of the frames perform the center point based object detection to recognize, localize and extract features of the objects simultaneously. By considering the continuous property of the spatial relationship between pedestrians in a short time interval, an object graph is then constructed from the extracted pedestrian features and the spatial relationship between objects in a frame and across frames to learn the object association under the high accuracy and the low time delay objectives of MOT.

In the tracking-by-detection based MOT implementation, object detection aims to provide accurate object localization and discriminative feature representation for subsequent data association. Recent MOT methods usually apply generic anchor based object detectors, e.g., Faster RCNN [6], YOLO [7–9], to locate objects as regular bounding boxes in an image frame. These detectors, on the one hand, need to generate lots of region proposals or anchors, which does not consider the requirement of the downstream MOT task and brings additional computational redundancy. On the other hand, the detected bounding boxes contain redundant information than the object location only, e.g., some background pixels. In fact, the object detection for MOT does not require to detect the entire object body. It is sufficient to use some key point as the object location representation for MOT, especially for pedestrians.

Moreover, as demonstrated in anchor-based object detection methods [6–9], high-level features extracted from the backbone network contain the representative information of objects. Hence, the feature points corresponding to the detected anchors and the resulted objects are effective object feature representation. Following this idea, we propose to extract the features of the detected objects directly from the multi-scale features of the backbone network according to the detected object center points. As a result, the pedestrian detection module in our CGTracker would provide both the object location and the corresponding feature representation required by the subsequent multi-object association process. This facilitates efficient one-stage multi-pedestrian-object detection and tracking implementation.

Furthermore, most of current MOT methods only consider the appearance feature of the object for object association. But we believe that besides the appearance feature, the relative relationship between pedestrians in the same frame and the temporal correlation between the same identity in consecutive frames are also important tracking cues. Hence, inspired by the object graph representation for videos [11], we build an object graph based on the detected objects for each frame, and convert the object association problem in MOT into the graphs matching process. Specifically, we denote both the appearance feature and the position of the object as the node description, and the position difference between two pedestrians in a frame as the edge description of the object graph. We then consider the association process as the matching between two object graphs, where the appearance matching between nodes of the two graphs, the edge matching between the edge description of the two graphs, and the relative displacement matching between the nodes of the two graphs are fused together to derive the final MOT results.

To summarize, our main contributions are as follows.

1) We propose a simple yet effective one-stage tracking method that combines both multi-object detection and data association modules in a unified framework, which we name as CGTracker.

2) In CGTracker, we propose to detect a pedestrian object as a center point, and directly extract the object feature based on the center point from the multi-

scale feature representations of the backbone network to realize the highly efficient one-stage multi-pedestrian detection and tracking framework.

3) CGTracker proposes to build an object graph based on the detected pedestrians in a video frame by considering the continuous property of spatial relationship between objects in a short-time period to improve the tracking accuracy. The multi-object tracking is then converted into the matching between two object graphs of two consecutive frames from three aspects: the appearance association for nodes between the two graphs, the relative relationship similarity for edges between the two graphs, and the displacement constraints between the nodes (objects) of the two graphs.

4) Extensive experiments are performed on the widely-used MOT datasets: MOT17 and MOT20. Results demonstrate that CGTracker is a highly efficient and accurate multi-pedestrian detection and tracking method.

The rest of this paper is organized as follows. In Section 2, we introduce some latest work in the field of MOT. In Section 3, we describe the implementation of our proposed joint detection and tracking method in detail. In Section 4, we present the experimental details, ablation study, and comparison results on the widely-used benchmarks MOT17 and MOT20. Finally, Section 5 draws conclusions of this paper.

2 Related Work

In recent years, with the development of deep learning, MOT techniques have also made great progress. The existing MOT methods are mainly divided into the following research directions.

Tracking-by-Detection Method. DeepSORT [12] is the first deep learning based tracking-by-detection MOT method. It applies the two-stage object detection method "Faster R-CNN" for detection, a pre-trained network for object feature extraction and the Kalman filter to realize the whole MOT process. Yu *et al.* [13] then showed that high-performance detection and appearance features that are extracted from multi-scale deep neural network layers are significant factors to improve MOT results in both online and offline tracking. These tracking-by-detection based methods, however, have some weakness. 1) The overall tracking performance is highly dependent on the detection results. 2) There are several independently trained modules, such as detection, feature extraction and data association in the MOT pipeline, which makes the whole MOT system complex and time-consuming.

Partially End-to-End MOT Method. In this strategy, researchers mainly combine object detection, feature extraction, and data association to form a partially end-to-end method. Sun *et al.* [14] proposed to perform an end-to-end data association by modeling the appearance and learning the affinity between the targets in different frames. Wang *et al.* [15] proposed a joint detection and embedding MOT paradigm by incorporating the embedding learning into the object detector for fast MOT systems. Similarly, Lu *et al.* [16] proposed single-stage RetinaTrack by improving the single-stage RetinaNet, which combines target detection with feature extraction. Zhu *et al.* [17] combined the Bi-LSTM network with an attention mechanism to achieve an end-to-end matching attention network. Although these methods attempt to optimize some modules of MOT in the end-to-end manner, they do not incorporate the entire detection and association learning in a unified framework for more efficient and accurate MOT systems.

Joint Detection and Tracking Methods. In the above tracking methods, object detection and tracking are often separated so that the global optimal result cannot be obtained. In recent years, a new multi-object tracking idea that jointly realizes multi-object detection and tracking in a unified framework has emerged. For examples, Chained-Tracker [18] converts the data association between consecutive frames into a paired object detection problem, and achieves multi-object tracking by linking the results of object detection in previous and subsequent frames. CenterTrack [19] predicts the offsets between objects in current frame and those in the previous frame, and associates the predicted objects with previous ones through the predicted offsets to achieve the final multi-object tracking. The very recent work FairMOT [20] performs object detection based on anchor-free detection method [21]. And with adding an object feature embedding head on the object detection network, FairMOT directly outputs both the object detection result and the object feature embedding. Finally, by using some post-processing methods, such as the Hungarian algorithm, Kallman filter and so on, the final tracking results can be obtained.

3 Methodology of CGTracker

Given a sequence of video frames, the goal of the MOT task is to associate the same identity in different frames and assign it a unique trajectory ID. Existing MOT methods mainly divide the task into three parts: object detection, feature extraction and object association. These methods, however, often simply apply

generic methods to implement each step, without fully investigating the characteristics of the object category for detection and tracking, especially for the commonly appearing pedestrians. By exploring the advantage of the center point based object detection method and the relationship of the detected pedestrians in a frame and across frames, we propose a center graph neural network for one-stage multi-pedestrian-object detection and tracking, referred to as CGTracker, which unifies object detection and object association into a single framework. In the following subsections, we introduce the pipeline of our method, and describe the proposed multi-object detection and association modules, respectively.

3.1 Architecture of the Proposed Method

CGTracker aims to realize highly efficient deep learning based pedestrian multi-pedestrian-object detection and tracking to facilitate online tracking for real-time applications, e.g., pedestrian detection tracking in autonomous driving. The method takes two frames with interval n in the training phase as input, and the multi-object detection and the multi-object tracking are mainly implemented in the center point based object detection and the object graph based association modules respectively. The entire framework is shown in Fig.1.

First, in order to render a more effective pedestrian detection for multi-object tracking, we propose to detect the object as the center point by following the idea of CenterNet [21], which is the center point prediction module in Fig.1. Because the multi-object tracking eventually relies on object feature association, the highly discriminative feature representation of the detected objects is very important for accurate MOT. Since using extra feature extraction is time-consuming, in CGTracker, we propose to extract multi-scale features from the backbone network, which is the DLA34 network proposed in [22], according to the object center point coordinate P_t of the t-th frame. The N_m multi-scale feature maps are then fused effectively to represent the appearance feature of the detected object. As a result, the object detection module in CGTracker will output both the pedestrian center-point coordinates and the representative appearance features. The joint detection and feature extraction process facilitates the one-stage object detection and tracking implementation and is expected by the subsequent object association step for high efficient MOT.

In the data association process, unlike recent object association methods that mainly rely on the appearance of the object, CGTracker proposes to construct an object graph based on the center points of all detected pedestrians for each frame, so as to effectively combine the relative position constraints between pedestrians in a frame, and the displacement constraints be-

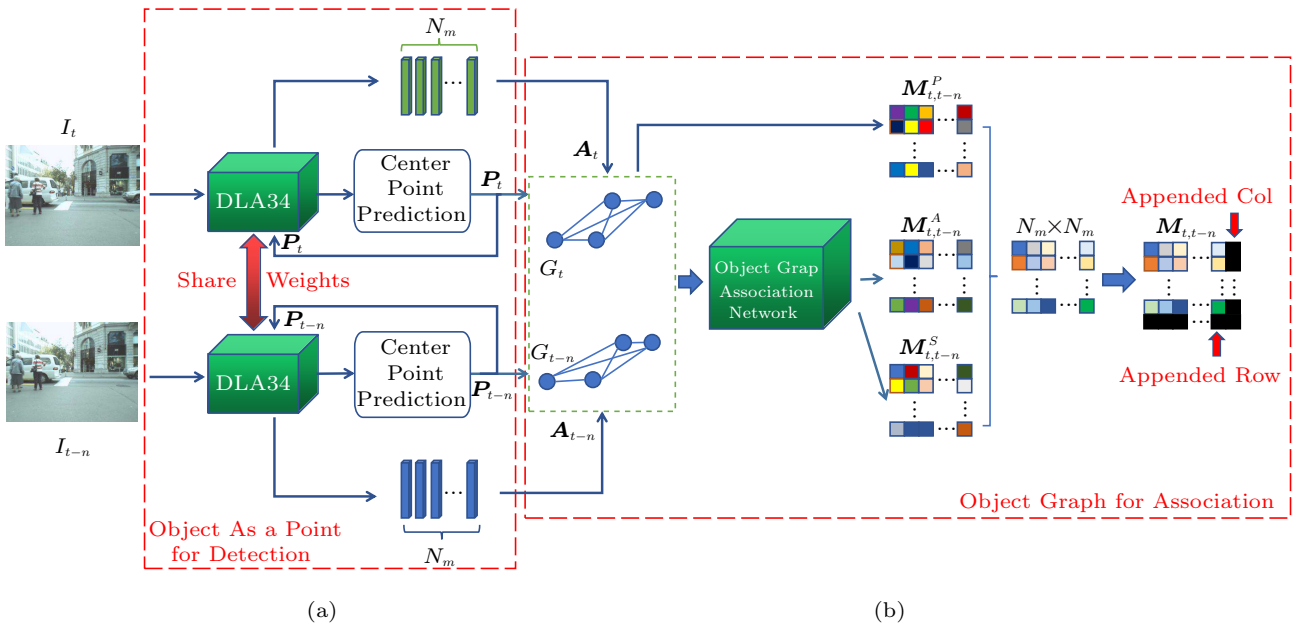


Fig.1. Architecture of the proposed CGTracker, consisting of (a) an object as a point for detection and (b) object graph based data association. Col: column.

tween objects across different frames, in addition to the appearance feature associations between different frames. As shown in Fig.1, two object graphs G_t and G_{t-n} are constructed for frames I_t and I_{t-n} respectively. The nodes in each graph encode the detected objects, and each node is described by the fusion of the object appearance feature $\boldsymbol{A}_t$ (or $\boldsymbol{A}_{t-n}$) and the position feature $\boldsymbol{P}_t$ (or $\boldsymbol{P}_{t-n}$). The edges of a graph encode the spatial relationship between different pedestrians in the frame. The two object graphs then facilitate an object graph association network to realize the multi-constraint data association between frames I_t and I_{t-n} through the matching of nodes appearance similarity $\boldsymbol{M}^A_{t,t-n}$, edge (or structure) similarity $\boldsymbol{M}^S_{t,t-n}$ and nodes displacement similarity $\boldsymbol{M}^P_{t,t-n}$. At last, the three matching matrices are then integrated to generate the object association result $\boldsymbol{M}^P_{t,t-n}$ and the final result of object tracking is obtained by the Hungarian algorithm. Here, we add an additional column and row on the matching matrix $\boldsymbol{M}^P_{t,t-n}$ to deal with the newly entered pedestrians and disappeared ones in frame I_t, respectively.

3.2 Object As a Point for Detection

As aforementioned, the pedestrian detection for MOT does not need to detect an object as a regular bounding box, and some key point that is able to represent the location and salient features of the object is sufficient. Therefore, different from recent tracking-by-detection based MOT methods that simply adopt generic object detection, CGTracker explores the ways to detect a pedestrian as a point.

As is well known, the center of an image region is the most representative point. In addition, there are many saliency-based object detection methods considering the center point and its surroundings as the most salient representation of an object [23]. On the other hand, the recent anchor-free based deep learning methods [21, 24, 25] have greatly advanced the object detection field. These methods learn to detect objects as key points, and have shown to be more efficient than the two-stage anchor based object detection methods and more accurate than anchor-based one-stage object detection methods. Inspired by these techniques, we propose to detect the center point of a pedestrian for the object detection of MOT. By following CenterNet [21], our center-point based object detection does not require preset anchors and the undifferentiable NMS [26] operation, but learns to locate the center point that is described by a set of neighbor points in the end-to-end manner, which greatly improves the detection and MOT efficiency.

Besides object localization, MOT needs to associate the same identity in different frames based on the feature representation of the object. Instead of applying an extra object re-identification network for the object association, we propose to extract feature representation from the backbone network of object detection according to the center point coordinate of an object. Specifically, we use the deep layer aggregation (DLA) [22] network as the backbone for pedestrian feature extraction. As shown in Fig.1, DLA is a network building in the tree structure, which can deeply aggregate multi-scale object features from low-level to high-level convolution layers.

In CGTracker, the two consecutive frames are first fed into the DLA network for feature extraction. And inspired by [5], we intentionally make the two-stream DLA network with shared weights. After the inference on the center point based object detection network, the center position of the detected pedestrians can be located. We then trace back to the backbone network to search for the best region of interest (RoI) feature representation according to the center location of the object $\boldsymbol{P}_t$. It is shown that high-level semantic features are good representation for object recognition, while the data association in the MOT task requires feature representation that can distinguish different objects, instead of identifying the object category. Hence, we propose to extract the multi-scale features from different down-sample layers of the DLA network, as shown in Fig.2. The extracted feature tensors are then passed to an additional 3×3 convolution layer and aggregated to be the final appearance feature representation for object association.

3.3 Object Graph for Association

In real-world scenes, multiple pedestrians often appear in crowds and groups. Although some of the objects may be occluded or motion-blurred at some time t so that their trackers get lost, their relative positions, in other words, the spatial relationship between objects, would be maintained in a short time period. This observation motivates us to investigate the temporal continuity of both individual object motion and the relationship between objects in the same frame.

In CGTracker, we construct an object graph G_t for each frame I_t at time t, where the node of the object

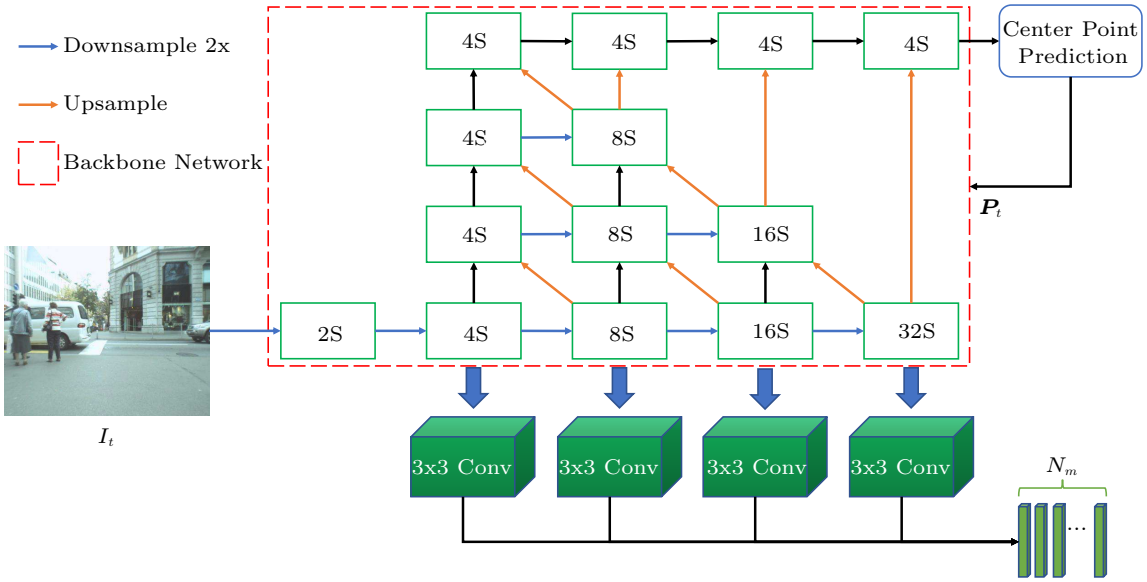


Fig.2. Illustration of the center point based object detection and multi-scale feature extraction from DLA [22]. The red frame indicates the structure of the backbone network DLA [22], where each green box represents the process of extracting high-level features from the initial feature layer through multiple convolutional layers. The solid green cubes indicate the extracted feature tensors. Conv: convolutional layer.

graph is composed of the object feature descriptors, and the edge is represented by the relative position between objects. Specifically, each node O_t^i in G_t is described by the appearance features $\boldsymbol{A}_t^i \in \mathbb{R}^{520}$ and the position information $\boldsymbol{P}_t^i \in \mathbb{R}^2$ of object i. In addition, the edge $\boldsymbol{E}_t^{i,j} \in \mathbb{R}^2$ of object graph G_t is described by the difference between center coordinates of the detected objects i and j. As illustrated in Fig. 3, two object graphs G_t and G_{t-n} are derived from frames I_t and I_{t-n} respectively, where $G_t = (O_t, E_t)$, with

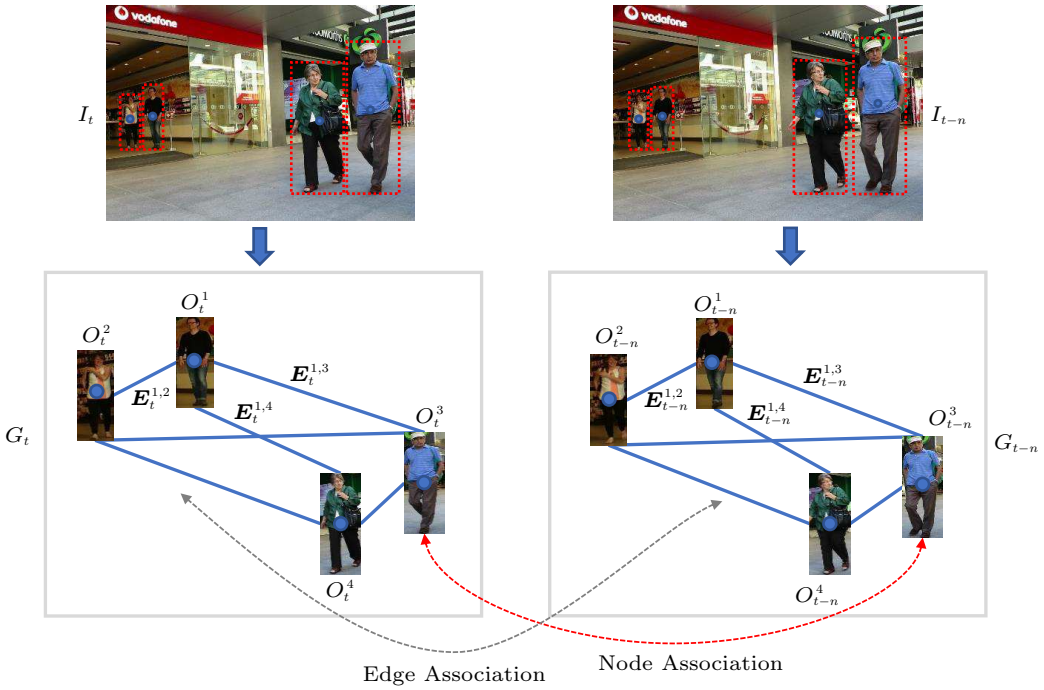


Fig.3. Illustration of two object graphs constructed from the t-th frame and the $(t-n)$-th frame. The solid blue lines in each object graph are the edges between adjacent objects, the red dash lines denote the object node correspondences, and the gray dash lines indicate the edge correspondences.

$O_t = \{(\boldsymbol{A}_t^i, \boldsymbol{P}_t^i)\}_{i=1}^{N_m}$ and $E_t = \{(\boldsymbol{E}_t^{i,j})\}_{i,j=1}^{N_m}$, and N_m denotes the maximum number of objects detected in frame I_t.

With the object graph representation for each frame, the MOT task can thus be translated into a graph matching process through the optimization of both node-to-node and edge-to-edge association between two consecutive frames.

3.3.1 Node Association

Based on the object graph for each frame, we perform node matching to realize object association for multi-pedestrian tracking. Node association is carried from the matching of nodes descriptors: the appearance feature $\boldsymbol{A}_i$ of object i and the position displacement $\boldsymbol{P}_i$ of object i in consecutive frames.

As shown in Fig.4, the nodes in the object graph for frame I_t are associated with the corresponding objects nodes in the object graph of frame I_{t-n}, which is the association results learned through the appearance similarity and the displacement similarity between objects in frame I_t and frame I_{t-n}.

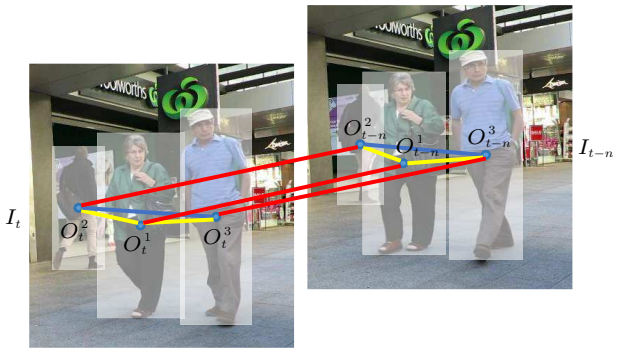


Fig.4. Node association and edge association between the two object graphs of I_t and I_{t-n}, respectively. The red solid line represents the association result of object nodes that are successfully matched by the node association strategy; the yellow solid line represents the structural similarity information of the object nodes learned by the edge association strategy.

Appearance Association of Object Nodes. The appearance feature of each detected object is extracted from multi-scale CNN layers of the backbone network according to the center position of the object, as shown in Fig.2. The selection of multi-scale CNN layers will be discussed in Subsection 4.4. The appearance features of all the objects in frame I_t are aggregated, and through the one-to-one correspondence of the appearance features of the objects in the two object graphs, an appearance feature matrix $\boldsymbol{A}_{t-n,t}^N \in \mathbb{R}^{1\,040 \times N_m \times N_m}$ is obtained. This matrix is then fed into a node association network, which is composed of five 3×3 convolution (conv) layers with $(512, 256, 128, 64, 1)$ as the channel number of each layer, to learn the appearance similarity matrix $\boldsymbol{M}_{t-n,t}^A \in \mathbb{R}^{N_m \times N_m}$ under the MOT objective.

Position Association of Object Nodes. As is known, the movement of pedestrians is temporally coherent. This means that there are few changes in the position of an object in a short-time period. We then consider measuring the displacement similarity between objects in different object graphs of consecutive frames. In special, the position distance between objects of the same identity in consecutive frames would be smaller than that between the objects of different identities. Hence, we calculate the position distance between all nodes in two consecutive object graphs and form the position similarity matrix $\boldsymbol{M}_{t,t-n}^P$, where each item is computed as:

$$\boldsymbol{M}_{i,j}^P = \frac{\mathrm{e}^{-d_{i,j}/Dia(I)} - \mathrm{e}^{-1}}{1 - \mathrm{e}^{-1}}, \tag{1}$$

where $d_{i,j}$ is the Euclidean distance between the center position of the i-th object node in frame I_{t-n} and the j-th node in frame I_t. By taking the length of image diagonal $Dia(I)$ as the largest distance between object i in frame I_t and corresponding object j in frame I_{t-n}, $d_{i,j}$ is first normalized by $Dia(I)$ to the range of $[0, 1]$. Here, we do not normalize $d_{i,j}$ by the relative largest distance between objects in two consecutive frames, because we tend to normalize the movements of all the objects over time with respect to the largest distance across the entire video so that the whole tracking trajectory is smoothly correlated. The normalized $d_{i,j}$ is then converted to the similarity measurement by the exponential decay function in (1).

3.3.2 Edge Association

In the multi-object tracking scenario, a moving pedestrian often moves along certain direction in a short time. If we consider the relative relation between a pedestrian and other pedestrians in a frame, such as pedestrian A at the northwest of pedestrian B at time t, this relation will be maintained in a short-time period, e.g., two consecutive frames in online tracking. Therefore, besides tracking over individual moving object, we propose an additional tracking objective by taking the relationship consistency over time into account.

As shown in Fig.3, based on the edge descriptor that calculates the direction vectors between objects in the same frame, CGTracker performs the edge-to-edge association between consecutive object graphs to realize

the relationship correspondence of pedestrians. In addition, the learning process of edge association is illustrated in Fig.5. $\boldsymbol{S}_t^i \in \mathbb{R}^{320}$ denotes the aggregated descriptors of all edges that are connected to object i, and by combining the edge descriptors of all edges of both object graphs, we derive the relation structure matrix $\boldsymbol{S}_{t-n,t}^E \in \mathbb{R}^{320\times N_m\times N_m}$. Similar to node association, we construct an edge association network to learn the relation structure similarity matrix $\boldsymbol{M}_{t-n,t}^S \in \mathbb{R}^{N_m\times N_m}$, which also consists of five 3×3 convolutional layers with $(160, 80, 40, 20, 1)$ as the channel number for each layer.

Finally, by comprehensively fusing the node association and edge association results of the two consecutive object graphs, we obtain the final object incidence matrix:

$$\boldsymbol{M}_{t-n,t} = (\boldsymbol{M}_{t-n,t}^A + \boldsymbol{M}_{t-n,t}^S) \odot \boldsymbol{M}_{t-n,t}^P,$$

where $\odot$ represents the dot product between two matrices. In order to solve the object entering or leaving problem in consecutive frames, we add an extra row and column to $\boldsymbol{M}_{t-n,t}$, and obtain the final object association matrix $\boldsymbol{M}_{t-n,t} \in \mathbb{R}^{(N_m+1)\times(N_m+1)}$ followed by the row and column regularization for MOT optimization, as shown in Fig.5.

3.4 Network Loss

In order to facilitate the whole network for learning, we optimize the object detection loss for object classification and center localization, and the graph association loss for multi-object association for MOT.

Object Detection Loss. We follow the object learning strategy of CenterNet [21] to predict the object center, which is mainly carried out by combining the prediction of the object category and the regression of the center location. In order to recognize the pedestrian and localize the object center, we use the Gaussian kernel function: $\boldsymbol{H}_{xyc} = \exp(-\frac{(x-\lfloor\frac{x_k}{r}\rfloor)^2+(y-\lfloor\frac{y_k}{r}\rfloor)^2}{2\sigma_k^2})$, to distribute the centers of all ground truth (GT) targets on the heatmap, $\boldsymbol{H} \in \mathbb{R}^{\frac{W}{R}\times\frac{H}{R}\times C}$ where R is the number of down-sampling operations, r is the r-th down-sampling pooling in the network, (x_k, y_k) is the center coordinate of the GT object k, and σ_k is an object size-adaptive standard deviation [24].

With the Gaussian-based center point representation, we optimize the loss between the predicted and the GT center category by following the focal loss in [21] to derive L_{cls}. And we use the L1 regularization to calculate the loss L_{size} between the GT size and the predicted size, and the object center offset loss L_{off}. In summary, the overall object detection learning objective L_{det} is as follows.

$$L_{\text{det}} = \lambda_1 L_{\text{cls}} + \lambda_2 L_{\text{size}} + \lambda_3 L_{\text{off}},$$

where $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 0.1$.

Object Association Loss. For object association, we mainly follow the loss function designed in DAN [14]. Specifically, our loss function combines the following four considerations.

1) *Forward Association Loss* L_1. We first learn to associate objects forwardly from frame I_{t-n} to frame

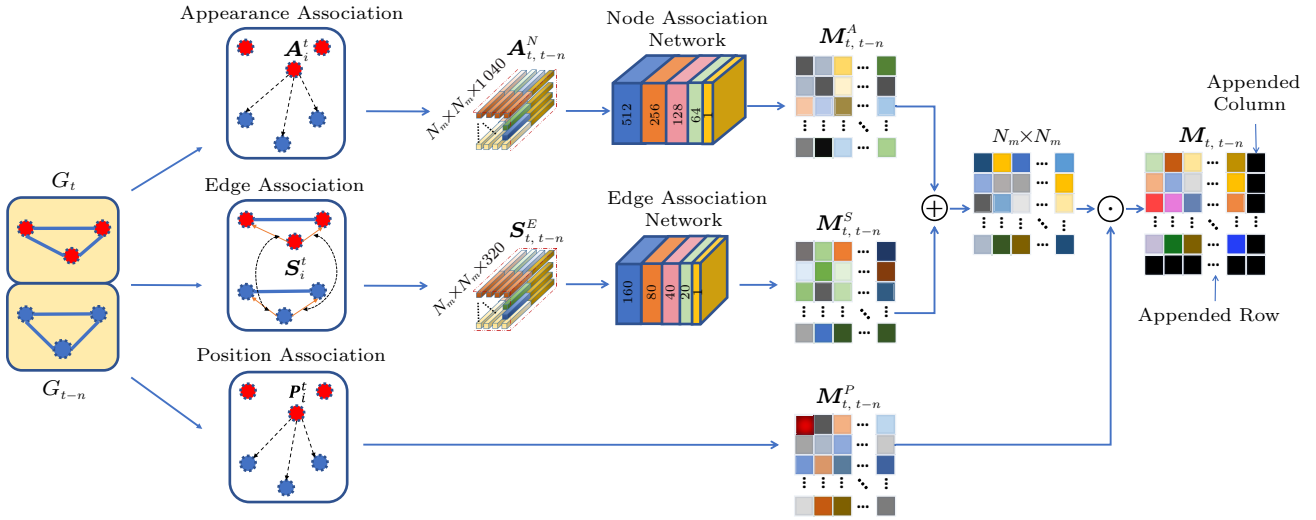


Fig.5. Object graph association network structure diagram. After constructing the object in the t-th frame image I_t and the $(t-n)$-th frame image I_{t-n} into the object graphs G_t and G_{t-n} respectively, we use the information such as appearance, displacement, and relative position, and use different association strategies to obtain the association matrix $\boldsymbol{M}_{t,t-n}$ of the pedestrians in the two frames.

I_t. Let us denote $\boldsymbol{M}_1 \in \mathbb{R}^{N_m \times (N_m+1)}$ as the first m rows of data of the object incidence matrix $\boldsymbol{M}_{t-n,t} \in \mathbb{R}^{(N_m+1)\times(N_m+1)}$, with N_m+1 representing the maximum number of objects in a frame plus an extra column of the newly entered target in I_t. The forward association objective can thus be supervised by the one-to-one correspondence matrix $\boldsymbol{G}_t \in \mathbb{R}^{N_m \times (N_m+1)}$ constructed from the tracking ground truth of objects in frame I_{t-n} to objects in I_t as:

$$L_1 = \frac{\sum\limits_{coeff} (\boldsymbol{G}_t \odot (-\log(S(\boldsymbol{M}_1))))}{\sum\limits_{coeff} (\boldsymbol{G}_t)}, \tag{2}$$

where S is the softmax function, *coeff* represents the summation of all the coefficients of a matrix, and $\odot$ is the Hadamard product.

2) *Backward Association Loss* L_2. In order to learn more accurate data association results, we further consider the backward object association from frame I_t to frame I_{t-n}. The ground truth matrix $\boldsymbol{G}_{t-n} \in \mathbb{R}^{(N_m+1)\times N_m}$ is constructed from the one-to-one correspondence of objects in frame I_t to frame I_{t-n}, with N_m+1 here representing the maximum number of objects in a frame plus an extra row of the disappeared target in I_t. The backward association loss L_2 is then calculated as:

$$L_2 = \frac{\sum\limits_{coeff} (\boldsymbol{G}_{t-n} \odot (-\log(S(\boldsymbol{M}_2))))}{\sum\limits_{coeff} (\boldsymbol{G}_{t-n})}, \tag{3}$$

where $\boldsymbol{M}_2 \in \mathbb{R}^{(N_m+1)\times N_m}$ represents the first m columns of data of the object incidence matrix $\boldsymbol{M}_{t-n,t} \in \mathbb{R}^{(N_m+1)\times(N_m+1)}$.

3) *Consistency Judgment Loss* L_3. Basically, the forward and backward association between objects in frames I_t and I_{t-n} would be consistent; hence, we formulate the bi-direction association consistency between (1) and (2) as:

$$L_3 = \left\| \widehat{S(\boldsymbol{M}_1)} - \widehat{S(\boldsymbol{M}_2)} \right\|_1. \tag{4}$$

4) *Joint Judgment Loss* L_4. Similar to [14], we perform the non-maximum suppression for both forward and backward object association results, which is formulated as:

$$L_4 = \frac{\sum\limits_{coeff} (\boldsymbol{G}_{t-n,t} \odot (-\log(\max(\widehat{S(\boldsymbol{M}_1)}, \widehat{S(\boldsymbol{M}_2)}))))}{\sum\limits_{coeff} (\boldsymbol{G}_{t-n,t})}. \tag{5}$$

By combining the four loss functions (2), (3), (4) and (5), we have the overall object association loss as:

$$L_{\text{ass}} = \frac{L_1 + L_2 + L_3 + L_4}{4}.$$

Finally, the total loss of CGTracker can be summarized as:

$$L_{\text{all}} = \eta_1 L_{\text{det}} + \eta_2 L_{\text{ass}}.$$

According to our experimental results, the hyper-parameters of η_1 and η_2 can be set as $\eta_1 = 1$ and $\eta_2 = 0.1$ for the best results.

4 Experiments

4.1 Dataset

We conduct experiments on the widely-used Multi-Object Tracking (MOT) benchmarks: MOT17 [27] and MOT20 [28]. MOT17 contains seven training sequences and seven testing sequences, and these videos are mainly from still or moving cameras in unconstrained environments. Pedestrians in the scene have frequent access, crowding and occlusion, and the frame rate is 25 FPS–30 FPS. MOT20 is the newly released pedestrian multi-object tracking challenge, which consists of four training sequences and four testing sequences. Compared with MOT17, the pedestrians in the MOT20 scene are more crowded and difficult for tracking. The video sequences used for training the model all provide accurate annotations, and the detection results from three different detectors, namely DPM [29], SDP [30], and Faster R-CNN [6]. For a fair comparison, labels of test data are not publicly released. Since the dataset does not provide an official validation set, we split the training data into training sets and validation sets respectively, each containing roughly half of the whole training data, where the first half frames are used for training, and the second half for validation. Because of the limited access to the test server, we evaluate our main results on the test set, but the other results on the validation set, e.g., ones from ablation experiments.

4.2 Evaluation Metrics

In order to evaluate the performance of object detection module, we use the widely-used metrics: average precision (AP), precision (Prcn) and recall rate (Rcll) to compare our proposed CGTracker with other algorithms. At the same time, in order to quantitatively evaluate the MOT results on the MOT challenge, we apply the official evaluation standard CLEAR

MOT metrics [31], including the multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP), false positives (FP), false negatives (FN), identity switches (IDSw) and IDF1 scores. In addition, evaluation criteria such as the percentage of mostly tracked targets (MT) and the percentage of mostly lost targets (ML) have also been adopted. MT refers to the ratio of ground-truth trajectories that are covered by any track hypothesis for at least 80% of their respective life span. ML is computed as the ratio of ground-truth trajectories that are covered by any track hypothesis for at most 20% of their respective life span. Recently, a new evaluation metric HOTA [32] has been proposed to evaluate the higher-order tracking association accuracy.

4.3 Implementation Details

We implement our proposed approach using the Pytorch framework [33]. Similar to recent FairMOT [20], we first perform pre-training for object detection module in CGTracker on multiple object detection datasets, such as Crowdhuman [34], Widerperson [35] and CityPersons [36]. The whole training is performed on an NVIDIA GeForce RTX 2080ti GPU with standard stochastic gradient decent (SGD) for 35 epochs. The input images are all resized to 544×960. Other hyper-parameters used in our implementation include the batch size `batch_size=3`, the maximum number of object detection per frame $N_m = 80$, and the initial learning rate `learning_rate=0.01`. The learning rate is decreased by 10 at the 13th, 22nd, 28th, and 35th epoch. During training, we select targets with visibility greater than 0.3 for association, and the maximum time interval between two frames $n = 30$. In the inference stage, we set n to 1 to associate objects between two consecutive frames.

4.4 Results and Analysis

In this subsection, we intend to evaluate the performance of our proposed method from the following three aspects. First, we compare the performance of different detectors on the MOT results. Second, we prove the effectiveness of the selected semantic features for object feature representation. Finally, we make ablation study of our CGtracker under different constraints and comparison with other methods on MOT17 and MOT20 challenges respectively. Note that for all Tables 1–6, the symbol ↑ indicates that higher is better, ↓ means that lower is better. The best result is highlighted in bold.

Detection Results on Tracking Task. We compare our proposed object detection method with the three public detection results provided on MOT Challenge official website①. These results are shown in Table 1. It is shown that although our proposed object detection method in CGTracker gets lower AP than that using SDP [30], it can better detect the existing objects with a higher recall rate.

Table 1. Evaluation Results on the MOT17 Test Set Using Public Detection and Our Private Detection Methods

Detector	AP↑	TP↑	FP↓	FN↓	Rcll↑	Prcn↑
DPM [29]	0.61	78 077	42 308	36 577	68.1	64.8
Faster R-CNN [6]	0.72	88 601	10 081	25 963	77.3	89.8
SDP [30]	**0.81**	95 699	**7 599**	18 865	83.5	**92.6**
Ours	0.75	**105 694**	12 901	**8 813**	**92.3**	89.1

Data Association. Table 2 shows the comparison between our proposed CGTracker and DAN [14] in terms of the MOT performance by using the same object detector. In order to obtain the comparable results, we choose VGG16 [37] as the feature extraction module for the two methods. It can be seen that our method obtains higher MOTA, MT and ML scores than DAN [14], which indicates that CGTracker has a higher tracking accuracy and better tracking stability than DAN [14].

Feature Extraction Layer. We believe that the fusion of different layers of features can make objects contain multi-scale information. As shown in Table 3, when we compare using multi-scale feature fusion with using deep semantic features as object feature representation, we find that the multi-scale features we selected are far superior compared with tracking by only using high-level features in terms of all evaluation metrics.

Object Graph Based Multi-Object Association. As aforementioned, we propose to associate the pedestrian targets between two frames through the appearance feature information, displacement information and relative

Table 2. Tracking Performance of Different Detectors on the MOT17 Test Set

Method	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDSw ↓
SDP [30]+DAN [14]	55.1	76.1	**52.9**	20.8	31.7	27 792	218 973	**6 915**
SDP [30]+Objectgraph	**56.8**	**76.7**	51.4	**23.9**	**29.7**	**22 773**	**213 459**	7 419

①https://motchallenge.net/, Apr. 2022.

Table 3. Comparisons of Tracking Results Using Different Feature Selection Methods on the MOT17 Validation Set

Feature Selection Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN ↓	IDSw ↓
Multi-scale feature fusion	**61.5**	**76**	**64.2**	58	**7**	**2 420**	**1 880**	**178**
Only deep semantic features	56.5	76	53.3	**59**	8	2 444	1 888	727

position information. In order to explore the influence of different information on the overall tracking results, we gradually add different association information on the appearance feature association to prove the effectiveness of the proposed object graph based data association for MOT. The experimental results on the MOT17 test set are shown in Table 4.

1) *Only Appearance Information.* This is the simplest implementation of CGTracker. When we only use appearance feature information, our tracker will confuse those pedestrians with similar appearances, thus leading to an increase of IDSw.

2) *Appearance Information and Displacement Information.* Compared with using the appearance feature only, when displacement information is included in the data association module, although the number of FP slightly increases, the position association effectively reduces the number of IDSw. Therefore, CGtracker achieves certain improvement in terms of MOTA.

3) *All Information.* As we can see in Table 4, when all the association strategies are included for data association, CGtracker achieves the best performance in terms of major MOT metrics. In special, CGTracker significantly reduces the number of IDSw, improves the stability of tracking in terms of MT and reduces the number of missing objects in tracking indicated by FN.

In order to further demonstrate the effectiveness of the proposed object graph association strategy, we visualize the tracking results of the proposed CGTracker and DAN [14] on three selected consecutive video sequences in the test set of MOT17 in Fig. 6. As we can see from all the example video sequences, when we follow DAN to track objects by using the appearance features association only, the misalignment between objects in consecutive frames occurs. But because CGTracker contains multiple association constraints, especially with the relative relationship temporal consistency, the tracking results of CGTracker obtain consistent ID labels for all pedestrians across frames.

4.5 Benchmark Evaluation

Since the test sequence does not contain annotations, we submit the results of CGTracker to the official website of MOT Challenge② to obtain the final evaluation results. Table 5 and Table 6 give the comparison results of methods exposed by the MOT17 and MOT20 challenges and our CGTracker. All the compared methods are online MOT methods, and on the leader board of both MOT17 and MOT20 challenges. In Table 5 and Table 6 we can see the followings.

1) For the evaluation results on MOT17, all the compared methods are joint multi-object detection and tracking implementations. In particular, compared with the original DAN [14] method using an extra object feature extraction network and a data association network based on object features, CGTrakcer comprehensively considers the object feature correlation and the relation structure consistency over time, leading to significant improvement over all the MOT evaluation metrics and inference speed in terms of Hz in Table 5. Moreover, compared with the other end-to-end joint detection and tracking methods, such as CTracker [18], CenterTrack [19], Tube_TK [38] and FairMOT [20], CGTracker performs much better than CTracker, CenterTrack and Tube_TK for most of the evaluation metrics, and obtains comparable results with the state-of-the-art MOT method, e.g., FairMOT [20]. Moreover, CGTracker achieves the highest MOTP, which indicates that CGTracker achieves the best precision of object position prediction in tracking.

Table 4. Multi-Constraint Relationship Ablation Experiments on the MOT17 Test Set

Ai	Di	Rpi	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN ↓	IDSw ↓
✓			65.2	77.5	36.4	19.6	**39 990**	151 959	4 176
✓	✓		**65.3**	77.5	36.4	**19.5**	40 119	151 689	4 128
✓	✓	✓	**65.3**	**77.5**	**36.6**	19.7	40 146	**151 626**	**3 885**

Note: Ai, Di, and Rpi denote appearance information, displacement information, and relative position information, respectively.

②https://motchallenge.net/, Apr. 2022.

Fig.6. Tracking visualization results comparison between CGTracker and DAN [14]. Example frames are extracted from three video segments of MOT17: (a) MOT17-01, (b) MOT17-06, and (c) MOT17-08. The first row of each video segment indicates the tracking results of DAN [14], where the data association is only based on object appearance features. And the second row of each video segment is the tracking results of our proposed CGTracker. The predicted objects and trajectory IDs are identified by different colors of bounding boxes and lines. And the red circle in each image highlights the position of a particular object that may have misalignment in DAN tracking results.

Table 5. Comparison of MOT Methods on the MOT17 Test Set

Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN ↓	IDSw ↓	Hz ↑
DAN [14]	52.4	49.5	76.9	504	723	25 423	234 592	8 431	6.3
CTracker [18]	66.6	57.4	78.2	759	570	22 284	160 491	5 529	**34.4**
CenterTrack [19]	67.8	64.7	78.4	816	579	**18 498**	160 332	**3 039**	22.0
Tube_TK [38]	63.0	58.6	78.3	735	468	27 060	177 483	4 137	3.0
FairMOT [20]	**73.7**	**72.3**	-	**1 017**	**408**	27 507	**117 477**	3 303	15.0
Ours (CGTracker)	69.3	62.8	**80.8**	909	465	22 434	145 017	5 682	9.0

Table 6. Comparison of MOT Methods on the MOT20 Test Set

Method	MOTA↑	IDF1↑	HOTA↑	MT↑	ML↓	FP↓	FN ↓	IDSw ↓
MLT [39]	48.9	54.6	43.2	384	274	**45 660**	216 803	**2 187**
RekTCL [40]	65.2	**70.1**	**55.3**	761	131	61 209	114 709	4 139
FairMOT [20]	61.8	67.3	54.6	**855**	**94**	103 440	**88 901**	5 243
Ours (CGTracker)	**65.3**	59.7	49.7	727	154	50 455	121 803	7 190

2) For the evaluation results on the more challenged MOT20 benchmark, CGTracker achieves the best MOTA score over all recent MOT methods. While FairMOT [20] is the best method in MOT17 challenge on both the tracking accuracy and efficiency, it is worse than CGTracker on MOT20 test set, which demonstrates that CGTracker is very effective for tracking in crowded MOT scenarios, and is a highly efficient implementation for real-time MOT applications.

5 Conclusions

In this paper, we introduced a graph-based one-stage multi-pedestrian-object detection and tracking method, referred to as Center Graph Network (CGTracker). With extensive experiments, we showed that the center point based object detection and the straight feature extraction strategy in CGTracker facilitate highly efficient one-stage multi-pedestrian detection and tracking. In addition, the object graph based data association module casts the online MOT task into a graph matching process and further improves the overall detection and tracking accuracy. Experimental results on the challenging MOT datasets MOT17 and MOT20 showed that CGTracker achieves the highest tracking accuracy of 69.3% and 65.3%, respectively, and is able to reach 9 FPS in terms of inference speed. In summary, CGTracker is an end-to-end framework that jointly learns the multi-pedestrian-object detection and tracking, which is highly efficient and can be applied in real-time MOT applications.

References

[1] Kim C, Li F, Rehg J M. Multi-object tracking with neural gating using bilinear LSTM. In *Proc. the 15th European Conference on Computer Vision*, October 2018, pp.208-224. DOI: 10.1007/978-3-030-01237-3_13.

[2] Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In *Proc. the 2016 IEEE International Conference on Image Processing*, September 2016, pp.3464-3468. DOI: 10.1109/ICIP.2016.7533003.

[3] Tang S, Andriluka M, Andres B, Schiele B. Multiple people tracking by lifted multicut and person re-identification. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp.3701-3710. DOI: 10.1109/CVPR.2017.394.

[4] Possegger H, Mauthner T, Roth P M, Bischof H. Occlusion geodesics for online multi-object tracking. In *Proc. the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp.1306-1313. DOI: 10.1109/CVPR.2014.170.

[5] He A, Luo C, Tian X, Zeng W. A twofold Siamese network for real-time object tracking. In *Proc. the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp.4834-4843. DOI: 10.1109/CVPR.2018.00508.

[6] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2016, 39: 1137-1149. DOI: 10.1109/TPAMI.2016.2577031.

[7] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp.6517-6525. DOI: 10.1109/CVPR.2017.690.

[8] Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv:1804.02767, 2018. https://arxiv.org/abs/1804.02767, Jan. 2022.

[9] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934, 2020. https://arxiv.org/abs/2004.10934, April 2022.

[10] Rosebrock A. Intersection over Union (IoU) for object detection. https://pyimagesearch.com/2016/11/07/intersectionover-union-iou-for-object-detection/, July 2021.

[11] Feng X, Xue Y, Wang Y. An object based graph representation for video comparison. In *Proc. the 2017 IEEE International Conference on Image Processing*, September 2017, pp.2548-2552. DOI: 10.1109/ICIP.2017.8296742.

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

[13] Yu F, Li W, Li Q, Liu Y, Shi X, Yan J. POI: Multiple object tracking with high performance detection and appearance feature. In *Proc. the 14th European Conference on Computer Vision Workshops*, October 2016, pp.36-42. DOI: 10.1007/978-3-319-48881-3_3.

[14] Sun S, Akhtar N, Song H, Mian A, Shah M. Deep affinity network for multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, 43(1): 104-119. DOI: 10.1109/TPAMI.2019.2929520.

[15] Wang Z, Zheng L, Liu Y, Li Y, Wang S. Towards real-time multi-object tracking. In *Proc. the 16th European Conference on Computer Vision*, August 2020, pp.107-122. DOI: 10.1007/978-3-030-58621-8_7.

[16] Lu Z, Rathod V, Votel R, Huang J. RetinaTrack: Online single stage joint detection and tracking. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020, pp.14656-14666. DOI: 10.1109/CVPR42600.2020.01468.

[17] Zhu J, Yang H, Liu N, Kim M, Zhang W, Yang M H. Online multi-object tracking with dual matching attention networks. In *Proc. the 15th European Conference on Computer Vision*, October 2018, pp.379-396. DOI: 10.1007/978-3-030-01228-1_23.

[18] Peng J, Wang C, Wan F, Wu Y, Wang Y, Tai Y, Fu Y. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *Proc. the 16th European Conference on Computer Vision*, August 2020, pp.145-161. DOI: 10.1007/978-3-030-58548-8_9.

[19] Zhou X, Koltun V, Krähenbühl P. Tracking objects as points. In *Proc. the 16th European Conference on Computer Vision*, August 2020, pp.474-490. DOI: 10.1007/978-3-030-58548-8_28.

[20] Zhang Y, Wang C, Wang X, Zeng W, Liu W. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 2021, 129(11): 3069-3087. DOI: 10.1007/s11263-021-01513-4.

[21] Zhou X, Wang D, Krähenbühl P. Objects as points. arXiv:1904.07850, 2019. https://arxiv.org/abs/1904.07850, April 2022.

[22] Yu F, Wang D, Shelhamer E, Darrell T. Deep layer aggregation. In *Proc. the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp.2403-2412. DOI: 10.1109/CVPR.2018.00255.

[23] Wang X, Liu Z. Salient object detection by optimizing robust background detection. In *Proc. the 18th IEEE International Conference on Communication Technology*, October 2018, pp.1164-1168. DOI: 10.1109/ICCT.2018.8600184.

[24] Law H, Deng J. CornerNet: Detecting objects as paired keypoints. In *Proc. the 15th European Conference on Computer Vision*, October 2018, pp.765-781. DOI: 10.1007/978-3-030-01264-9_45.

[25] Tian Z, Shen C, Chen H, He T. FCOS: Fully convolutional one-stage object detection. In *Proc. the 2019 IEEE/CVF International Conference on Computer Vision*, Oct. 27-Nov. 2, 2019, pp.9626-9635. DOI: 10.1109/ICCV.2019.00972.

[26] Neubeck A, van Gool L. Efficient non-maximum suppression. In *Proc. the 18th International Conference on Pattern Recognition*, August 2006, pp.850-855. DOI: 10.1109/ICPR.2006.479.

[27] Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K. MOT16: A benchmark for multi-object tracking. arXiv:1603.00831, 2016. https://arxiv.org/abs/1603.00831, Jan. 2022.

[28] Dendorfer P, Rezatofighi H, Milan A *et al.* MOT20: A benchmark for multi object tracking in crowded scenes. arXiv:2003.09003, 2020. https://arxiv.org/abs/ 2003.09003, March 2022.

[29] Felzenszwalb P F, Girshick R B, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009, 32(9): 1627-1645. DOI: 10.1109/TPAMI.2009.167.

[30] Yang F, Choi W, Lin Y. Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.2129-2137. DOI: 10.1109/CVPR.2016.234.

[31] Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008: Article No. 1. DOI: 10.1155/2008/246309.

[32] Luiten J, Ošep A, Dendorfer P, Torr P, Geiger A, Leal-Taixé L, Leibe B. HOTA: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 2021, 129(2): 548-578. DOI: 10.1007/s11263-020-01375-2.

[33] Paszke A, Gross S, Chintala S *et al.* Automatic differentiation in PyTorch. In *Proc. the 31st Conference on Neural Information Processing Systems Workshop*, Dec. 2017.

[34] Shao S, Zhao Z, Li B, Xiao T, Yu G, Zhang X, Sun J. CrowdHuman: A benchmark for detecting human in a crowd. arXiv:1805.00123, 2018. https://arxiv.org/abs/1805.00123, Jan. 2022.

[35] Zhang S, Xie Y, Wan J, Xia H, Li S Z, Guo G. WiderPerson: A diverse dataset for dense pedestrian detection in the wild. *IEEE Transactions on Multimedia*, 2019, 22(2): 380-393. DOI: 10.1109/TMM.2019.2929005.

[36] Zhang S, Benenson R, Schiele B. CityPersons: A diverse dataset for pedestrian detection. In *Proc. the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, June 2017, pp.4457-4465. DOI: 10.1109/CVPR.2017.474.

[37] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014. https://arxiv.org/abs/1409.1556, April 2022.

[38] Pang B, Li Y, Zhang Y, Li M, Lu C. TubeTK: Adopting tubes to track multi-object in a one-step training model. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020, pp.6307-6317. DOI: 10.1109/CVPR42600.2020.00634.

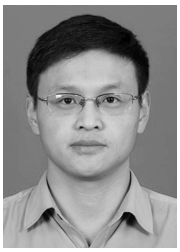
[39] Zhang Y, Sheng H, Wu Y, Wang S, Ke W, Xiong Z. Multiplex labeling graph for near-online tracking in crowded scenes. *IEEE Internet of Things Journal*, 2021, 7(9): 7892-7902. DOI: 10.1109/JIOT.2020.2996609.

[40] Li W, Xiong Y, Yang S, Xu M, Wang Y, Xia W. Semi-TCL: Semi-supervised track contrastive representation learning. arXiv:2107.02396, 2021. https://arxiv.org/abs/2107.02396, Jan. 2022.

Xin Feng received her B.S. degree in computer science and technology from Chongqing University, Chongqing, in 2004. She got her Ph.D. degree in computer applications from Chongqing University, Chongqing, in 2011. She is currently an associate professor of Chongqing University of Technology, Chongqing. She studied at New York University, New York, as a postdoctor from 2014 to 2016. Her research falls in the area of computer vision, image and video processing.

Hao-Ming Wu received his B.S. degree in software engineering from Chongqing University of Technology, Chongqing, in 2019. He is currently studying for his Master's degree at Chongqing University of Technology, Chongqing. His research falls in the areas of machine learning and computer vision.

Yi-Hao Yin received his B.S. degree in computer science and technology from Chongqing University of Posts and Telecommunications, Chongqing, in 2018, and his M.S. degree in computer science from Chongqing University of Technology, Chongqing, in 2021. His research falls in the areas of machine learning and computer vision.



Li-Bin Lan received his B.S. degree in computer science and technology from Chongqing University of Posts and Telecommunications, Chongqing, in 2008, and his M.S. and Ph.D. degrees in computer science from Chongqing University, Chongqing, in 2011 and 2021 respectively. He is currently a lecturer with the Chongqing University of Technology, Chongqing, and a postdoctoral researcher with the College of Computer Science, Chongqing University, Chongqing. His research falls in the areas of machine learning, computer vision, and machine vision.