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

Xin Feng[1,*], Haoming Wu[2], Yihao Yin[3], Yongbo Li[4], Libin Lan[5]
College of Computer Science and Engineering, Chongqing University of Technology
Chongqing 400054, China
Email: xfeng@cqut.edu.cn, 1453119471@qq.com.cn, bleach@2019.cqut.edu.cn,
971005586@qq.com, lanlbn@cqut.edu.cn

Abstract

Most current online multi-object tracking (MOT) methods usually 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 consider combining 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 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 object detection and tracking method, named CGTracker. In particular, our CGTracker detects the pedestrian target as the center point of object, and directly extracts the object features from the feature representation of the object center, which is used to predict the axis-aligned bounding box. Meanwhile, the detected pedestrian objects 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. It is evaluated on the popular MOT17 challenge, and achieves 65.3% MOTA at 9 FPS. 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 challenge at the time of submission of this work. The code will be made publicly available.

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 there is less information available, it is extremely challenging for online tracking methods to satisfy both high tracking accuracy and low time delay.

Currently, tracking-by-detection [1, 36, 19, 7, 14, 4, 29] has become the main framework in the field of MOT. In this framework, the detector is used to locate the object frame by frame, and the data association method [29, 24, 10] is then used to associate the same target 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: i) 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 a anchor-based object detection method [27, 25, 26, 5], 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 network structure not only makes the model more complex, but also reduces the tracking efficiency. ii) Most MOT methods focus on associating objects based on appearance features of the detected objects through Intersection over Union (IOU). This data association, however, does not consider the spatial relationships between different objects in the same frame and same objects in the consecutive frames.

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 the main problem of MOT. In order to realize high efficient and accurate online multi-pedestrian object tracking, we design a novel one-stage multi-object detection and tracking method,

named Center Graph Tracker (CGTracker) by jointly learning the multi-object detection and tracking prediction in an unified framework. 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 spatial relationship between pedestrian objects, an object graph is then constructed from the extracted pedestrian object features and spatial relationships between objects in a frame and across frames to learn the object association under the online MOT objectives.

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 [27], YOLO [25, 26, 5], etc., to locate object as a regular bounding box. These detectors, on one hand, need to generate lots of region proposals or anchors, which is less efficient or leads to inferior detection performance. On the other hand, the detected bounding box contains more information than the object location only, e.g. some background pixels. In fact, object detection for MOT does not require to detect the entire object body, especially for pedestrian object, but some key point that is able to provide the pedestrian location information is sufficient.

Moreover, as demonstrated in anchor-based object detection methods, the high-level features that are extracted from the backbone network, e.g. darknet-53 in YOLOV3 [26], will provide semantic object information for object classification and localization. Hence, the feature points that are corresponding to the detected anchor and the resulted object are effective object feature representation. Following this idea, we propose to extract the feature of the detected object directly from the multi-scale features of the backbone network according to the detected object center point. As a result, the pedestrian object detection module in our CGTracker would provide both the object location and corresponding feature representation that are required by the subsequent multi-object association process. This facilitates the more 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. While we believe that besides appearance feature, the relative relationship between pedestrian objects in the same frame, and temporal correlation between same identity in consecutive frames are also importance tracking cues. Hence, inspired by the object graph representation for videos [9], 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 pedestrian objects in a frame as the edge description of the object graph. We then consider the association process as 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, which combines both multi-object detection and data association modules in an unified framework.

2. We detect the pedestrian object as a center point, and directly extract the target features based on the center point from multi-scale feature representations of the backbone network according to the center point coordinates of the object. Our experiment verifies the effectiveness of the extracted features for the MOT task well.

3. In order to ensure the accuracy of the CGTracker, we build an object graph based on the detected pedestrian objects in a video frame, and apply the matching between two object graphs of two consecutive frames from three aspects: (i) the appearance association for nodes between the two graphs; (ii) the relative relationship similarity for edges between the two graphs ;(iii) the displacement constrains between the nodes (objects) of the two graphs.

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 our tracking method in detail. Section 4 gives experimental details and results, and evaluates the effectiveness of our tracking components on widely used benchmark MOT17 by some ablation experiments. Finally, Section 5 summarizes 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 [33] 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 Kalman filter to realize the whole MOT process. Yu *et al.* [38] then show 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

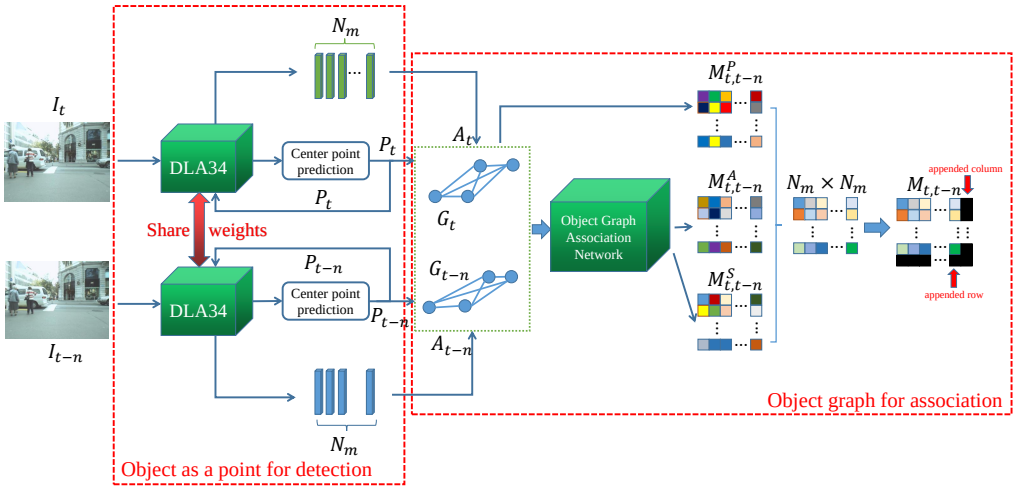

Figure 1. The architecture of the proposed CGTracker, consisting of: (i) object as a point for detection; (ii) object graph based data assoiation

offline tracking. These tracking-by-detection based methods, however, have some weakness in: (i) The overall tracking performance is highly dependent on the detection results. (ii) 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.* [28] propose 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.* [32] propose a joint detection detection and embedding MOT paradigm by incorporating the embedding learning into the object detector for fast MOT system. Similarly, Lu *et al.* [17] propose single-stage RetinaTrack by improving the single-stage RetinaNet, which combines target detection with feature extraction. Zhu *et al.* [42] combine Bi-LSTM network with attention mechanism to achieve an end-to-end matching attention network. Although these methods attempt to jointly learn some of the modules of MOT in the end-to-end manner, they have not incorporated the entire detection and association learning in an unified framework for more efficient and accurate MOT system.

Future prediction MOT method. Very recently, the MOT system is proposed to jointly learn the object detector and moving offset in an unified framework [2, 23, 40, 6]. In these methods, the predicted bounding boxes are further used as region proposals for both detection and tracking in the future frames. The simplicity of this approach is very attractive, but the accuracy of its tracking results needs to be further improved.

3. Methodology

Given a sequence of video frames, the goal of MOT task is to associate the same identity in different frames and assign it a unique trajectory ID. Existing MOT methods often 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 investigate the characteristics of object category for detection and tracking, especially for the commonly appeared pedestrian objects. By exploring the advantage of the center point based object detection method, and the relationship of the detected pedestrian objects 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 association into a single framework, and simultaneously completes object detection, feature extraction and object association in the network inference. In the following subsections, we introduce the pipeline of our method, and describe the proposed multi-object detection and association modules, and our adopted loss functions.

3.1. Architecture of proposed method

CGTracker aims to realize high efficient deep learning based pedestrian multi-pedestrian-object tracking to facilitate online tracking in real-time tracking applications, such as auto driving, etc. The method takes two consecutive frames with n-frame interval as input, and the multi-object detection and tracking are mainly implemented in the center

point based object detection and object graph based association module. The entire framework is shown in Figure 1.

First, in order to render a more effective pedestrian object detection for multi-object tracking, we propose to detect the object as the center point by following the idea of CenterNet [41]. Because the multi-object tracking eventually relies on object feature association, 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 [39], 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, which 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-point of the detected pedestrian objects for each frame, so as to effectively combine the relative position constraint between pedestrians in a frame, and the displacement constraint between objects across different frames, in addition to the appearance feature association. As shown in Figure 1, two object graphs G_t and G_{t-n} are constructed for frame t and $t-n$ respectively. The nodes in each graph encode the detected objects described by its appearance feature A_t (or A_{t-n}) and position feature P_t (or P_{t-n}), and the edges of a graph encode the spatial relationship between different pedestrian objects in the frame. The two object graphs then facilitate an object graph association network to realize the multi-constraint data association between frame t and $t-n$ through matching of nodes appearance similarity $M^A_{t,t-n}$, edge (or structure) similarity $M^S_{t,t-n}$ and nodes displacement similarity $M^P_{t,t-n}$. Finally, the three matching matrices are then integrated to generate the object association result $M^P_{t,t-n}$ and the final result of object tracking is obtained by the Hungarian algorithm.

3.2. Object as a point for detection

As aforementioned, pedestrian object detection for MOT does not need to detect object as a regular bounding box, but only 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 the pedestrian object 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 consider center

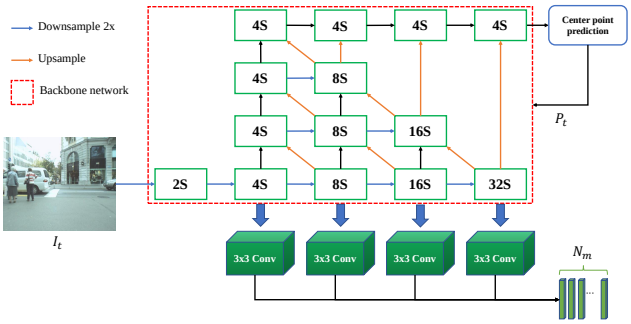

Figure 2. Illustration of center point based object detection and multi-scale feature extraction from DLA [39]. The red frame indicates the structure of the backbone network DLA [39], where each green box represents the process of extracting high-level features from the initial feature layer through multiple convolutional layers. The solid green cubes indicate the extracted feature tensors.

point and its surroundings as the most salient representation of an object [31]. On the other hand, the recent anchor-free based deep learning methods [41, 15, 30] have greatly advanced the object detection field. These methods learn to detect the object as key points, and have shown to be more efficient than the two-stage anchor based object detection methods and more accurate than the anchor based one-stage object detection methods. Inspired by these techniques, we propose to detect the center point of the pedestrian object for object detection of MOT. By following CenterNet [41], our center-point based object detection does not require preset anchors and the undifferentiable NMS [20] 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 object association, we propose to extract feature representation from the backbone network of object detection according to the center point coordinate of object. Specifically, we use the Deep Layer Aggregation (DLA) [39] network as the backbone for pedestrian object feature extraction. As shown in Figure 1, DLA is a network building in 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, respectively. And inspired by [10], we intentionally make the two-stream DLA network with shared weights. After inference on the center based object detection network, the center position of the detected pedestrian objects can be located. We then trace back to the backbone network to search for the best

RoI (region of interest) feature representation according to the center location of the object P_t. It is shown that high-level semantic features is good representation for object recognition, while data association in the MOT task requires feature representation that can distinguish different objects, but not recognize the object categories only. Hence, we propose to extract the multi-scale features from different down-sample layers of DLA network, as is shown in Figure 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 pedestrian objects 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 position, in other word, the spatial relationship between objects will 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 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 $A_t^i \in \mathbb{R}^{520}$ and position information $P_i^t \in \mathbb{R}^2$ of the object i. In addition, the edge $E_t^{i,j} \in \mathbb{R}^2$ of object graph G_t is described by the difference between center coordinates of the detected objects i and j. As illustrated in Figure 3, two object graphs G_t and G_{t-n} are derived from frame I_t and I_{t-n} respectively, where $G_t = (O_t, E_t)$, with $O_t = \{(A_t^i, P_t^i)\}_{i=1}^{N_m}$ and $E_t = \{(E_t^{i,j})\}_{i=1\,j=1}^{N_m\,N_m}$, 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 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 matching of nodes descriptors: the appearance feature A^i of object i and the position displacement P^i of object i in consecutive frames.

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

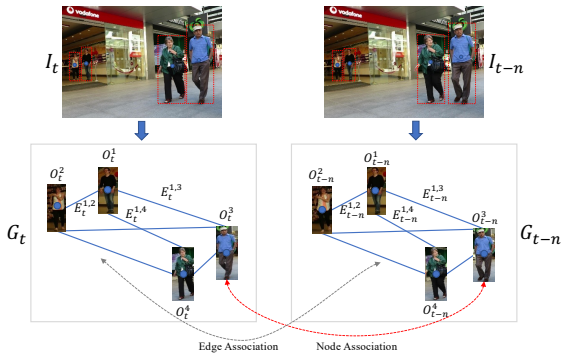

Figure 3. Illustration of two object graphs constructed from the t^{th} frame and $(t-n)^{th}$ frame. The solid blue lines in each object graph are the edges between adjacent objects, the red dash lines denote the object node correspondences, and the gray dash lines indicate the edge correspondences.

Figure 4. The node association and edge association between two object graphs of I_t and I_{t-n}. The red solid line represents the object node that is successfully matched by the node association strategy; the orange solid line represents the structural similarity information of the object node 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 backbone network according to the center position of the object, as shown in Figure 2. The selection of multi-scale CNN layers will be discussed in Section 4.4. The appearance features of all the objects in frame I_t are aggregated, and through the one-to-one correspondence of appearance features of objects in the two object graphs, an appearance feature matrix $A_{t-n,t}^N \in \mathbb{R}^{1040 \times N_m \times N_m}$ is obtained. This matrix is then fed into a node association network, which is composed of 5 3×3 convolution layers with $[512, 256, 128, 64, 1]$ for channel number of each layer, to learn the appearance similarity matrix $M_{t-n,t}^A \in \mathbb{R}^{N_m \times N_m}$ under the MOT objective.

Position association of object nodes: As is known that the movement of pedestrian objects is temporally coherent, which means the position of an object would have few variance in short time period. We then consider measuring the displacement similarity between objects in different object

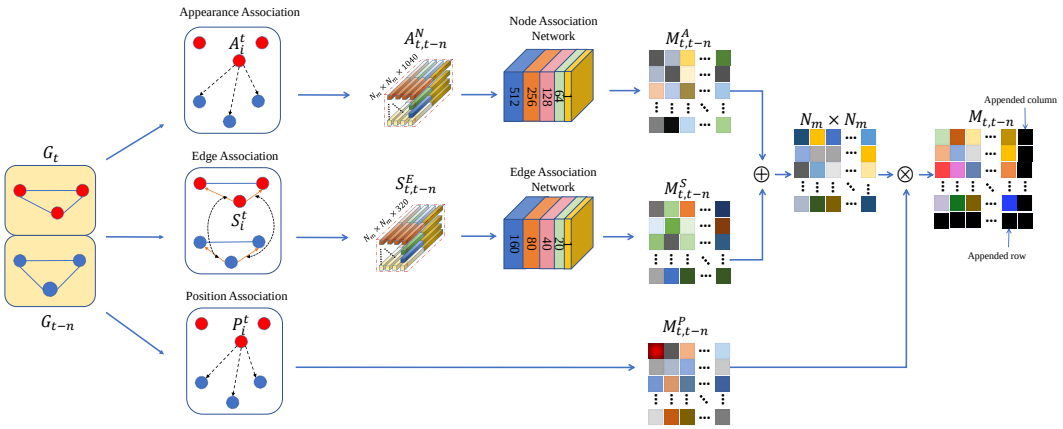

Figure 5. Object graph association network structure diagram. After constructing the object in the t^{th} frame image I_t and the $(t-n)^{th}$ frame image I_{t-n} into the object graphs G_t and G_{t-n}, we use information such as appearance, displacement, and relative position, and use different association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

graphs of consecutive frames. In special, the position distance between objects of the same identity in consecutive frames would be smaller than objects of different identities. Hence, we calculate the position distance between all nodes in two consecutive object graphs and form the position similarity matrix $M^P_{t,t-n}$, where each item is computed as:

$$M^P_{i,j} = \frac{e^{-d_{i,j}/Dia(I)} - e^{-1}}{1 - e^{-1}}. \tag{1}$$

where $d_{i,j}$ is the Euclidean distance between the center position of 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 is 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 Equation (1).

3.3.2 Edge association

In multi-object tracking scenario, a moving pedestrian often moves along certain trajectory, while the relative position between this pedestrian and other objects will also be maintained in a moment. For example, object a is to the right of object b at time $t-n$, a will most likely be at the right of b at time t as well, when n is a short interval. Therefore, besides tracking over individual moving object, we propose an additional tracking objective by taking relationship consistency over time into account.

As shown in Figure 3, based on the edge descriptor that calculates the direction vector between objects in the same frame, CGTracker performs the edge-to-edge association between consecutive object graphs to realize the relationship correspondence of pedestrian objects. In addition, the learning process of edge association is illustrated in Figure 5. $S^t_i \in \mathbb{R}^{320}$ 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 $S^E_{t-n,t} \in \mathbb{R}^{320*N_m*N_m}$. Similar with node association, we construct an edge association network to learn the relation structure similarity matrix $M^S_{t-n,t} \in \mathbb{R}^{N_m*N_m}$, which also consists of 5 3×3 convolutional layers with $[160, 80, 40, 20, 1]$ as 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:

$$M_{t-n,t} = (M^A_{t-n,t} + M^S_{t-n,t}) \odot M^P_{t-n,t}. \tag{2}$$

where $\odot$ represents the dot product between two matrix. In order to solve the objects entering or leaving problem in consecutive frames, we add an extra row and column to $M_{t-n,t}$, and obtain the final object association matrix $M_{t-n,t} \in \mathbb{R}^{(N_m+1)*(N_m+1)}$ followed by row and column regularization for MOT optimization, as shown in Figure 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 [41] 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 object and localize the object center, we use Gaussian kernel function: $H_{xyc} = exp(-\frac{(x-\lfloor \frac{x_k}{r} \rfloor)^2+(y-\lfloor \frac{y_k}{r} \rfloor)^2}{2\sigma_k^2})$, to distribute the centers of all GT targets on the heatmap, $H \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times C}$ 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 GT object k, and σ_k is an object size-adaptive standard deviation [15].

With the Gaussian based center point representation, we optimize the loss between predicted and GT center category by following the focal loss in [16] to derive L_{cls}, and L1 loss for the object size L_{size} and offset L_{off} regression. In summary, the overall object detection learning objective is:

$$L_{det} = \lambda_1 L_{cls} + \lambda_2 L_{size} + \lambda_3 L_{off}. \tag{3}$$

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

Object association loss. For object association, we mainly follow the loss function designed in DAN [28]. 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 I_t. Let's denote $M_1 \in \mathbb{R}^{N_m*(N_m+1)}$ as the first m rows of data of the object incidence matrix $M_{t-n,t} \in \mathbb{R}^{(N_m+1)*(N_m+1)}$, with N_m+1 represents 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 $G_t \in \mathbb{R}^{N_m*(N_m+1)}$ constructed from the tracking ground truth of object in I_{t-n} to object I_t as:

$$L_1 = \frac{\sum_{\text{coeff}} (G_t \odot (-log(S(M_1))))}{\sum_{\text{coeff}} (G_t)}. \tag{4}$$

where S is the softmax function, coeff represents 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 result, we further consider the backward object association from frame I_t to I_{t-n}. The ground truth matrix $G_{t-n} \in \mathbb{R}^{(N_m+1)*N_m}$ is constructed from the one-to-one correspondence of objects in I_t to frame I_{t-n}, with N_m+1 here represents 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:

$$L_2 = \frac{\sum_{\text{coeff}} (G_{t-n} \odot (-log(S(M_2))))}{\sum_{\text{coeff}} (G_{t-n})}. \tag{5}$$

where $M_2 \in \mathbb{R}^{(N_m+1)*N_m}$ represents the first m columns of data of the object incidence matrix $M_{t-n,t} \in \mathbb{R}^{(N_m+1)*(N_m+1)}$.

(3) Consistency judgment loss L_3. Basically, the forward and backward association between objects in frame I and I_n would be consistent, hence, we formulate the bi-direction association consistency between (1) and (2) as:

$$L_3 = \left\| \widehat{S(M_1)} - \widehat{S(M_2)} \right\|_1. \tag{6}$$

(4) Joint judgment loss L_4. Similar with [**?**], we perform the non-maximum suppression for both forward and backward object association results, which is formulated as:

$$L_4 = \frac{\sum_{\text{coeff}} (G_{t-n,t} \odot (-log(max(\widehat{S(M_1)}, \widehat{S(M_2)}))))}{\sum_{\text{coeff}} (G_{t-n,t})}. \tag{7}$$

By combining the four loss functions, we have the overall object association loss as:

$$L_{ass} = \frac{L_1 + L_2 + L_3 + L_4}{4}. \tag{8}$$

Finally, the total loss of CGTracker can be summarized:

$$L_{all} = \eta_1 L_{det} + \eta_2 L_{ass}. \tag{9}$$

According to our experimental results, the hyper-parameters of η_1 and *etc.*$_2$ 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) benchmark: MOT17 [18]. MOT17 is one of the latest released online challenges in pedestrian tracking, which contains 7 training sequences and 7 test sequences. These videos mainly come from a stationary or moving camera in an unconstrained environment. Pedestrians in the scene have frequent access, crowding and occlusion, and the frame rate is 25-30 FPS. The video sequences used for training model all provide accurate annotations, and the detection results from three different detectors, namely DPM [8], SDP [35], and Faster R-CNN [27]. 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 set and validation set, each containing roughly half of the whole training data, where the first half frames are used for training, and the second for validation. Because of limited access to the test server, we evaluate our main results on the test set, but other results on the validation set, e.g., ones from ablation experiment.

Table 1. Evaluation results on the MOT17 test set using public detection and our private detection. The symbol ↑ indicates that higher is better, ↓ means that that lower is better. The best result is highlighted in **bold**.

Detector	AP↑	TP↑	FP↓	FN↓	Rcll↑	Prcnn↑
DPM [8]	0.61	78077	42308	36577	68.1	64.8
Faster R-CNN [27]	0.72	88601	10081	25963	77.3	89.8
SDP [35]	**0.81**	95699	**7599**	18865	83.5	**92.6**
Ours	0.75	**105694**	12901	**8813**	**92.3**	89.1

Table 2. Tracking performance of different detectors on the MOT17 test set.

Method	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDSw ↓
SDP [35]+DAN [28]	55.1	76.1	**52.9**	20.8	31.7	27792	218973	**6915**
SDP [35]+Objectgraph	**56.8**	**76.7**	51.4	**23.9**	**29.7**	**22773**	**213459**	7419

4.2. Evaluation metrics

In order to quantitatively evaluate our results on the MOT17 challenge, we choose the official evaluation standard CLEAR MOT metrics [3], including the multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP), false positives (FP), false negatives (FN), identity switches (IDS) and IDF1 score. 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.

4.3. Implementation details

We implement our proposed approach using Pytorch framework [22]. Training is performed on an NVIDIA GeForce RTX 2080ti GPU with standard SGD for 35 epochs. The input resolution is resized to 544×960. Other hyper-parameter values used in our implementation include batch size `batch_size=3`, maximum number of object detection per frame $N_m = 80$, and 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 $n = 30$.

4.4. Results and analysis

In this section, we intend to explain why our proposed approach is effective from the following three aspects. First, we compare the performance of different detectors on the tracking. Second, we prove the effectiveness of the selected semantic features of the target. Finally, we compare the performance of our tracker under different constraints.

Detection results on tracking task. We compare our proposed object detection method with the three public detection results provided on MOT Challenge official website. These results are shown in Table 1. It can be seen from Table 1 that although our detector is lower than the SDP [35] in Average Precision (AP), it can better detect the existing targets with a higher recall rate, and is more beneficial for multi-object tracking tasks.

Data Association. Table 2 shows the performance comparison between our proposed CGTracker and DAN [28] using the same detector. In order to obtain the comparable results, we choose VGG16 as the feature extraction module for the two methods. It can be seen that our tracking method is superior to DAN [28] in terms of tracking accuracy and continuous tracking ability.

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 multi-scale feature fusion with only deep semantic features, we find that the multi-scale features we selected are far superior compare to the tracking with only high-level features in terms of all evaluation metrics.

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

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

(2) Appearance information and displacement information. When the displacement information of the object is added, although the FP has a small increase, compared with only the appearance feature information, the position association branch effectively reduces the situation of the target

Table 3. Comparisons of tracking results using different feature selection methods on the MOT17 validation set.

Layers	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN ↓	IDSw ↓
Multi-scale feature fusion	**61.5**	**76**	**64.2**	58	**7**	**2420**	**1880**	**178**
Only deep semantic features	56.5	76	53.3	**59**	8	2444	1888	727

IDSw. Therefore, our tracker has a certain improvement in MOTA.

(3) All information. As we can see, after using all association branches, our tracker achieves the best in most metrics. As shown in the table 4, CGTracker significantly reduces the number of IDSw and improves the stability of tracking. In addition, it reduces the number of missing objects.

As shown in Figure 6, we selects three consecutive video sequences in the test set to show the results of pedestrian multi-object tracking in our ablation experiment. The first line of each video sequence shows the tracking results using only the appearance features of the object, and the second line of pictures demonstrates the tracking results after using all the information. In the first row, the target in the red area is similar in shape to the distant target in the previous frame, and there is a mismatch, resulting in an abnormal trajectory. In the second line of the picture, after adding the spatial information between the targets, the wrong matching of the targets disappeared.

4.5. Benchmark evaluation

Since the test sequence does not contain annotations, we can only submit the results of our test to the official website of MOT Challenge to obtain the final evaluation result of our method. Table 5 shows some of the results from the methods disclosed by the challenge server and the results of the current most popular multi-object trackers. Though our method is an online tracking one, it performs competitively with the best offline tracking methods. From Table 5 we can see that:

(1) In the part of the tracking method using the public detector, because the offline tracking method can use the information of the entire video stream, good results can be obtained in IDSw, but our method is significantly better than them on the main metric MOTA. Compared with the online tracking method, we can see that our method is superior to other methods in most metrics. Part of the reason is that our detector performs better and faster than other methods.

(2) In the private detection part, our MOTA is only 2.5 lower than CenterTrack [40] and 1.3 lower than CTracker [23], which is the best online tracking algorithm today. These methods either only consider the displacement information of the target or only the appearance information of the target. The CGTracker comprehensively considers all the information of the target, so our tracker is obviously stronger than them in MT, ML and other metrics, which shows that our tracking algorithm has excellent performance in continuous tracking ability.

5. Conclusion

In this paper, we introduce a graph based one-stage multi-pedestrian-object detection and tracking method, referred to as center graph network (CGTracker). It first detects pedestrian object and locates the object center for two consecutive video frames respectively. And the feature of the object is extracted directly from the feature map of backbone network based on the objects center location. It then constructs an object graph for each frame to realize the cross-frame object association, where the node of the graph consists of object appearance feature and center coordinate, while the edge of the graph describes the relative position between any two objects in a frame. With the proposed object graph, we cast the online MOT task into a graph matching process by not only considering the feature association of individual objects across frames, but also the consistency of relative relationship between objects in consecutive frames. Extensive experimental results demonstrate that CGTracker achieves the most advanced tracking accuracy in the MOT17 benchmark, and is also very efficient in terms of inference speed. 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, such as auto-driving.

References

[1] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1218–1225, 2014. 1

[2] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. Tracking without bells and whistles. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 941–951, 2019. 3, 10

[3] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 01 2008. 8

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 1

[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 1, 2

[6] P. Chen, D. Dong, H. Lv, and L. Zhu. A user motion data acquisition and processing method for the design of rehabil-

Table 4. Multi-constraint relationship ablation experiments on the MOT17 test set. Ai, Di, and Rpi denotes appearance information, displacement information, and relative position information, respectively.

Ai	Di	Rpi	MOTA↑	MOTP↑	MT↑	ML↓	FP↓	FN ↓	IDSw ↓
✓			65.2	77.5	36.4	19.6	**39990**	151959	4176
✓	✓		**65.3**	77.5	36.4	**19.5**	40119	151689	4128
✓	✓	✓	**65.3**	**77.5**	**36.6**	19.7	40146	**151626**	**3885**

Table 5. Evaluation results on the MOT17 test set.

Public Detection

Process	Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN ↓	IDSw ↓	Hz ↑
Offline	MHT_bLSTM [14]	47.5	51.9	77.5	429	981	25981	268042	3124	**1.9**
	JCC [13]	51.2	54.5	75.9	493	872	25,937	247,822	1802	1.8
	FWT [11]	51.3	47.6	77	505	830	24101	247921	4279	0.2
	Lif_T [12]	**60.5**	**65.6**	**78.3**	**637**	**791**	**14966**	**206,619**	**1,189**	0.5
Online	DEEP_TAMA [37]	50.3	53.5	76.7	453	883	25479	252996	3978	1.5
	STRN [34]	50.9	**56**	75.6	446	797	25295	249365	9363	**13.8**
	Tracktor [2]	**56.3**	55.1	**78.8**	498	831	**8866**	235449	**3763**	1.5

Private Detection

Process	Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN ↓	IDSw ↓	Hz ↑
Online	DAN [28]	52.4	49.5	76.9	504	723	25423	234592	8431	6.3
	CTracker [40]	66.6	57.4	78.2	759	570	22284	160491	5529	**34.4**
	CenterTrack [23]	**67.8**	**64.7**	**78.4**	816	579	**18498**	160332	**3039**	22
	Tube_TK [21]	63	58.6	78.3	735	468	27060	177483	4137	3
	Ours(CGTracker)	65.3	60.4	77.5	**861**	**465**	40146	**151626**	3885	9

itation robot with few degrees-of-freedom. *Journal of Engineering and Science in Medical Diagnostics and Therapy*, 3(2), 2020. 3

[7] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3029–3037, 2015. 1

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 7, 8

[9] X. Feng, Y. Xue, and Y. Wang. An object based graph representation for video comparison. In *International Conference on Image Processing (ICIP'17)*, pages 2548–2552, 2017. 2

[10] A. He, C. Luo, X. Tian, and W. Zeng. A twofold siamese network for real-time object tracking. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4834–4843, 2018. 1, 4

[11] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn. Fusion of head and full-body detectors for multi-object tracking. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1509–150909, 2018. 10

[12] A. Hornakova, R. Henschel, B. Rosenhahn, and P. Swoboda. Lifted disjoint paths with application in multiple object tracking, 2020. 10

[13] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele. Motion segmentation amp; multiple object tracking by correlation co-clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1):140–153, 2020. 10

[14] C. Kim, F. Li, and J. M. Rehg. Multi-object tracking with neural gating using bilinear lstm. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 208–224, Cham, 2018. Springer International Publishing. 1, 10

[15] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 765–781, Cham, 2018. Springer International Publishing. 4, 7

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 7

[17] Z. Lu, V. Rathod, R. Votel, and J. Huang. Retinatrack: Online single stage joint detection and tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14656–14666, 2020. 3

[18] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking, 2016. 7

[19] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler. Online multi-target tracking using recurrent neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4225–4232. AAAI Press, 2017. 1

[20] A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855, 2006. 4

[21] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model.

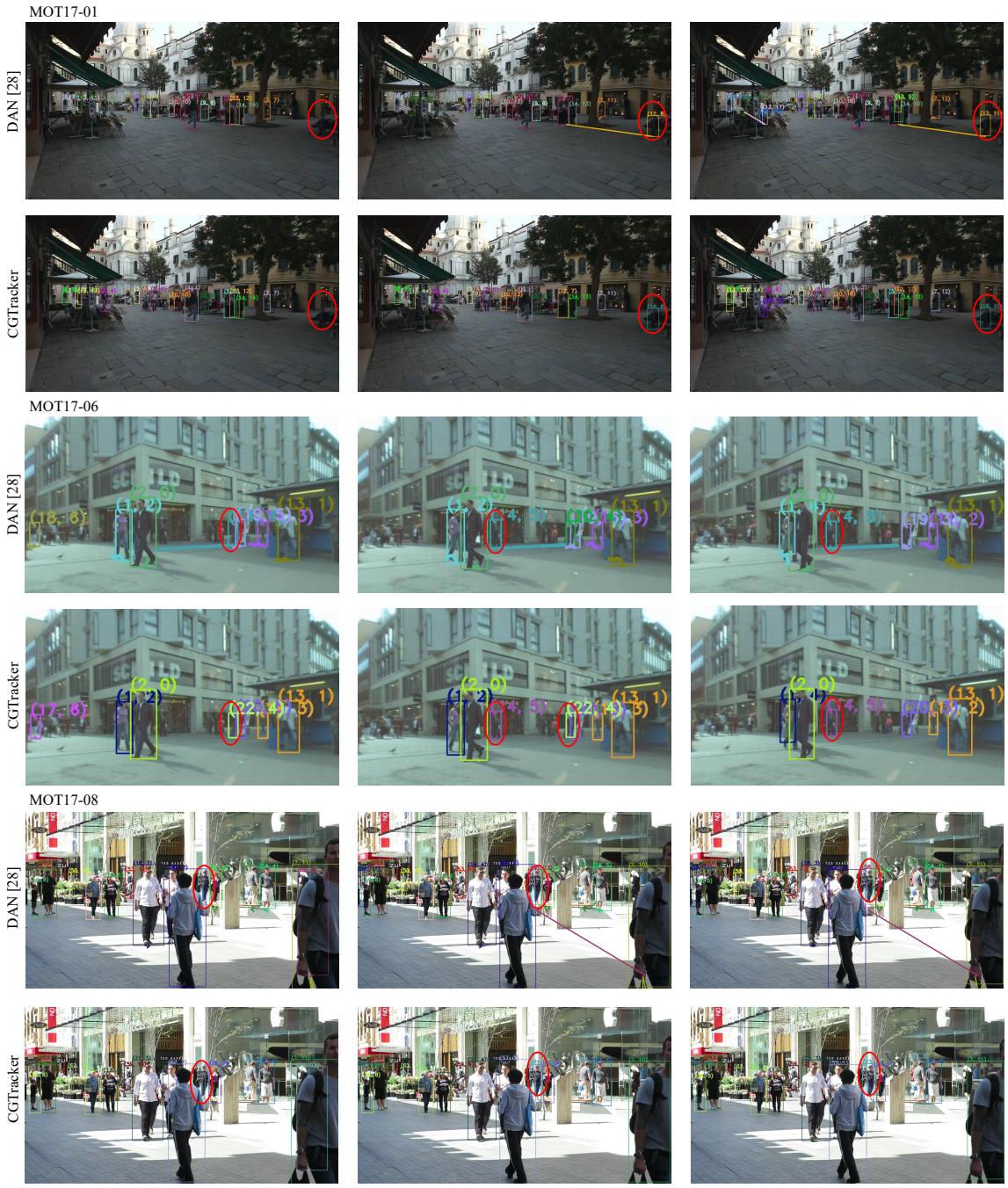

Figure 6. Comparison of CGTracker tracking results with DAN [28]. Example frames are extracted from three video segments of MOT17: MOT17-01 (the first and second rows), MOT17-06 (the third and forth rows), MOT17-08 (the last two rows). The first row for each video segment indicates the tracking results with appearance association only, which is the data association part in DAN[28]. And the second row for 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.

In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6307–6317, 2020. 10

[22] A. Paszke, am Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 1–4, Long Beach, CA, USA, December 2017. 8

[23] 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 A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 145–161, Cham, 2020. Springer International Publishing. 3, 9, 10

[24] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1306–1313, 2014. 1

[25] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1, 2

[26] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 1–14. Curran Associates, Inc., 2015. 1, 2, 7, 8

[28] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah. Deep affinity network for multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):104–119, 2021. 3, 7, 8, 10, 11

[29] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3701–3710, 2017. 1

[30] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635, 2019. 4

[31] X. Wang and Z. Liu. Salient object detection by optimizing robust background detection. In *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pages 1164–1168, 2018. 4

[32] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang. Towards real-time multi-object tracking. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 107–122, Cham, 2020. Springer International Publishing. 3

[33] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017. 2

[34] J. Xu, Y. Cao, Z. Zhang, and H. Hu. Spatial-temporal relation networks for multi-object tracking, 2019. 10

[35] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2129–2137, 2016. 7, 8

[36] J. H. Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon. Online multi-object tracking via structural constraint event aggregation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1392–1400, 2016. 1

[37] Y.-C. Yoon, D. Y. Kim, Y.-M. Song, K. Yoon, and M. Jeon. Online multiple pedestrians tracking using deep temporal appearance matching association. *Information Sciences*, 561:326–351, 2021. 10

[38] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 36–42, Cham, 2016. Springer International Publishing. 2

[39] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018. 4

[40] X. Zhou, V. Koltun, and P. Krähenbühl. Tracking objects as points. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 474–490, Cham, 2020. Springer International Publishing. 3, 9, 10

[41] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points, 2019. 4, 6

[42] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang. Online multi-object tracking with dual matching attention networks. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 379–396, Cham, 2018. Springer International Publishing. 3