Rethinking the Competition Between Detection and ReID in Multiobject Tracking

Chao Liang, Zhipeng Zhang, Xue Zhou, *Member, IEEE*, Bing Li, Shuyuan Zhu, *Member, IEEE*, and Weiming Hu, *Senior Member, IEEE*

***Abstract*— Due to balanced accuracy and speed, one-shot models which jointly learn detection and identification embeddings, have drawn great attention in multi-object tracking (MOT). However, the inherent differences and relations between detection and re-identification (ReID) are unconsciously overlooked because of treating them as two isolated tasks in the one-shot tracking paradigm. This leads to inferior performance compared with existing two-stage methods. In this paper, we first dissect the reasoning process for these two tasks, which reveals that the competition between them inevitably would destroy task-dependent representations learning. To tackle this problem, we propose a novel reciprocal network (REN) with a self-relation and cross-relation design so that to impel each branch to better learn task-dependent representations. The proposed model aims to alleviate the deleterious tasks competition, meanwhile improve the cooperation between detection and ReID. Furthermore, we introduce a scale-aware attention network (SAAN) that prevents semantic level misalignment to improve the association capability of ID embeddings. By integrating the two delicately designed networks into a one-shot online MOT system, we construct a strong MOT tracker, namely CSTrack. Our tracker achieves the state-of-the-art performance on MOT16, MOT17 and MOT20 datasets, without other bells and whistles. Moreover, CSTrack is efficient and runs at 16.4 FPS on a single modern GPU, and its lightweight version even runs at 34.6 FPS. The complete code has been released at https://github.com/JudasDie/SOTS**

***Index Terms*— Multiobject tracking, reciprocal representation learning, scale-aware attention, one-shot, ID embedding.**

I. Introduction

MULTI-OBJECT tracking (MOT), aiming to estimate the locations and scales of multiple targets in a video sequence, is one of the most fundamental yet challenging tasks in computer vision [1]. It may be applied to many practical scenarios, such as intelligent driving, human-computer interaction, and pedestrian behavior analysis.

Convolutional Neural Network (CNN) based object detection and re-identification (ReID) have demonstrated excellent performance in multi-object tracking in the past few years. The related ReID-based trackers can be divided into two classes, *i.e.,* two-stage and one-shot structures. The two-stage models follow the tracking-by-detection paradigm (or more precisely, tracking-after-detection) [2]–[5], which divides MOT into two separate tasks, *i.e.,* detection and association. It first obtains bounding boxes of objects in each frame through an off-the-shelf detector, and then associates the candidate boxes with existing tracklets by matching the extracted identification (ID) embedding of each bounding box across frames. Although effective, two-stage methods suffer from massive computational cost, since the ID embedding extraction model (*e.g.*, ReID network [6]) needs to perform forward inference for each individual bounding box. Alternatively, the one-shot methods [7]–[10] integrate detection and ID embedding extraction into a unified framework. By redesigning the prediction head of the detector, the one-shot tracker can simultaneously yield detection results and ID embeddings, which may bring an increase in speed but at the same time a decrease in accuracy.

In this paper, we first dissect the reasoning process of one-shot tracker and analyze why the performance is lower than the two-stage methods. Our analysis reveals that the performance degradation primarily derives from excessive competition between detection and ReID tasks, which can be summarized into two aspects: **1) Competition of Object Representation Learning:** In one-shot methods, object class confidence, target scale and ID information are obtained by processing a shared feature tensor. Although one-shot methods

Manuscript received September 27, 2021; revised March 12, 2022; accepted March 23, 2022. Date of publication April 12, 2022; date of current version April 22, 2022. This work was supported in part by the Natural Science Foundation of China under Grant 61972071 and Grant U20A20184, in part by the Sichuan Science and Technology Program under Grant 2020YJ0036, in part by the Research Program of Zhejiang Laboratory under Grant 2019KDAB02, in part by the Open Project Program of the National Laboratory of Pattern Recognition under Grant 201900014, and in part by the Intelligent Terminal Key Laboratory of Sichuan Province under Grant SCITLAB-1005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Weiyao Lin. *(Chao Liang and Zhipeng Zhang contributed equally to this work.) (Corresponding author: Xue Zhou.)*

Chao Liang is with the School of Automation Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu 610056, China (e-mail: 201921060415@std.uestc.edu.cn).

Zhipeng Zhang and Bing Li are with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100045, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 101408, China (e-mail: zhangzhipeng2017@ia.ac.cn).

Xue Zhou is with the School of Automation Engineering and the Shenzhen Institute of Advanced Study, University of Electronic Science and Technology of China (UESTC), Chengdu 610056, China, and also with the Intelligent Terminal Key Laboratory of Sichuan Province, Yibin 644000, China (e-mail: zhouxue@uestc.edu.cn).

Shuyuan Zhu is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu 610056, China.

Weiming Hu is with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100045, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), Beijing 101408, China, and also with the CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai 200031, China.

Digital Object Identifier 10.1109/TIP.2022.3165376

1941-0042 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

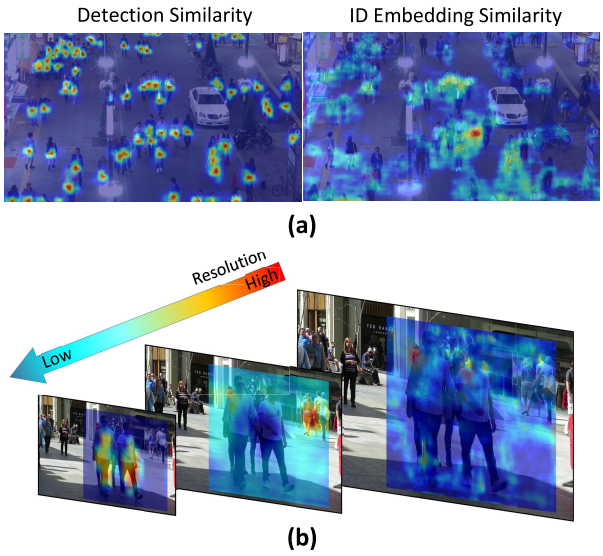


Fig. 1. Motivation of CSTrack. (a) visualizes the similarity maps about detection and ReID tasks respectively, where detection expects all the pedestrians have high response values whereas ReID tends to focus on the specific pedestrian. From this point of view, they are contradictory to each other. (b) represents that different resolution focuses on different scale of targets in FPN-based model, where the arrow indicates the output resolution from high to low. This arrangement can help detector to detect pedestrians with different sizes, but not suitable for semantic matching of pedestrians with different sizes in ReID task.

are efficient, the inherent differences between different tasks are ignored, as shown in Fig. 1 (a). Specifically, object detection requires that objects belonging to the same category (*e.g.*, pedestrian) have similar semantics, while ReID tends to distinguish two different pedestrian, which is contradictory to the purpose of detection task. The competition between the above two tasks may lead to ambiguous learning, which means that the pursuit of high performance in one task may result in performance degradation or stagnation in the other. **2) Competition of Semantic Level Assignment:** The one-shot trackers [7]–[9] usually convert detector to tracker by adding a parallel branch to yield ID embeddings. Although it is simple, it ignores that semantic level assignment of detection task is not suitable for ReID task. Specifically, modern object detectors always introduce feature pyramid networks (FPNs) [11] to improve localization accuracy for various target sizes. In this framework, objects of different scales will be assigned to different resolution features, as shown in Fig. 1 (b). However, we observe that this assignment is not suitable for ReID in the one-shot tracker since it will lead to a semantic level misalignment, whether for different targets or the same target across frames. Existing one-shot trackers [7], [8] usually extract ID embedding from a certain feature resolution depending on detection results, which may lead to a semantic level gap when matching targets with drastic scale changes. Furthermore, it will impair the performance of subsequent data association (Please see the ablation experiments in Sec. IV-B for detailed discussion and verification).

To tackle the above problems, we fully consider the essential differences of two tasks in one-shot tracker, and design two sub-networks to solve the above problems. Specifically, we propose a novel reciprocal network (REN) to conduct collaborative learning among different tasks. The reciprocal network separates the feature maps of object detection and ID embedding extraction into two different task-driven branches, by which the task-dependent representations can be learned. Specifically, given shared features, a novel structure combining self-relation and cross-relation is designed to enhance the feature representation. The former impels hidden nodes to learn task-dependent features, and in parallel the latter aims to improve the collaborative learning of these two tasks. Meanwhile, we design a scale-aware attention network (SAAN) to improve the alignment of ID embedding extraction, and enhance the model resilience to scale changes. In this work, we develop an architecture that combines spatial and channel attention to achieve our goal, and this architecture enhances the influence of object-related regions and channels. The output of SAAN is a feature tensor that includes all the targets with rich semantics in all resolutions. Following the above design idea, aggregation feature-based ID embedding is helpful to prevent semantic misalignment during matching.

The main contributions of our work are three-folds:

- We propose a novel reciprocal network to learn task-dependent representations. It not only effectively mitigates the deleterious competition, but also improves the capability of collaborative learning between detection and ReID tasks in one-shot MOT methods.
- We introduce a scale-aware attention network to apply spatial and channel attention to feature maps at different resolutions. By fusing these features and extracting ID embeddings from a consistent representation, it effectively prevents semantic level misalignment, and improves the resilience to objects with different scales during matching.
- The extensive experiments demonstrate that our method effectively improves the performance of the one-shot MOT method, especially for association ability of re-identification features, which is competitive with the two-stage methods.

The organization of this paper is as follows: Section II reviews related work in the literature. Section III presents our proposed approach. The experimental results and analysis are provided in Section IV. Section V concludes the paper.

II. Related Work

We firstly review the related MOT methods, including two-stage and one-shot structures. After that, we briefly review the related works for joint detection and tracking without ReID in MOT task.

A. Two-Stage MOT Methods

With the development of object detectors, many MOT trackers follow the tracking-by-detection paradigm (or more precisely, tracking-after-detection). In particular, these trackers [2]–[5], [12], [13] divide MOT into two separate tasks, *i.e.,* detection and association. The object bounding boxes are firstly predicted by high-performance detectors such as Faster R-CNN [14], SDP [15] and DPM [16]. Then, the candidate boxes are linked into tracklets across frames by

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

an association network. Since the candidate boxes can be directly yielded by off-the-shelf detector, two-stage MOT methods focus on improving association performance. In the early works, Sort [12] firstly performs Kalman filtering to predict candidate boxes in the next frame and uses Hungarian method to associate across frames by measuring bounding box overlap. IOU-Tracker [13] directly associates the last frame bounding box with candidate boxes in the current frame using IOU matching. Although simple and efficient, both Sort and IOU-Tracker may fail in challenging scenarios such as incomplete detections or crowded scenes since they rely on the nearest neighbor hypothesis (*i.e.,* only the candidate box closest to the target has the same ID). Therefore, some works [2]–[5] attempt to apply ReID network to yield more robust predictions. In a nutshell, each target is cropped from the image and fed into an additional network to extract ID appearance embedding, which is employed to associate with existing tracklets by measuring the cosine distance across frame. Moreover, some works [17], [18] introduce graph neural networks to further improve matching capacity. For example, MPN [18] replaces the Hungarian algorithm with graph neural networks to dissect the information and associate them with the edge classification.

Although the above methods achieve impressive performance, they still suffer from massive computation costs since the ID information extraction network needs to perform forward inference for each bounding box. In contrast, one-shot trackers consider constructing an entire model to simultaneously generate detection boxes and ID embeddings, which exhibits attractive efficiency.

B. One-Shot MOT Methods

Due to balanced accuracy and speed, the one-shot paradigm [7]–[10], which integrates detection and ID embedding extraction into a unified network, offer a new solution for MOT. Tong *et al.* [7] first modify a detector (*i.e.,* Faster RCNN [14]) to handle both detection and ReID tasks. By introducing extra fully-connected layers (FC), the detector obtains the ability to yield ID embeddings. Recently-proposed JDE [8] and RetinaTrack [9] convert a FPN-based detector (*i.e.,* Yolo-v3 [19] and RetinaNet [20]) to a one-shot tracker by redesigning the prediction head. Although those simple modifications help generate detection and ID embeddings with a small overhead and achieve impressive performance, they ignore the inherent differences between detection and ReID. This induces their performance not so good as the two-stage methods, especially evaluated by the IDF1 score. Recent released method, FairMOT [10], uses the anchor-free method [21] to reduce the ambiguity of anchors. Despite alleviating the scale-aware competition by aggregating multi-layer features to yield ID embeddings, it fails to handle the competition between object detection and ReID.

C. Joint Detection and Tracking Without ReID

Besides the one-shot MOT methods, several methods [22]–[25] attempt to train joint detection and tracking models by integrating other association ways instead of classical ReID. For instance, CTracker [22] uses adjacent frame pairs as input and generates a pair of bounding boxes for the same target by a chaining structure. TubeTK [23] utilizes tubes to encode the target's temporal-spatial position and local moving trail. Tracktor [24] converts existing object detectors (*i.e.,* Faster-RCNN [14]) to trackers by exploiting the regression head of detector to perform the regression of the object boxes from last frame to current image. The following work, CenterTrack [25], which shares a similar spirit with Tracktor [24], predicts the location of the existing objects by learning the offset of heat points (the center point of object boxes). However, their association performances are also inferior to two-stage methods due to the poor capability of the direct bounding boxes transduction compared with ReID.

III. Methodology

Our proposed framework, named **CSTrack**, consists of two main components, *i.e.*, reciprocal network designed in Sec. III-B and scale-aware attention network described in Sec. III-C. Before shedding light on the two essential parts, we first give a brief overview of the whole framework in Sec. III-A. Moreover, we describe the details of training and online inference in Sec, III-D and Sec, III-E, respectively.

A. Overview

Before describing our model, we review the popular JDE architecture [8] (Joint Detection and Embeddings), which is the baseline of our method, and explain why the vanilla model is not suitable for mitigating the competition between detection and ReID tasks. And then, we introduce the CSTrack framework, which strikes for a better balance between detection and ReID tasks by integrating our proposed sub-networks.

1) Preliminaries: JDE [8] redesigns the prediction head of an object detector to conduct object detection and ID embedding extraction simultaneously in a one-shot model. Specifically, given a frame $\boldsymbol{x}$, it is first processed by a feature extractor Ψ (including *Backbone* and *Neck*), which yields multi-resolution features $\boldsymbol{F}_i|_{i=1,2,3}$ (see Fig. 2 (a)),

$$\boldsymbol{F}_i|_{i=1,2,3} = \Psi(\boldsymbol{x}). \tag{1}$$

Then, $\boldsymbol{F}_i|_{i=1,2,3}$ is fed into the *Head* network (*i.e.*, including the parallel object detection and ReID branches) to predict detection result $\boldsymbol{D}$ and raw ID embedding maps $\boldsymbol{E}$, as illustrated in Fig. 2 (b). The detection result will be post-processed by non maximum suppression (NMS) [14] to yield candidate boxes. Each candidate box extracts its corresponding ID embedding from the row ID embedding maps $\boldsymbol{E}$ (following the same feature resolution level as the candidate box) to link with the existing tracklets.

In this framework, the shared features are directly fed into two independent task-driven branches, which generate the output directly applying a 1×1 convolution layer, ignoring the inherent differences between these two tasks. It may lead to ambiguous learning during training. Moreover, ReID branch extracts ID embeddings from probable arbitrary feature level guided by detection task, which may induce semantic level misalignment during matching.

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

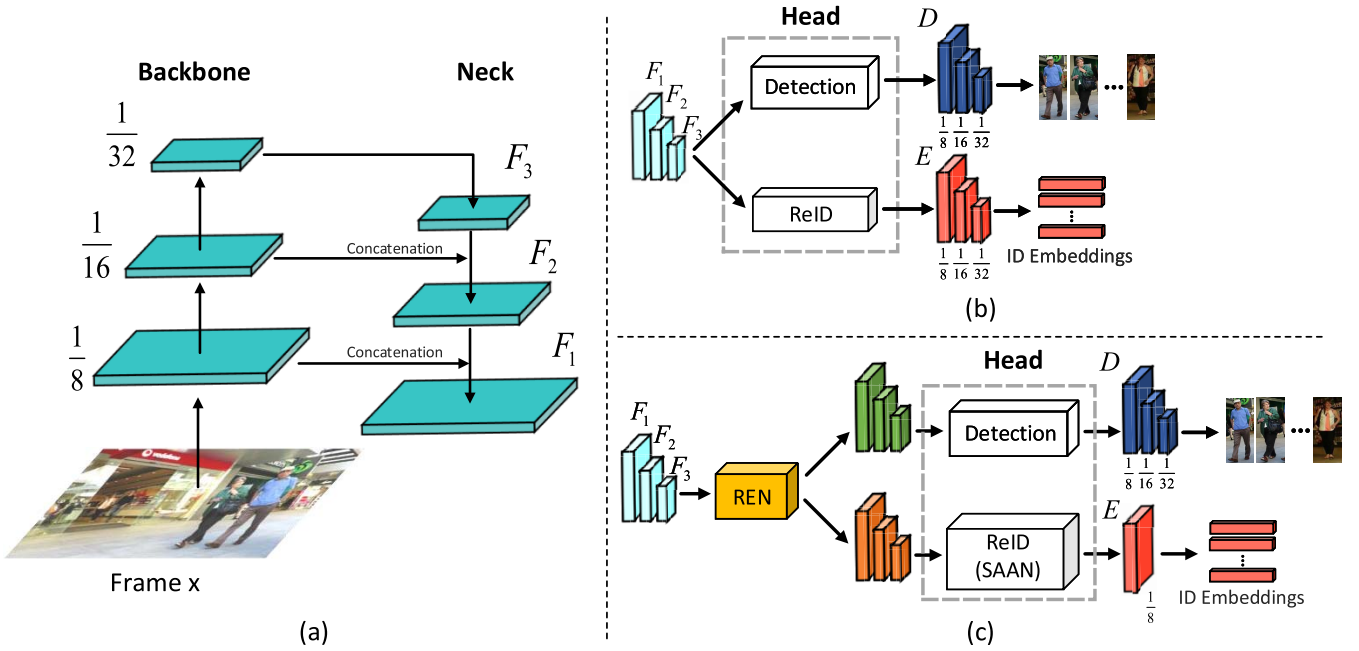


Fig. 2. Architecture diagrams. (a) is the feature extractor including backbone and neck (FPN). (b) illustrates the vanilla prediction structure of JDE. (c) illustrates our proposed prediction structure of CSTrack. Different from JDE, CSTrack introduces a reciprocal network (REN) to learn task-dependent representations and a scale-aware attention network to generate discriminative embeddings, which efficiently mitigate the competition.

2) CSTrack: We build our framework based on JDE [8], as shown in Fig. 2 (c). To alleviate the competition of object representation learning and enhance task-dependent representations, we propose a novel reciprocal network (REN) to decouple the feature $\boldsymbol{F}_t$ before feeding it into different task branches and then exchange semantic information of different tasks with a cross-relation layer. The idea of designing REN is inspired by recent self-attention [26] and multi-task decoding mechanisms [27], which can enhance representations for each task with only small overheads. Note that, REN not only focuses on the specificities of features for different tasks, but also learns commonalities to encode the feature by constructing the cross-relation weight maps. For ReID branch, instead of applying a 1×1 convolution layer on each feature resolution, we design a scale-aware attention network (SAAN, detailed in Fig. 4) to fuse features from different feature resolutions. Meanwhile, both spatial and channel-wise attention modules [28] are adopted to suppress noisy background and learn object-related representations.

B. Reciprocal Network

In this section, we propose a reciprocal network to learn both specificities (task-dependent) and commonalities (collaboration) of features for detection and ReID tasks. For specificities learning, self-relation reflecting correlations between different feature channels are learned to enhance feature representation for each task. For commonalities learning, a cross-relation layer is elaborately designed to exchange the semantic information of different tasks.

The structure of our reciprocal network is illustrated in Fig. 3, where the features outputted from feature extractor are denoted as $\boldsymbol{F}_i|_{i=1,2,3} \in \mathbb{R}^{C\times H\times W}$. First, we pass $\boldsymbol{F}_i$ through

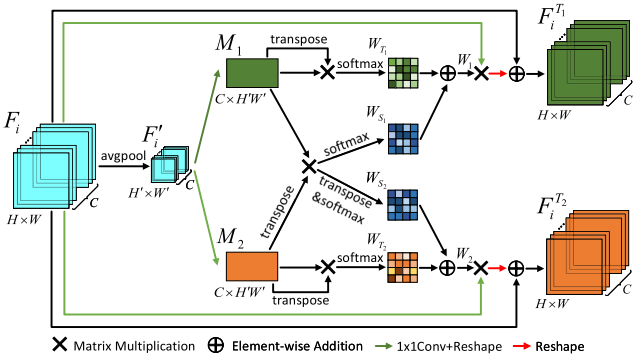


Fig. 3. Diagram of reciprocal network (REN). For the original feature map $\boldsymbol{F}_i$, we construct the self-relation and cross-relation maps to impel the generation of task-dependent features $\boldsymbol{F}_i^{T_1}$ and $\boldsymbol{F}_i^{T_2}$.

an *avg-pooling* layer to obtain the statistical information $\boldsymbol{F}_i' \in \mathbb{R}^{C\times H'\times W'}$. Second, tensors $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ for detection and ReID are respectively generated by passing $\boldsymbol{F}_i'$ through different convolution layers and reshape them to $C \times N'$, where $N' = H' \times W'$. Third, we perform matrix multiplication on $\boldsymbol{M}_1/\boldsymbol{M}_2$ (/ indicates "or") and its corresponding transpose tensor. A row *softmax* layer is followed to calculate the self-relation weight maps $\{\boldsymbol{W}_{T_1}, \boldsymbol{W}_{T_2}\} \in \mathbb{R}^{C\times C}$ for each task, and the calculation is as follows:

$$w_{T_k}^{ij} = \frac{\exp\left(\boldsymbol{m}_k^i \cdot \boldsymbol{m}_k^j\right)}{\sum_{j=1}^{C} \exp\left(\boldsymbol{m}_k^i \cdot \boldsymbol{m}_k^j\right)}, \quad k \in \{1, 2\} \tag{2}$$

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

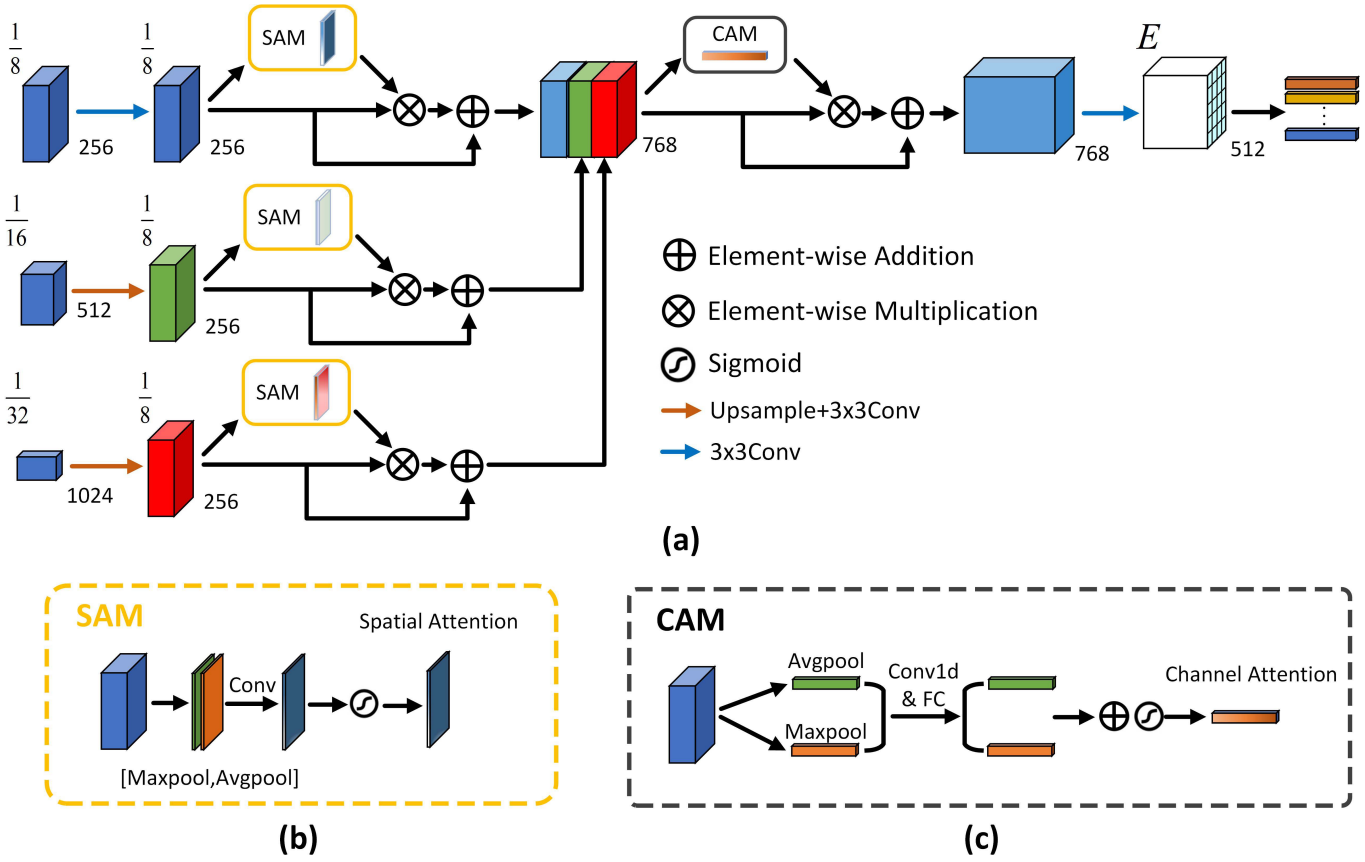


Fig. 4. The details of scale-aware attention network (SAAN). Wherein, (a) is the overall structure of the network, (b) is the diagram of spatial attention module (SAM), and (c) is the diagram of channel attention module (CAM).

where $\cdot$ denotes dot product operation. $\boldsymbol{m}_k^i$ and $\boldsymbol{m}_k^j$ indicate the i^{th} and j^{th} row of $\boldsymbol{M}_1/\boldsymbol{M}_2$, respectively. $w_{T_k}^{ij}$ denotes the value at location of (i, j) on $\boldsymbol{W}_{T_k}$, which represents the relation of the i^{th} and j^{th} channel in a tensor. Then, we perform matrix multiplication between $\boldsymbol{M}_{1/2}$ and the transpose of $\boldsymbol{M}_{2/1}$ to learn commonalities between different tasks, and then a row *softmax* layer is followed to generate cross-relation weight maps $\{\boldsymbol{W}_{S_1}, \boldsymbol{W}_{S_2}\} \in \mathbb{R}^{\mathrm{C}\times\mathrm{C}}$,

$$w_{S_k}^{ij} = \frac{\exp\left(\boldsymbol{m}_k^i \cdot \boldsymbol{m}_h^j\right)}{\sum_{j=1}^{C} \exp\left(\boldsymbol{m}_k^i \cdot \boldsymbol{m}_h^j\right)}, \quad (k, h) \in \{(1, 2), (2, 1)\} \quad (3)$$

where $w_{S_k}^{ij}$ denotes the effect of the i^{th} feature channel of task 1/2 on the j^{th} feature channel of task 2/1. Finally, the self-relation and cross-relation weights are finally fused by a trainable parameter λ, obtaining $\{\boldsymbol{W}_1, \boldsymbol{W}_2\} \in \mathbb{R}^{C\times C}$

$$\boldsymbol{W}_k = \lambda_k \times \boldsymbol{W}_{T_k} + (1 - \lambda_k) \times \boldsymbol{W}_{S_k}, \ k \in \{1, 2\}. \quad (4)$$

The original feature map $\boldsymbol{F}_i$ is processed by one 1×1 convolution layer and rearranged to the shape of $\mathbb{R}^{C\times N}$, where $N = H \times W$. Then, we perform matrix multiplication between the reshaped feature and the learned weight maps $\boldsymbol{W}_{1/2}$ to obtain an enhanced representation for each task. The enhanced representation is rearranged to the same shape as the original feature map $\boldsymbol{F}_i$ (*i.e.,* $\mathbb{R}^{C\times H\times W}$) and fused with original $\boldsymbol{F}_i$ by residual attention to prevent information loss. The feature tensors $\boldsymbol{F}_i^{T_i}$ and $\boldsymbol{F}_i^{T_2}$ will be sent to different task branch for further processing, respectively.

C. Scale-Aware Attention Network

We build a scale-aware attention network (SAAN), as illustrated in Fig. 4 to aggregate features from different resolutions, which guarantees semantic level alignment of ID embeddings from objects with different sizes. In this network, we introduce the spacial attention module (SAM) [28] and channel attention module (CAM) to learn 'where' and 'what' feature is more important for yielding discriminative ID embeddings. In particular, the features from the scale of 1/16 and 1/32 (compared to the size of input image) are upsampled to 1/8 firstly. Then, a 3×3 convolutional layers are followed to encode the upsampled feature maps. For better 'awaring' useful information for each target in different resolutions, we firstly arrange SAM to enhance the target-related features and suppress background noise. Specially, we perform *avg-pooling* and *max-pooling* on channel dimension to yield two *2D* maps with size $1 \times H \times W$. These maps are concatenated and processed by a 7×7 convolutional layer and a *Sigmoid* layer sequentially to generate a spatial attention map. The obtained spatial attention map learned individually in each resolution and is fused with the original feature by element-wise multiplication and residual attention. Then, we concatenate

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

the feature maps from different scales and pass them through the channel attention module. The channel attention module comprises global *avg-pooling* and *max-pooling* layers, which learn different statistical information of the different resolution features. The outputs of pooling layers are first processed by the shared network consisting of a *1D convolutional* layer and a *fully-connected* layer. Then, the feature maps will be fused by element-wise addition and normalized by a *Sigmoid* layer to generate an attention map with only one channel, which is applied to the vanilla features similar to the spacial attention map. For the arrangement of the attention module, the experimental result shows that our design is helpful to improve association ability of ID embeddings, as discussed in Sec. IV-B.

Finally, we use a 3×3 convolution layer to map features to 512 channels, as $\boldsymbol{E} \in \mathbb{R}^{512 \times W \times H}$. Each ID embedding $\boldsymbol{e_{xy}} \in \mathbb{R}^{512 \times 1 \times 1}$ denotes the identity information of the object at location (x, y), which can be extracted according to detection results for the subsequent ReID task.

D. Training Details

For jointly optimizing the object detection and ReID tasks, we minimize a weighted sum of detection losses (consisting of classification loss and box regression loss) and ReID loss. Specifically, the ground-truth annotation for an input image is denoted as $\{B_i\}$, where $B_i = (x^{(i)}, y^{(i)}, w^{(i)}, h^{(i)}, c^{(i)})$. Here, $(x^{(i)}, y^{(i)})$ indicate the coordinates of the center of bounding box and $(w^{(i)}, h^{(i)})$ indicate the size of bounding box. $c^{(i)}$ is the ID index that the object belongs to the i^{th} pedestrian. C is the number of ID in training dataset, which denotes the number of pedestrians. For each location (x, y) on the detection result maps of different resolution, we represent it as a 5D vector $\boldsymbol{t} = (x^*, y^*, w^*, h^*, p)$, following the standard anchor-based protocol [19]. Here (x^*, y^*, w^*, h^*) are the bounding box predictions and p indicates the foreground probability of the bounding box. For foreground and background classification, we define the anchor points in the location $(x, y) = (\left\lfloor \frac{x^{(i)}}{r} \right\rfloor, \left\lfloor \frac{y^{(i)}}{r} \right\rfloor)$ as positive samples (r is the downsample ratio, *i.e.,* 8, 16, 32, and $\lfloor \cdot \rfloor$ indicates rounding down to an integer). Therefore, the classification loss [20] can be formulated as:

$$\mathcal{L}_{cls}(p_t) = -\alpha\,(1 - p_t)^{\gamma}\,\log(p_t), \tag{5}$$

where

$$p_t = \begin{cases} p & \text{if } (x, y) = (\left\lfloor \frac{x^{(i)}}{r} \right\rfloor, \left\lfloor \frac{y^{(i)}}{r} \right\rfloor) \\ 1 - p & \text{otherwise} \end{cases}. \tag{6}$$

We set $\alpha = 0.25$ and $\gamma = 0$ following YOLO-v5. For bounding box regression, we employ the CIOU loss [29], which can be defined as,

$$\mathcal{L}_{reg}(\hat{b}_{x,y}) = \begin{cases} 1 - \mathbb{C}(b_i, \hat{b}_{x,y}) & \text{if } (x, y) = (\left\lfloor \frac{x^{(i)}}{r} \right\rfloor, \left\lfloor \frac{y^{(i)}}{r} \right\rfloor) \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

where $\mathbb{C}$ represents the CIOU operation. b_i indicates the ground-truth box and $\hat{b}_{x,y}$ is the box prediction at location (x, y) of feature maps. We define our detection loss as follows,

$$\mathcal{L}_{det} = \frac{1}{N_{pos}} \sum_{i}^{M} \sum_{x,y} \left(\mathcal{L}_{cls}\left(p^i_{x,y}\right) + \beta \mathcal{L}_{reg}\left(\hat{b}^i_{x,y}\right) \right), \tag{8}$$

where M is the number of the resolutions (*i.e.,* 3 in our model) and N_{pos} denotes the number of positive samples. β indicates the loss weight. We set $\beta = 0.05$ as that in YOLO-v5.

The definition and training of ReID loss follow the baseline tracker JDE [8], which uses the cross-entropy loss as the objective for ID embedding learning. In this work, we model this task as a classification task, and use a *FC* layer to map ID embeddings to a class distribution vector $P = \{p(c), c \in [1, 2, \ldots, C]\}$. By comparing with the one-hot representation of the GT class label $Y^i(c) \in \mathbb{R}^{C \times 1 \times 1}$, we compute the ReID loss as

$$L_{id} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} Y^i(c) \log(p(c)), \tag{9}$$

where N denotes the number of objects in the current image.

The joint loss can be written as a weighted linear sum of detection loss L_{det} and ReID loss L_{id}, as

$$\mathcal{L}_{total} = \mathcal{L}_{det} + \eta \mathcal{L}_{id}, \tag{10}$$

where η is experimentally set to 0.02 to balance object detection and ReID tasks.

E. Online Tracking

In this section, we present how to link the detected candidate boxes with existing tracklets. For a fair comparison, we follow the cascade matching design in JDE [8], as shown in Fig. 5. Specifically, we first calculate the similarity of ID embeddings between candidate boxes and templates (*i.e.*, the ID embeddings of the previous tracklets) with the cosine metric. Then a Kalman filter is used to provide distance restriction, which removes unreasonable matches between candidate boxes and existing tracklets. After that, the Hungarian algorithm is employed to find the optimal bipartite matching. In the second matching stage, the unmatched tracklets will be re-checked again by an IOU matching, as in Sort [12]. For the successful matched tracklets, the templates will be updated to adapt to appearance variations, as

$$t^t_i = \varepsilon t^{t-1}_i + (1 - \varepsilon) e^t_i, \tag{11}$$

where e^t_i is an ID embedding of candidate box and i indicates the ID index of target. t^{t-1}_i and t^t_i represent the corresponding templates before and after update, respectively. ε is a weight term, which is set as 0.9, following JDE [8]. The unmatched candidate boxes will be initialized as new tracklets, where the corresponding ID embeddings are simultaneously initialized as the template of new tracklets. The unmatched tracklets are set as the 'inactive' state. If a tracklet cannot find its matched box for 30 frames, it will be terminated. On the contrary, the unmatched tracklet will be restored to'active' state when it is successfully matched with a candidate box before terminating. The successful matched tracklets, the new tracklets, and the 'inactive' tracklets will be used to link with the candidate boxes in the next frame.

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

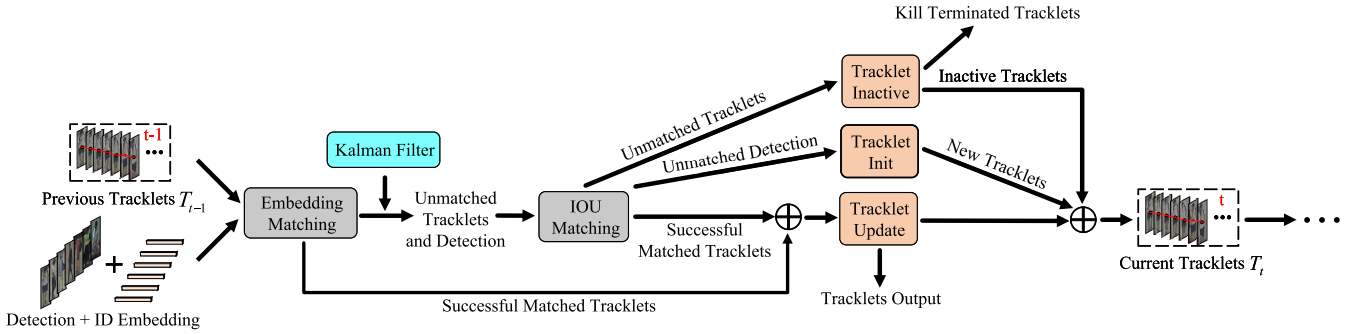


Fig. 5. The details of online tracking. For the detected candidate boxes, we will link them with existing tracklets by a cascade matching design following JDE [8].

IV. Experiments

To demonstrate the advantages of the proposed CSTrack, we firstly study the effectiveness of each component in our tracking framework in Sec. IV-B. Then we compare our method with state-of-the-art approaches in Sec. IV-C. After that, we further analyze the data association ability of our framework in Sec. IV-D. Finally, we visualize the qualitative results of CSTrack and analyze the failure cases.

A. Implementation Details

1) Training: For a fair comparison, we use the same training datasets as the baseline tracker JDE [8], consisting of ETH [30], CityPerson [31], CalTech [32], MOT17 [33], CUDK-SYSU [7] and PRW [34]. Wherein, ETH and CityPerson only provide box annotations, so we only use them to train the detection branch. The other four datasets provide both detection and ID annotations, allowing us to train both object detection and ReID tasks. We chop off the videos in ETH [30] that are overlapped with the testing benchmark MOT16 [33]. For further improving performance, we introduce CrowdHuman [35] into training. We train our network with the SGD optimizer for 30 epochs on a single RTX 2080 Ti GPU. The batch size is set to 8. The initial learning rate is 5×10^{-4}, and we decay the learning rate to 5×10^{-5} at the 20th epoch. The hyperparameters of the object detector are set the same as Yolo-v5 and others follow IDE [8]. The *Backbone* and *Neck* networks are initialized with the parameters pre-trained on COCO [36]. For the reciprocal network, we set the feature size after *avg-pooling* layer as $(H', W') = (6, 10)$ and initialize the trainable parameters $\lambda_{1/2} = 0.5$.

2) Testing and Evaluation Metrics: We evaluate our network on MOT16 [33], MOT17 [33] and the recently-released MOT20 [37]. Specifically, MOT16 [33] and MOT17 [33] contain the same 7 testing videos, yet some videos are re-annotated. MOT20 [37] consists of 4 testing videos under extremely crowded scenes. Following the common practices in MOT Challenge,[1] we adopt the CLEAR metric [38], particularly Multiple Object Tracking Accuracy (MOTA) to evaluate the overall performance. ID F1 Score (IDF1) [39] is employed to evaluate the association performance. We also report other metrics including the Most Tracked ratio (MT) for the ratio of most tracked (> 80% time) objects, Most Lost ratio (ML) for most lost (<20% time) objects, the number of Identity Switch (ID Sw.) [40] and FPS for measuring frame rate of the overall system.

Our tracker is implemented using Python 3.7 and PyTorch 1.6.0. The experiments are conducted on a single RTX 2080Ti GPU and Xeon Gold 5218 2.30GHz CPU.

TABLE I
Component-Wise Analysis of CSTrack on MOT16 Testing Set

NUM	Method	REN	SAAN	MOTA↑	IDF1↑	ID Sw.↓
①	JDE(yolo-v3)			64.4	55.8	1544
②	JDE(yolo-v5)			68.9	60.7	1798
③	JDE(yolo-v5)	✓		70.8	63.1	1365
④	JDE(yolo-v5)		✓	69.4	69.3	1226
⑤	JDE(yolo-v5)	✓	✓	**72.9**	**71.6**	**1121**

B. Ablation Studies [2]

1) Component-Wise Analysis: In this section, we verify the effectiveness of our proposed reciprocal network (REN in Sec. III-B) and scale-aware attention network (SAAN in Sec. III-C). For a fair comparison with JDE, all experiments are trained using the six datasets mentioned above and validated in the MOT16 testing set [33]. The comparison experimental results are presented in Tab. I. We first replace the object detector in JDE [8] from YOLO-v3 [19] to YOLO-v5, which ensures a better performance and faster inference speed (① *vs.* ②). The YOLO-v5 replacement provides a strong baseline for our following design. When equipped with the proposed reciprocal network (REN), it achieves 1.9 points gains on MOTA, 2.4 points gains on IDF1 and the IDs decrease from 1798 to 1365 (② *vs.* ③). This demonstrates the effectiveness of our REN component in learning task-dependent representations to improve both detection and association performance. The introduced SAAN module aims to avoid the semantic gap when matching ID embeddings, and it brings

[1] https://motchallenge.net

[2] Due to the limited submissions of the MOT Challenge (totally 4 times per model on each benchmark), some ablation studies are verified on validation set.

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

TABLE II

ABLATION STUDY OF REN ON MOT17 VALIDATION SET. SR REPRESENTS SELF-RELATION, CR REPRESENTS CROSS-RELATION

NUM	SR	CR	λ_1	λ_2	MOTA↑	IDF1↑	ID Sw.↓
①	✓		–	–	65.3	69.5	713
②		✓	–	–	64.8	67.2	616
③	✓	✓	0.5	0.5	64.5	68.9	679
④	✓	✓	0.12122	0.31519	**66.0**	**70.7**	**535**
⑤	w/o REN				64.6	68.7	739
⑥	replace REN with CNN				64.9	69.0	717

TABLE III

IMPACT OF ATTENTION MODULE ARRANGEMENT IN OUR TRACKER, WHERE THE MODULE WILL BE ARRANGED SERIALLY, AS [28]

NUM	Each resolution		concatenate		MOTA↑	IDF1↑	ID Sw.↓
	SAM	CAM	SAM	CAM			
①					64.1	66.4	723
②	✓				66.0	70.4	633
③		✓			65.4	68.4	659
④	✓	✓			64.9	69.8	690
⑤	✓		✓		64.1	69.3	721
⑥	✓			✓	**66.0**	**70.7**	**535**
⑦	✓		✓	✓	64.3	69.9	700
⑧	Basic ReID Head (w/o SAAN)				63.8	61.3	1047

TABLE IV

ANALYSIS OF TRAINING DATASET SETTINGS ON MOT16 TESTING SET. "S" INDICATES ONLY USING MOT17 FOR TRAINING. "M" INDICATES USING SIX DATASETS MENTIONED IN THE MANUSCRIPT. "B" DENOTES TRAINING ON THE SEVEN DATASETS INCLUDING "M" AND CROWDHUMAN DATASET. IT VERIFIES THAT CSTRACK CAN ACHIEVE EFFICIENT PERFORMANCE, EVEN ONLY USING MOT17 FOR TRAINING

NUM	Settings	image	bbox	ID	MOTA↑	IDF1↑	ID Sw.↓
①	JDE (M)	54K	270K	8.7K	64.4	55.8	1544
②	CSTrack (S)	5K	112K	0.5K	71.3	68.6	1356
③	CSTrack (M)	54K	270K	8.7K	72.9	71.6	**1050**
④	CSTrack (B)	73K	740K	8.7K	**75.6**	**73.3**	1121

8.6 points gains on IDF1 (③ *vs.* ⑤). Compared with the vanilla JDE [8], our tracker significantly improves tracking performance (① *vs.* ⑤), *i.e.,* MOTA +8.5%, IDF1 +15.8% and ID Sw. decreased from 1544 to 1121.

2) Self-Relation and Cross-Relation Analysis: To verify the effectiveness of self-relation and cross-relation in our tracking framework, we conduct experiments with different settings and present the comparison results in Tab. II. We train all the models on half of the training set and validate on another half. Compared with the model without REN (① *vs.* ⑤), employing self-relation to learn specificities can bring 0.7% gains on MOTA and 0.8% gains on IDF1, respectively. When only introducing the cross-relation, our method sightly outperforms the model without REN by 0.2% on MOTA, but IDF1 decreases from 68.7% to 67.2% (② *vs.* ⑤). The reason for this degradation is that only learning commonalities is insufficient

TABLE V

IMPACT OF ID LOSS WEIGHT SETTING IN OUR TRACKER. η INDICATES THE ID LOSS WEIGHT

NUM	η	MOTA↑	IDF1↑	ID Sw.↓
①	1	21.1	35.2	966
②	0.8	33.5	45.7	1002
③	0.6	54.2	60.5	693
④	0.4	55.0	62.3	683
⑤	0.2	64.5	69.0	713
⑥	0.1	65.3	69.1	642
⑦	0.05	65.2	70.3	652
⑧	0.02	66.0	**70.7**	**535**
⑨	0.01	**66.2**	68.6	701
⑩	0.005	65.8	67.2	742

to resolve the underlying competition between object detection and ReID. When we merge self-relation and cross-relation with equal weights (*i.e.,* we set $\lambda_{1/2}$ as constant weights 0.5, see ③), the performance is inferior to the model only introducing self-relation (① *vs.* ③). But surprisingly, when introducing learning weights to adjust the fusion of self-relation and cross-relation, we obtain promising results with MOTA of 66.0% and IDF1 of 70.7%. Notably, the ID Sw. decreased about 25% (① *vs.* ④), demonstrating the effectiveness of our reciprocal network. It is noted that the learning weights λ_1 and λ_2 converge to 0.12122 and 0.31519, respectively. This reveals that different tasks have different requirements for features. Meanwhile, it verifies the specificities and commonalities learning can efficiently learn task-dependent representations, which improves the performance remarkably. What's more, to further verify the effectiveness of our reciprocal network, we replace REN with CNN module which has a nearly similar computational cost (74.6M and 1304 GFlops for REN version *v.s.* 77.2M and 1308 GFLOPs for CNN version). We observe that REN can achieve better results, even with fewer parameters and FLOPs (④ *vs.* ⑥).

3) Impact of Attention Module Arrangement: In order to understand the impact of the attention module arrangement in SAAN (see Sec. III-C), we conduct ablation studies on MOT17 validation set [33]. As shown in Tab. III, compared with one-shot tracker equipped with the basic ReID head, our model achieves better MOTA and IDF1 (①∼⑦ *vs.* ⑧). This verifies the effectiveness of our designed SAAN network in learning discriminative embeddings. As our core motivation is to robustly aggregate information from different resolutions, we validate our model with different attention module arrangements. Firstly, we configure different combinations of spatial attention modules (SAM) and channel attention modules (CAM) at each resolution. The results show that introducing spatial attention outperforms the basic model by +1.9% in MOTA, +1.0% in IDF1 and ID Sw. decreased from 723 to 633, respectively (see ①∼④). This demonstrates that SAM module is useful to enhance the target-related features and suppress background noise. Considering results from ⑤ to ⑦, we observe that jointly applying channel attention

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

TABLE VI

COMPARISON WITH STATE-OF-THE-ART ONLINE MOT TRACKERS ON THE MOT16, MOT17 AND MOT20 BENCHMARKS. # INDICATES ONE-SHOT METHOD. ∗ INDICATES OTHER JOINT DETECTION AND TRACKING METHODS WHICH ADOPT NON-REID METHODS FOR DATA ASSOCIATION. OTHERS WITHOUT SPECIAL SIGN INDICATE TWO-STAGE METHODS. FP AND FN MEAN FALSE POSITIVE AND FALSE NEGATIVE, RESPECTIVELY. FPS DENOTES FRAME RATES FOR THE WHOLE TRACKING FRAMEWORK. "CSTRACK-S" INDICATES OUR LIGHTWEIGHT VERSION (DETAILED IN SEC. IV)

Dataset	Method	Published	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	ID Sw.↓	FPS↑
MOT16	SORT_POI [12]	ICIP2016	59.8	53.8	25.4	22.7	8698	63245	1423	<8.6
	POI [3]	ECCV2016	66.1	65.1	34.0	21.3	5061	55914	805	<5.2
	DeepSORT-2 [2]	ICIP2017	61.4	62.2	32.8	18.2	12852	56668	781	<6.7
	RAN [5]	WACV2018	63.0	63.8	39.9	22.1	13663	53248	**482**	<1.5
	TAP [4]	ICPR2018	64.8	**73.5**	38.5	21.6	12980	50635	794	<8.0
	IAT [42]	WACV2019	48.8	-	15.8	38.1	5875	86567	936	-
	CNNMTT [43]	MTA2019	65.2	62.2	32.4	21.3	6578	55896	946	<5.2
	∗Tracktor++ [24]	ICCV2019	56.2	54.9	20.7	35.8	**2394**	76844	617	1.8
	STPP [44]	TNNLS2020	50.5	-	19.6	39.4	5939	83694	638	-
	TPM [45]	PR2020	50.9	-	19.4	39.4	4866	84022	619	-
	∗TubeTK_POI [23]	CVPR2020	66.9	62.2	39.0	16.1	11544	47502	1236	1.0
	∗CTrackerV1 [22]	ECCV2020	67.6	57.2	32.9	23.1	8934	48305	1897	6.8
	#JDE [8]	ECCV2020	64.4	55.8	35.4	20.0	10642	52523	1544	18.8
	#FairMOTv2 [10]	IJCV2021	74.9	72.8	**44.7**	**15.9**	10163	34484	1074	18.9
	#CSTrack	Ours	**75.6**	73.3	42.8	16.5	9646	**33777**	1121	16.4
	#CSTrack-S	Ours	73.8	71.0	40.1	18.7	9157	37329	1313	**34.6**
MOT17	∗Tracktor++ [24]	ICCV2019	56.3	55.1	21.1	35.3	**8866**	235449	**1987**	1.8
	STPP [44]	TNNLS2020	52.4	-	22.4	40.0	20176	246158	2224	-
	TPM [45]	PR2020	52.4	-	22.4	40.0	19922	246183	2215	-
	∗TubeTK [23]	CVPR2020	63.0	58.6	31.2	19.9	27060	177483	4137	3.0
	∗CTrackerV1 [22]	ECCV2020	66.6	57.4	32.2	24.2	22284	160491	5529	6.8
	∗CenterTrack [25]	ECCV2020	67.8	64.7	34.6	24.6	18498	160332	3039	22.0
	#JDE [8]	ECCV2020	63.0	59.5	35.7	17.3	39888	162927	6171	18.8
	#FairMOTv2 [10]	IJCV2021	73.7	**72.3**	**43.2**	**17.3**	27507	117477	3303	18.9
	#CSTrack	Ours	**74.9**	**72.3**	41.5	17.5	23847	**114303**	3567	16.4
	#CSTrack-S	Ours	72.9	70.3	38.7	18.9	23148	125418	4119	**34.6**
MOT20	#JDE [8]	ECCV2020	40.4	33.5	8.9	25.3	24228	271853	12301	6.7
	#FairMOTv2 [10]	IJCV2021	61.8	67.3	**68.8**	**7.6**	103440	**88901**	5243	8.9
	#CSTrack	Ours	**66.6**	**68.6**	50.4	15.5	25404	144358	**3196**	4.5
	#CSTrack-S	Ours	64.2	65.0	47.4	18.4	**21088**	160253	3854	**12.4**

MOT17 #CStrack CPU 13th Gen Intel(R) CoreOl) i3-13100 (8 CPUs), N3.4GHz ~2

in aggregated feature can achieve even better results, further decreasing ID Sw. from 633 to 535.

4) Training Dataset Setting: We conduct another ablation study to evaluate the impact of different training dataset setting, and the corresponding results are presented in Tab. IV. According to the results of Tab. IV, when only using the MOT17 dataset [33] for training (see ②), our tracker achieves 71.3 MOTA and 68.6 IDF1, which outperforms many existing trackers. For a fair comparison, we employ the same setting as our baseline tracker JDE [8], *i.e.,* training on the six datasets mentioned in Sec. IV-A. From the result presented in Tab. IV (① *vs.* ③), our model surpasses JDE by 8.5 point in MOTA and 15.8 point in IDF1, and the ID Sw. decreased from 1544 to 1050. It further verifies that our design is simple and effective to improve tracking performance by alleviating the competition between object detection and ReID in the one-shot tracker. Moreover, we further add CrowdHuman [35] images into training, and observe that it brings gains of 2.7 point on MOTA and 1.7 points on IDF1 (③ *vs.* ④). One possible reason is that training with CrowdHuman dataset can improve model tracking performance in crowded scene.

5) Impact of ID Loss Weight: To jointly optimize object detection and target representation (*i.e.* ID embedding) learning, we balance their loss with a handcrafted weight η. To study the impact of η for ID embedding learning, an ablation experiment is conducted and the results are shown in Tab. V. ①∼⑤ demonstrate that a large ID loss weight η will degrade detection performance (see MOTA score), while a small ID loss weight degrades matching performance (see IDF1 and ID Sw. score). We set η to 0.02 considering better scoring across different evaluation criteria.

C. Comparisons With State-of-the-Art Trackers

We compare our multi-object tracker CSTrack with other start-of-the-art MOT tracking methods on MOT16 [33], MOT17 [33] and MOT20 [37] benchmarks. The compared

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

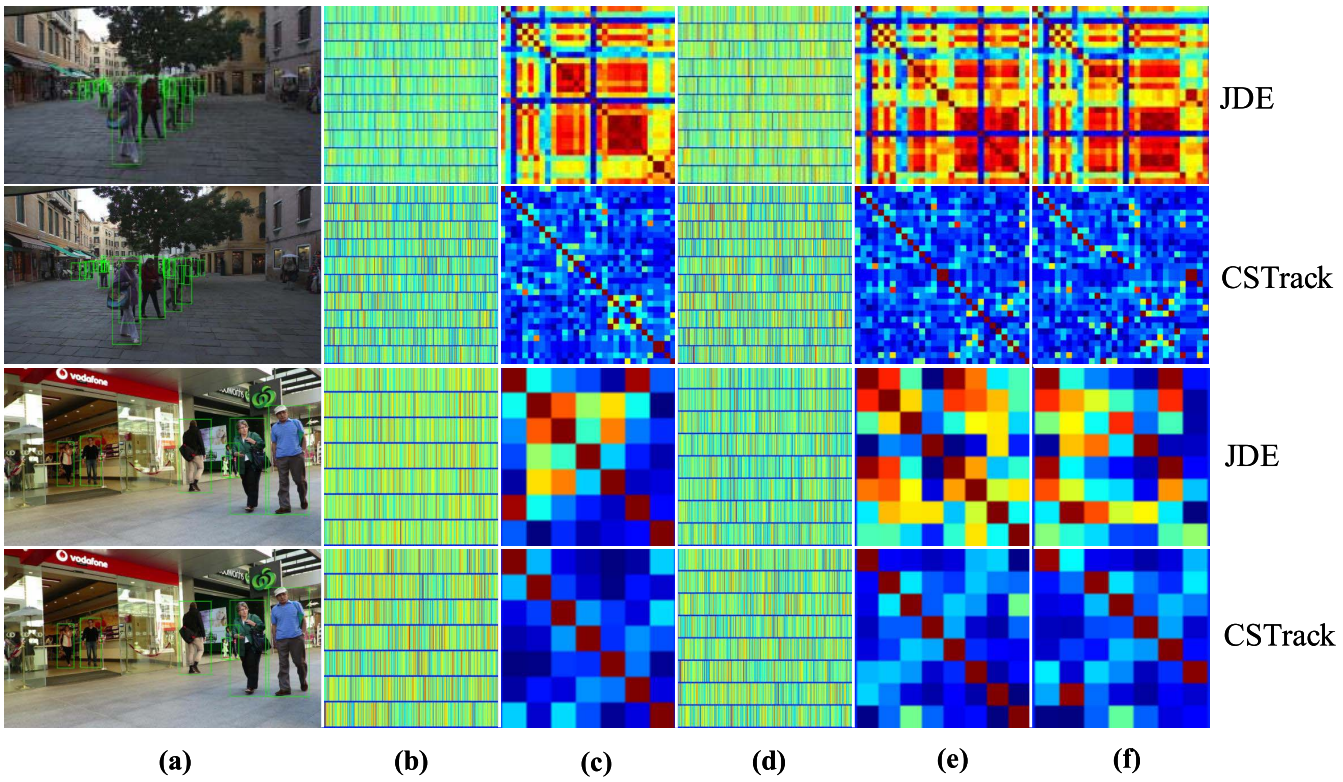


Fig. 6. Visualization about the discriminative ability of ID embeddings: (a) Example Image with ground truth detection results. (b) ID embeddings of the current example frame. (c) Cosine metric matrix between two ID embeddings of the current example frame. (d) Templates (*i.e.,* ID embeddings of the previous tracklets). (e) Cosine metric matrix between two ID embeddings of the template. (f) Cosine metric matrix between ID embeddings of the current example frame and the template. Red color indicates a higher correlation between two features. Best viewed in zoom.

TABLE VII
COMPARISON WITH STATE-OF-THE-ART TRACKERS ON THE HiEve BENCHMARKS (PRIVATE DETECTION)

Model	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	ID Sw.↓
DeepSORT [2]	27.1	28.5	8.4	41.4	5894	42668	3122
JDE [8]	33.1	36.0	15.1	24.1	9526	33327	3605
FairMOT [10]	35.0	46.6	16.2	44.1	6523	37750	2312
CenterTrack [25]	42.4	38.3	**23.9**	26.5	5802	**29766**	2940
NewTracker [46]	46.4	43.2	**26.3**	30.8	4667	30489	2133
CSTrack	**48.6**	**51.4**	20.4	33.5	**2366**	31933	**1475**

TABLE VIII
TO SHOW THE POTENTIAL OF CSTRACK. IDP AND IDR DESCRIBE THE PRECISION AND RECALL OF MATCHING TARGET TRACKLETS

Model	MOTA↑	IDF1↑	IDP↑	IDR↑	ID Sw.↓
JDE [8]	97.6	87.6	88.3	86.9	871
DeepSORT-2 [2]	**98.9**	95.6	95.9	95.3	**93**
CSTrack	**98.9**	**96.6**	**97.1**	**96.1**	162

methods can be divided into three categories. The first one is two-stage method including SORT_POI [12], POI [3], DeepSORT-2 [2], RAN [5], TAP [4], CNNMTT [42], IAT [41], STPP [43] and TPM [44]. The second one is the one-shot method including JDE [8] and FairMOTv2 [10]. The third one is the other joint detection and tracking method without using ReID for data association, including CTrackerV1 [22], TubeTK [23], Tracktor++ [24] and CenterTrack [25]. The evaluations of all above methods on these benchmarks are performed by the official online server .[3]

1) Two-Stage Methods: The comparison with two-stage methods follows the protocols in [33]. All the two-stage methods use the private detector (*i.e.,* FasterRCNN [14]), which is trained on a large private pedestrian detection dataset. The main differences among them lie in their ID embedding extraction and association strategies. As shown in Tab. VI, our proposed CSTrack outperforms the prior state-of-the-art methods by a significant margin and performs much faster.

[3]https://motchallenge.net

For example, comparing with the top performance two-stage tracker (*i.e.,* POI [3]), CSTrack outperforms it by +9.5% on MOTA and +8.2% on IDF1. Considering the inference speed, the proposed CSTrack runs faster than existing two-stage methods, *i.e.,* 16.4 FPS *vs.* 0.5∼8.6 FPS on MOT16 [33]. Furthermore, we construct a light version, namely CSTrack-S, which reduces convolution channels and layers of CSTrack to 30% of its original settings. For the lightweight vision, the running speed is clearly sped up to 34.6 FPS with only a small performance drop (MOTA −1.8∼2.4% and IDF1 −2.3∼3.6%). In the term of data association, it is not feasible to directly compare these methods because of the nonuniform object detectors. Therefore, we further analyze the upper bound of association under the same detections input in Sec. IV-D.

2) One-Shot Trackers: We present the evaluation results with the comparisons to recently-prevailing one-shot trackers on MOT16 [33], MOT17 [33] and MOT20 [37]. Our CSTrack achieves MOTA score of 75.6 and IDF1 score of 73.3 on MOT16, which evidently outperforms JDE [8] on MOTA of +11.2% and IDF1 of +17.5%. Besides, we compare our

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

Fig. 7. Qualitative results of our CSTrack on MOT17 [33]. Different colored bounding boxes denote different identity. The line under each bounding box indicates the tracklet of each target. The frame number is in the upper left of the figure. Best viewed in zoom.

model with the recently-proposed one-shot FairMOTv2 [10], which achieves the second performance of MOTA in all three benchmarks. FairMOTv2 [10] is trained on the same datasets as CSTrack, which makes the comparison fair. On MOT16 and MOT17, the false negative (FN) and false positive (FP) of our tracker surpass FairMOTv2 for 5.1%∼13.3% on FP and 2.1%∼2.7% on FN. On MOT20, our tracker outperforms FairMOTv2 by significantly decreasing FP from 103440 to 25404 and ID Sw. from 5243 to 3196. It means that our tracker can achieve a comparable or even superior IDF1 score when we use fewer boxes, which actually verifies that our method authentically improves the performance of data association.

3) Other Joint Detection and Tracking Methods: To further evaluate the proposed method, we conduct some comparisons

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

Fig. 8. Qualitative results of our CSTrack on MOT20 [37]. Different colored bounding boxes denote different identity. The line under each bounding box indicates the tracklet of each target. The frame number is in the upper left of the figure. Best viewed in zoom.

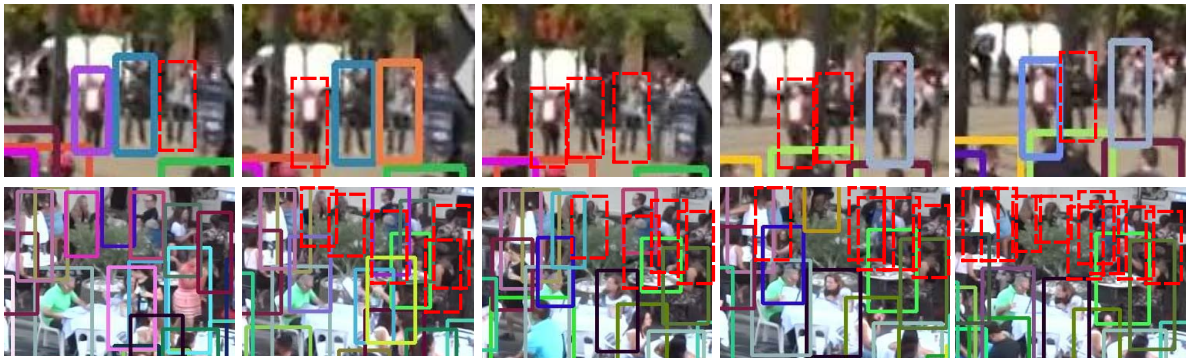


Fig. 9. Failure cases: the first row (a) illustrates image blur and the second row (b) illustrates over-occlusion. The red dotted boxes indicate false negatives.

with other joint detection and tracking methods, which adopt non-ReID methods for data association. The evaluation results on MOT16 and MOT17 are presented in Tab. VI. Compared with these methods, our tracker significantly improves tracking performances, especially on data association ability, *i.e.,* IDF1 +11.1%∼18.4% in MOT16 and +7.6%∼17.2% in MOT17. It verifies that our design effectively use ID information, which makes the data association ability of our tracker outperform the existing joint detection and tracking methods.

D. Further Analysis

1) HiEve Challenge: For further evaluating the strength of CSTrack, we show a comparison on a more challenging benchmark, *i.e.,* Human in Events dataset (HiEve) [46]. HiEve focuses on human-centric complex events, *e.g.,* fighting, quarreling, accident, robbery and shopping, including 19 videos for training and 13 videos for testing. For evaluating on the HiEve benchmark, we fine-tune our method on the training set of HiEve following the same training procedure on MOT challenge benchmarks (See Sec. IV-A). All evaluations are performed by the official online server ,[4] as shown in Tab. VII. The proposed CSTrack outperforms its baseline JDE [8] on MOTA for +16.5% and IDF1 for +15.4%. Compared with other top-ranked methods, our tracker still shows better tracking performance on MOTA and IDF1.

[4]http://humaninevents.org/

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

2) Upper-Bound Analysis: In this section, we study the upper-bound of the data association ability of our model. As a common wisdom, tracking performance is greatly influenced by the adopted object detectors [38]. To eliminate the influence of object detectors, we validate on the MOT16 training set by replacing the detection results with ground-truth bounding boxes. For a fair comparison, all of these methods are trained on the same training set, *i.e.,* MOT17 [33]. As shown in Tab. VIII, compared with JDE [8], our method yields substantial improvements on data association reflected by 9 points gains on IDF1 score. Moreover, the IDF1 score of our method surpasses the widely-used two-stage method DeepSORT-2 [2], which employs Wide Residual Network (WRN) [47] as ID embedding extraction model. It further verifies our design can effectively improve association ability of one-shot tracker.

3) Discriminative Ability of ID Embeddings: We visualize the discriminative ability of ID embeddings between JDE and our method. As shown in Fig. 6 (b) and (d), we observe that ID embeddings of JDE look more similar among targets than CSTrack, not only in the current example frame but also in the temple (*i.e.,* ID embeddings of the previous tracklets). For a more straightforward comparison, we construct the cosine metric matrix between ID embeddings of the current example frame in Fig. 6 (c) and the cosine metric matrix between the templates in Fig. 6 (e). Here, red color indicates a higher correlation between two ID embeddings, and blue indicates less correlated. The value on the diagonal indicates the similarity between the target and itself. The ideal cosine metric matrixes in Fig. 6 (c) and Fig. 6 (e) are ones with only the diagonals red and the rest blue. Compared with JDE [8], we observe that our method can learn more discriminative embeddings to avoid ambiguous matching between targets. It is also efficient for the update of the discriminative template. Moreover, we visualize the cosine metric matrix between ID embeddings of the current frame and previous tracklets template in Fig. 6 (f). During matching, we hope that each row and each column has at most one red color in ideal cosine metric matrix, and the rest are blue. As shown in Fig. 6 (f), we observe that our method can significantly improve the association ability. The advantage over JDE [8] is most pronounced on the robustness to occlusion and scale changes (see Fig. 6 (a)), which verifies that our proposed SAAN network is efficient to alleviate scale-aware competition and improve the resilience to objects with different scales.

4) Qualitative Results and Failure Cases: In this section, we show the qualitative results in MOT17 testing set [33] (see Fig. 7) and MOT20 testing set [37] (see Fig. 8). We observe that our method is efficient to deal with large-scale variations and keep correct identities (see MOT17-08 and MOT17-12). It mainly attributes to our SAAN module as it focuses on multi-resolution information to learn more discriminative embeddings. Moreover, we observe that our tracker performs well in many challenging scenes. For instance, from the results of MOT17-07 and MOT17-14, we can find that our tracker can predict accurate bounding boxes for small targets. In the visualization of MOT17-03, we find that our tracker performs robustly to occlusion objects, even in the presence of overcrowding (*e.g.,* the scenes in MOT20 testing set).

Despite the promising results, we also should be aware of the unaddressed cases. We discuss two scenarios in which CSTrack fails in Fig. 9. In particular, Fig. 9 (a) illustrates the negative influence of image blur, which makes our detector misclassify objects as background at some points. Fig. 9 (b) shows another failure circumstance, i.e., over-occlusion. As we can see, only a fraction of the targets have been detected over time, and the same local-visible objects will be missed by the detector. These cases are challenging for MOT because the missed targets will break the temporal consistency of tracklets. Despite being different in nature, these two cases arguably arise from the over-rely on image-based detection, which only focuses on single-frame features from the current frame. When a single-frame feature is not reliable, the tracker will miss the objects. In further work, we will further improve the one-shot tracker by employing temporal information.

V. Conclusion

In this paper, we propose a new one-shot online model CSTrack for the MOT task. A novel reciprocal network (REN) and scale-aware attention network (SAAN) are introduced to mitigate the competition and improve the collaboration of detection and ReID subtasks in MOT. The experimental results demonstrate the effectiveness and the efficiency of our framework. Compared with the methods on public benchmarks, our model achieves state-of-the-art performance, which outperforms our baseline JDE by +11.2% on MOTA and +17.5% on IDF1. Besides, CSTrack is a nearly real-time MOT tracker and its lightweight version can run at 34.6 FPS with only a minor performance drop, which is more suitable for real applications.

References

[1] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artif. Intell.*, vol. 293, Apr. 2021, Art. no. 103448.

[2] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[3] 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 *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands, Springer, 2016, pp. 36–42.

[4] Z. Zhou, J. Xing, M. Zhang, and W. Hu, "Online multi-target tracking with tensor-based high-order graph matching," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1809–1814.

[5] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 466–475.

[6] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, and J. Gu, "A strong baseline and batch normalization neck for deep person re-identification," *IEEE Trans. Multimedia*, vol. 22, no. 10, pp. 2597–2609, Oct. 2020.

[7] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3415–3424.

[8] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," 2019, *arXiv:1909.12605*.

[9] Z. Lu, V. Rathod, R. Votel, and J. Huang, "Retinatrack: Online single stage joint detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 14668–14678.

[10] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," 2020, *arXiv:2004.01888*.

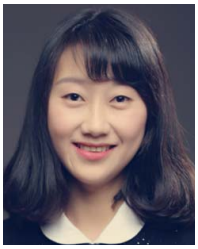
[11] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

[12] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.
[13] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.
[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[15] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2129–2137.
[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[17] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 3988–3998.
[18] G. Braso and L. Leal-Taixe, "Learning a neural solver for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6247–6257.
[19] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[21] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
[22] J. Peng *et al.*, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K., Springer, Aug. 2020, pp. 145–161.
[23] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "TubeTK: Adopting tubes to track multi-object in a one-step training model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6308–6318.
[24] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.
[25] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K., Springer, Aug. 2020, pp. 474–490.
[26] A. Vaswani *et al.*, "Attention is all you need," 2017, *arXiv:1706.03762*.
[27] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "PAD-Net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 675–684.
[28] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.
[29] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000, doi: 10.1609/aaai.v34i07.6999.
[30] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[31] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3213–3221.
[32] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 304–311.
[33] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.
[34] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, "Person re-identification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1367–1376.
[35] S. Shao *et al.*, "CrowdHuman: A benchmark for detecting human in a crowd," 2018, *arXiv:1805.00123*.
[36] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Springer, Sep. 2014, pp. 740–755.
[37] P. Dendorfer *et al.*, "MOT20: A benchmark for multi object tracking in crowded scenes," 2020, *arXiv:2003.09003*.
[38] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Feb. 2008.
[39] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands, Springer, Oct. 2016, pp. 17–35.
[40] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2953–2960.
[41] P. Chu, H. Fan, C. C. Tan, and H. Ling, "Online multi-object tracking with instance-aware tracker and dynamic model refreshment," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 161–170.
[42] N. Mahmoudi, S. M. Ahadi, and M. Rahmati, "Multi-target tracking using CNN-based features: CNNMTT," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 7077–7096, Mar. 2019.
[43] T. Wang *et al.*, "Spatio-temporal point process for multiple object tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 8, 2020, doi: 10.1109/TNNLS.2020.2997006.
[44] J. Peng *et al.*, "TPM: Multiple object tracking with tracklet-plane matching," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107480.
[45] B. Shuai *et al.*, "Application of multi-object tracking with Siamese track-RCNN to the human in events dataset," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 4625–4629.
[46] W. Lin *et al.*, "Human in events: A large-scale benchmark for human-centric video analysis in complex events," 2020, *arXiv:2005.04490*.
[47] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.

Chao Liang received the bachelor's degree from the School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China, in 2019. He is currently pursuing the M.S. degree with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu. His research interests include object detection, object tracking, and person re-identification.

Zhipeng Zhang received the bachelor's degree in optics and information engineering from the University of Electronic Science and Technology of China in 2017. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems with the Institute of Automation, Chinese Academy of Sciences. His current research interests include video understanding, including object tracking, object segmentation, and video representation learning.



Xue Zhou (Member, IEEE) received the B.S. degree in automatic control from the University of Electronic Science and Technology of China, Chengdu, China, in 2003, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008. She is currently a Professor with the School of Automation Engineering, University of Electronic Science and Technology of China. Her current research interests are video object perception, including video object segmentation, level-set-based object tracking, multiple object tracking, and person re-identification.

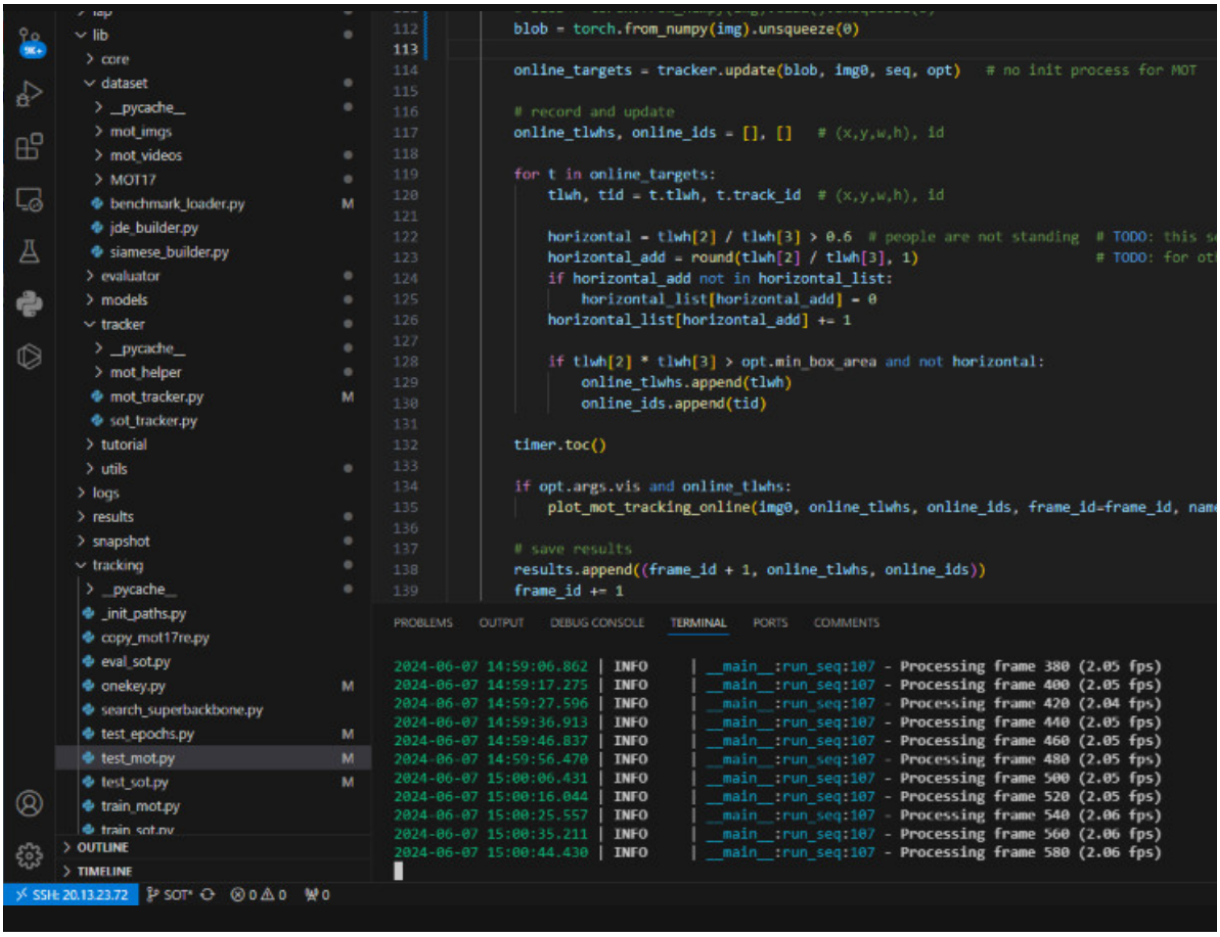
Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.

Bing Li received the Ph.D. degree from the Department of Computer Science and Engineering, Beijing Jiaotong University, Beijing, China, in 2009. He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include video understanding, color constancy, visual saliency, multiinstance learning, and web content security.

Weiming Hu (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Zhejiang University, Zhejiang, China, in 1998. From 1998 to 2000, he was a Postdoctoral Research Fellow with the Institute of Computer Science and Technology, Peking University, Beijing. He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing. His research interests are visual motion analysis, recognition of web objectionable information, and networks intrusion detection.

Shuyuan Zhu (Member, IEEE) received the Ph.D. degree from The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010. From 2010 to 2012, he worked with HKUST and Hong Kong Applied Science and Technology Research Institute Company Ltd. In 2013, he joined the University of Electronic Science and Technology of China, where he is currently a Professor with the School of Information and Communication Engineering. His research interests include image/video compression and image processing. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Current Date/Time:	Friday, June 7, 2024, 3:01:31 PM
Computer Name:	SP-DUCTQ
Operating System:	Windows 11 Pro 64-bit (10.0, Build 22000)
Language:	English (Regional Setting: English)
System Manufacturer:	Dell Inc.
System Model:	OptiPlex Tower 7010
BIOS:	1.7.1
Processor:	13th Gen Intel(R) Core(TM) i3-13100 (8 CPUs), ~3.4GHz
Memory:	12288MB RAM
Page file:	9526MB used, 4241MB available
DirectX Version:	DirectX 12



Authorized licensed use limited to: King Mongkuts Institute of Technology Ladkrabang provided by UniNet. Downloaded on May 30,2024 at 04:57:42 UTC from IEEE Xplore. Restrictions apply.