



Collaborative Multi-task Learning for Multi-Object Tracking and Segmentation

YIMING CUI*, University of Florida, USA

CHENG HAN* and DONGFANG LIU, Rochester Institute of Technology, USA

The advancement of computer vision has pushed visual analysis tasks from still images to the video domain. In recent years, video instance segmentation, which aims to track and segment multiple objects in video frames, has drawn much attention for its potential applications in various emerging areas such as autonomous driving, intelligent transportation, and smart retail. In this paper, we propose an effective framework for instance-level visual analysis on video frames, which can simultaneously conduct object detection, instance segmentation, and multi-object tracking. The core idea of our method is collaborative multi-task learning which is achieved by a novel structure, named associative connections among detection, segmentation, and tracking task heads in an end-to-end learnable CNN. These additional connections allow information propagation across multiple related task tasks, so as to benefit these tasks simultaneously. We evaluate the proposed method extensively on KITTI MOTS and MOTS Challenge datasets and obtain quite encouraging results.

CCS Concepts: • **Computing methodologies** → **Computer vision**; **Computer vision tasks**.

Additional Key Words and Phrases: Video instance segmentation, multi-object tracking, video object detection, associative connections, collaborative multi-task learning

1 INTRODUCTION

In the past decade, the computer vision community has achieved significant progress in many tasks with the development of deep learning [32, 39, 58, 62]. Among various visual tasks, instance segmentation [34] has drawn wide attention due to its importance in many emerging applications, such as autonomous driving [52, 55, 57, 59, 88, 95], augmented reality [9, 10], and video captioning [96, 97]. Technically, it is quite challenging as it is a compound task consisting of both object detection and segmentation, each of which is a difficult task and has been studied for a long time.

Compared to instance segmentation on images, multi-object tracking, and segmentation are much more challenging because it not only needs to perform instance-level segmentation on individual frames but also has to depict the coherence of each instance in consecutive video frames [3]. Due to these challenges, multi-object tracking and segmentation have received much attention in recent years [21, 56, 81, 94, 98, 99]. In general, multi-object tracking and segmentation contain object detection, segmentation, and tracking simultaneously in consecutive video frames. Compared to video object segmentation [83] that deals with object segmentation and tracking in videos, multi-object tracking and segmentation require additional object detection. It also has to generate object masks compared to video object detection which contains both object detection and tracking in videos [19, 22, 23, 82, 106, 107].

Authors' addresses: Yiming Cui, University of Florida, Gainesville, Florida, USA; Cheng Han; Dongfang Liu, Rochester Institute of Technology, Rochester, New York, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, or post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2833-0528/2023/11-ART \$15.00

<https://doi.org/10.1145/3632181>

Currently, the state-of-the-art methods are mainly based on Mask R-CNN [34] while adding tracking sub-network. The Mask R-CNN family [81, 94] has demonstrated an appealing performance on this task. However, these methods still have several drawbacks. For instance, TrackR-CNN [81] uses proposal-based ROI features, which are not fine enough for mask and tracking heads to produce accurate predictions. In detail, the current methods use the coarse proposal-based ROI features directly, which is not enough. Instead, our method processes the coarse proposal-based ROI features first and then uses the refined version for the downstream tasks. In addition, using proposal-based features not only has high computational complexity but also makes it prone to incorrect or redundant predictions. What is more, TrackR-CNN models object movements and scene consistency among video frames by using a 3D convolutional operation, which is also parameter-heavy and computationally expensive. Although PointTrack [94] achieves a significant improvement in segmentation results by proposing a more powerful mask head, it is a multi-step architecture and so cannot be jointly optimized. In PointTrack, detection, and segmentation operations are applied to the input first. This partial model is optimized first without the tracking task. Then the detected and segmented objects are transformed into point cloud formats for the tracking task, which is optimized. In general, the whole model contains multiple steps and is not optimized jointly or end-to-end trainable. Moreover, its point-cloud strategy requires additional post-processing. Meanwhile, although these methods are trained by multiple learning objectives corresponding to different tasks, the relations among these tasks have not been explored.

Intuitively, object detection could benefit instance segmentation, and good object masks are also helpful for multi-object tracking. Inspired by this intuition, we propose a novel idea of collaborative multi-task learning for multi-object tracking and segmentation. Associative connections are added among different task heads to enable information propagation across them. Although our method is also based on Mask R-CNN [34], it fundamentally differs from other Mask R-CNN variants [81, 94] as our method exploits associative connections to facilitate accurate information propagation across the detection, segmentation, and tracking heads to benefit individual tasks. With associative connections, our segmentation and tracking can perform predictions on refined ROI features instead of using features pooled from the very coarse region proposals. Extensive experimental results demonstrate significant improvements over the existing state-of-the-art methods for multi-object tracking and segmentation on two benchmarks (KITTI MOTS [81] and MOTS Challenge [64]). The principal contributions of this work can be summarized as follows,

- We propose a novel idea of collaborative multi-task learning for multiple object tracking and segmentation in videos.
- We design associative connections among different tasks to enable information flow through different task heads, to simultaneously learn from multiple tasks and consider their intrinsic relations in the meantime.
- Compared to existing methods, our method achieves better results on KITTI MOTS and MOTS Challenge benchmarks, especially on the consistency of multi-object tracking.

2 RELATED WORK

Visual tasks in the video domain have been under-explored in the literature compared to image-level tasks. Particularly, video instance segmentation which includes detection, segmentation, and tracking is largely ignored due to its extreme difficulties. This section offers a review of recent works about several related tasks to video instance segmentation.

2.1 Image Object Detection

State-of-the-art object detection methods [25, 53, 72, 103?] are generally based on deep CNNs for feature extraction and a shallow detection structure for detection prediction, including classification and bounding box regression. R-CNN [31] proposed a multi-stage pipeline to classify region proposals at different semantic levels

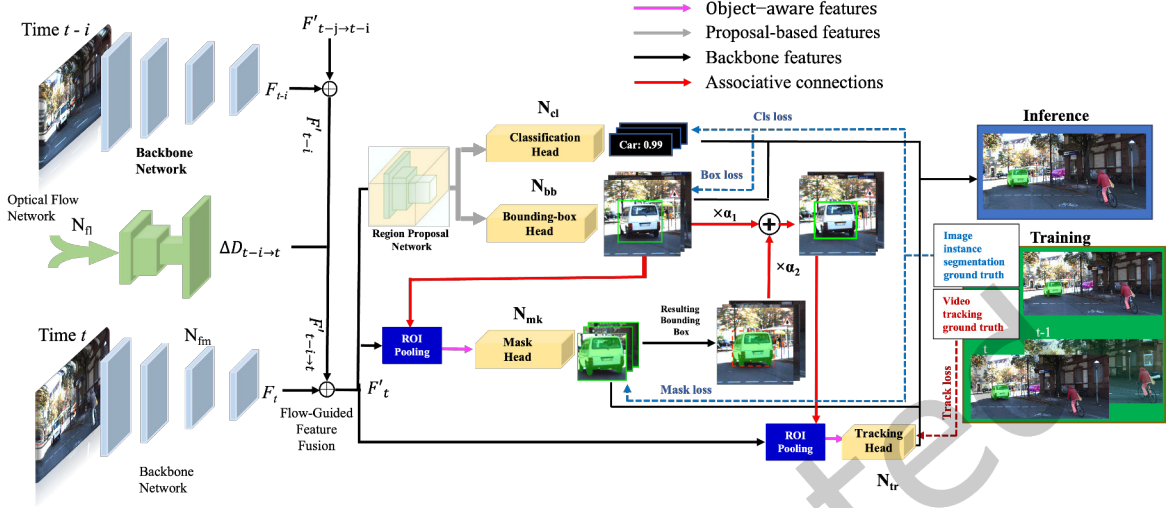


Fig. 1. Illustration of the proposed framework. It adds associative connections among different task heads in TrackR-CNN [81] to enable collaboration among multiple tasks by facilitating information propagation across them. Particularly, the bounding boxes outputted by the detection head are linked to the mask head so as to extract object-aware features for instance segmentation. The outputs of the mask head are fused with the outputs of the detection head to obtain more reliable bounding boxes, which are then inputted into the tracking head for better tracking ability. Besides, a lightweight optical flow network is used to model the object movements across successive frames, which is used for aligning object features among these frames to obtain more powerful feature maps in the currently processed frame.

for object detection. To speed up, Fast R-CNN [30] and Libra R-CNN [68] used ROI pooling on the feature maps which are shared on the entire image. As a representative work of the multi-stage detection family, Faster R-CNN [72] introduced a Region Proposal Network (RPN) to generate region proposals and then the proposal-based features are shared between classification and bounding box regression heads. R-FCN [24] replaced ROI pooling with position-sensitivity ROI pooling to further improve the recognition accuracy while still facilitating feature sharing.

Traditional two-stage object detectors, exemplified by the R-CNN family [30, 72, 73], rely on a plethora of predefined anchor boxes to designate initial object candidate locations. In contrast, one-stage methods [4, 70] were introduced to enhance the efficiency and inference speed of object detectors by forgoing the use of region proposals. Recently, query-based approaches [11, 75, 105] have emerged, replacing anchor boxes and region proposals with learned proposals or queries. DETR [11] adapts an encoder-decoder architecture based on transformers [80] to generate a sequence of prediction outputs. It introduces a set loss function to facilitate bipartite matching between predicted and ground-truth objects. Deformable-DETR [105] enhances the convergence of DETR by refining feature spatial resolutions. Sparse R-CNN [75] employs a fixed sparse set of learned object proposals to classify and localize objects in the image. It utilizes dynamic heads to generate final predictions directly, eliminating the need for post-processing techniques like non-maximum suppression.

2.2 Image Instance Segmentation

Instance segmentation not only predicts semantic classes on each pixel but also groups pixels into different object instances. Due to the effectiveness of R-CNN[31], many instance segmentation methods perform mask prediction on top of region proposals. Some early methods are based on the bottom-up segment strategy. For instance, the DeepMask family [12, 99] learns to segment proposal candidates first and then classify them using Fast R-CNN. These methods have segmentation precede detection, which is slow and inaccurate.

Another strategy for instance segmentation is based on a parallel prediction of masks and class labels [34]. Li et al. [49] proposed the fully convolutional instance segmentation by combining the segmentation proposal system in [12] and object detection system in R-FCN [24]. Basically, [12, 49] all use a set of position-sensitive output channels to simultaneously predict object classes, bounding boxes, and masks, thus being fast and fundamentally similar. However, they struggle to deal with occlusion or truncation instances as they tend to create spurious edges in those cases.

Besides two-stage methods, one-stage instance segmentation frameworks like YOLACT [5, 6], SipMask [7], and SOLO [86, 87] have been introduced to strike a balance between inference speed and accuracy. Recently, QueryInst [28] extended the query-based object detection method Sparse R-CNN [75] to the instance segmentation task by incorporating a dynamic mask head and parallel supervision. Nevertheless, all the two-stage and query-based methods mentioned earlier employ a fixed number of proposals, which may not be adaptable to images with varying objects or to devices with different computational resource constraints.

2.3 Object Tracking

Tracking-by-detection is a popular strategy for multi-object tracking [17, 60, 85]. A common practice is to associate the tracks with detection based on their confidences [17]. To improve the tracking accuracy, Sun et al. [76] exploited multiple detectors by considering outputs from multiple over-detected detectors. However, this method enhances tracking performance at the cost of high computational complexity. More recently, Kim et al. [47] proposed to use a single object tracker based on a binary classifier for online multi-object tracking. Their proposed architecture has shared features for tracking and classification to speed up the process. Even though, [47] is still computationally expensive for real-time applications.

Many previous methods deal with tracking tasks as a global optimization problem [77, 99]. This kind of method formulates temporal information from nearby frames to reduce noisy detection and handle ambiguities for object association. The principal strategy is to re-identify objects using the embedding vectors since each association vector represents the identity of an object [77, 81]. Inspired by the aforementioned methods, our work also leverages deeply learned ReID features. With the proposed associative connections, our work has better object-aware features to obtain improved identification performance.

2.4 Video Object Detection

Video object detection involves the identification and localization of objects of interest in each frame, even in the presence of potential degradation due to rapid motion. Present approaches [14, 19, 20, 22, 33, 41, 42, 84, 91, 104, 106] typically extend image-based object detectors into the realm of videos. These models fall into two categories: Post-processing-based and feature-aggregation-based.

Post-processing-based models extend image object detectors to video by linking prediction results across frames based on temporal relationships [33, 41, 42]. Examples include T-CNN [42] and Seq-NMS [33]. T-CNN employs a CNN-based pipeline with straightforward object tracking for video object detection. Seq-NMS associates prediction results from each frame using the IOU threshold. While these models outperform single-image object detectors, they heavily rely on individual frame detections and lack joint optimization. If single-frame outputs are

erroneous, the post-processing pipeline cannot rectify them, resulting in suboptimal performance. Furthermore, these models tend to be slower as they process each frame independently before post-processing.

In contrast, feature-aggregation-based and Transformer-based models can aggregate information across frames and jointly optimize predictions, yielding improved performance. These models effectively utilize temporal and spatial cues to track and detect objects across consecutive frames, making them better suited for video object detection tasks. While post-processing-based models offer some improvement over single-image object detectors, they are constrained by their reliance on individual frame results and slower inference. Feature-aggregation-based and Transformer-based models present a more promising approach, leveraging data from multiple frames and optimizing jointly.

Feature-aggregation-based models enhance current frame representations by incorporating features from adjacent frames, assuming they can mitigate feature degradation. Several models embody this concept. For instance, FGFA [106] employs estimated optical flow to fuse neighboring features, while MANet combines pixel-level and instance-level object features. Conversely, SELSA [91] calibrates features based on semantic similarity rather than temporal relations. MEGA [14] integrates local and global temporal information to enhance performance. Although these models surpass post-processing-based ones in performance, they typically demand more computational resources, resulting in slower inference speeds. In summary, while feature-aggregation-based models have enhanced video object detection, they often come at the expense of slower inference. Transformer-based models offer a promising solution to this issue by efficiently fusing information across frames, suggesting further advancements in video object detection tasks.

2.5 Video Instance Segmentation

Recent endeavors on video instance segmentation [81, 94, 99] are intuitive extensions of Mask R-CNN [34] while considering additional motion cues [16] or temporal consistency [27, 69, 100] in videos. There are three main categories of existing methods for Visual Instance Segmentation (VIS): two-stage, one-stage, and transformer-based. Two-stage approaches [3] build upon the Mask R-CNN family [34, 72], incorporating an additional tracking branch for object association. One-stage methods [8, 56] employ anchor-free detectors [79], often utilizing linear mask basis combination [5] or conditional mask prediction generation [78]. Transformer-based models [15, 37, 89, 101] introduce innovative adaptations of the transformer architecture for VIS tasks. VisTr [89] pioneers the application of transformers in VIS, and IFC [40] enhances efficiency through the use of memory tokens. Seqformer [92] introduces frame query decomposition, while Mask2Former [15] incorporates masked attention. VMT [44] extends the Mask Transfuser [43] to video for high-quality VIS, and IDOL [93] specializes in online VIS.

2.6 Multi-object Tracking and Segmentation

Multi-object tracking (MOT) is a crucial task in autonomous driving, encompassing both object detection and tracking within video sequences. Numerous datasets have been curated with a focus on driving scenarios, including KITTI tracking [29], MOTChallenge [64], UA-DETRAC [90], PathTrack [63], and PoseTrack [1]. However, none of these datasets offer segmentation masks for annotated objects, thus lacking pixel-level representations and intricate interactions seen in MOTS data. More advanced datasets, such as Cityscapes [18], ApolloScape [38], BDD100K [102], and KITTI MOTS dataset [81], do provide instance segmentation data for autonomous driving. Nevertheless, Cityscapes only supplies instance annotations for a small subset (i.e., 5,000 images), and ApolloScape does not offer temporal object descriptions over time. Consequently, neither dataset is suitable for the joint training of MOTS algorithms. In contrast, KITTI MOTS [81] stands as the first public dataset that addresses the scarcity of data for the MOTS task, albeit with a relatively modest number of training samples. To date, BDD100K boasts the largest scale of data from intensive sequential frames, which may be considered redundant for training

purposes. In comparison to the aforementioned datasets, our DGL-MOTS dataset encompasses a wider range of diverse data with finely detailed annotations.

3 THE PROPOSED METHOD

3.1 Overview

Fig. 1 shows the proposed network for multi-objects tracking and segmentation. It follows the basic network architectures for object detection, instance segmentation, and object tracking, containing two parts: one for extracting feature maps and the other for different tasks sharing the extracted features as inputs. More specifically, the first part of our method consists of two components: a backbone feature extraction network \mathbf{N}_{fm} to compute per frame feature maps and an optical flow network \mathbf{N}_{fl} to estimate object movements across nearby frames. With the help of optical flow, feature maps from previous frames can be warped into the current frame, which is used to combine with the extracted feature maps at the current frame, to obtain an enhanced feature representation of the current frame that should be more robust to image blur, occlusion, etc. Accordingly, the second part of our method is constituted of three heads, each of which corresponds to a specific task, i.e., object detection, instance segmentation, and multi-object tracking. The detection head contains classification head \mathbf{N}_{cl} and bounding-box regression head \mathbf{N}_{bb} while the instance segmentation is achieved by the mask prediction head \mathbf{N}_{mk} . The tracking head \mathbf{N}_{tr} aims to identify the same objects that appeared in multiple frames.

To train such a network with different heads end-to-end, existing methods resort to multi-task learning that simultaneously optimizes losses related to individual tasks. However, those methods ignore the intrinsic correlations among these tasks, which could benefit each other if used properly. In detail, each loss is only designed and optimized specifically for one task and there is no interaction between different tasks and their corresponding losses. In other words, the performance of the instance segmentation task will not affect that of the object detection task and vice versa. In this case, we argue that the designs of losses can be optimized. Therefore, we propose collaborative multi-task learning to enable information propagation among these individual tasks when optimizing the total learning objective containing all these tasks. For this purpose, we introduce associative connections among detection, segmentation, and tracking heads so that the network training could be aware of the interactions among these three tasks. The introduced associative connections are shown by the red arrows in Fig. 1. Besides enabling information propagation among different task heads, our network is more efficient compared to previous methods. Existing MOTS methods [50, 61, 81] that also use three task heads encounter the problem of computing redundant feature representations, as they rely on proposal features for instance segmentation and tracking. With the help of associative connections, our segmentation and tracking heads only take the detected bounding boxes or masks for input, thus avoiding computing features on unrelated proposals. In addition, due to the added associative connections, the mask head and tracking head can use more accurate features extracted on object bounding boxes instead of the region proposals. Therefore, higher-quality instance segmentation and tracking results can be expected.

In the following subsections, we describe in detail the proposed method, including feature extraction network, associative connections, training, and inference, as well as the network architecture. Table 1 lists the main symbols used in this paper for the neatness of description.

3.2 Network Architecture

Our network architectures have general and flexible design options. We craft state-of-the-art architectures into the proposed methods for different visual tasks. Particularly, the proposed method has three contributing modules: (i) the CNN backbone architecture employed for feature extraction over the input frame, (ii) the optical flow architecture used for motion estimation across frames, and (iii) the task heads for classification, location regression

Table 1. Summarization of the main notations used in this paper.

Notation	
\mathbf{N}_{fm}	Feature extraction network
\mathbf{N}_{fl}	Optical flow network
\mathbf{N}_{cl}	Classification head
\mathbf{N}_{bb}	Bounding box head
\mathbf{N}_{mk}	Mask head
\mathbf{N}_{tr}	Tracking head
F_t	Feature maps at time t
F'_t	Enhanced feature maps at time t
I_t	Input frame at time t
$F_{t-i \rightarrow t}$	Warped feature maps from time $t - i$ to t
$\Delta D_{t-i \rightarrow t}$	Object movement between t and $t - i$
$\omega_{k \rightarrow t}$	Weight for fusing feature of the k th frame at the time t
b^i	The bounding box of the i^{th} object
b_{mk}^i	The bounding box computed from the predicted mask of the i^{th} object
b_{wb}^i	Weighted bounding box of the i^{th} object connecting the detection and segmentation heads to the tracking head
\mathcal{L}_{cls}	Loss for classification
\mathcal{L}_{box}	Loss for bounding box regression
\mathcal{L}_{mask}	Loss for mask generation
\mathcal{L}_{track}	Loss for tracking
\mathcal{L}_{total}	Total loss for training

(bounding box), mask generation, and object tracking. With associative connections, the mask head and tracking head are applied to object-aware RoI instead of using RoI from proposals.

3.2.1 Feature extraction. The ResNet-101 [35] is used as our backbone feature extraction network \mathbf{N}_{fm} in this paper. According to the practice of using ResNet-101 as the backbone in Faster R-CNN, the outputs of its final convolutional layer C4 are feedforwarded to the task heads.

3.2.2 Motion estimation. The simple version of FlowNet [26] is used as the flow network \mathbf{N}_{fl} for motion estimation across video frames. It is pretrained on the synthetic Flying Chairs dataset [26]. According to [106, 107], we have the input frame half-sized and the output stride 4. Therefore, the output resolution of the generated flow field is 1/8 of the original frame size. Since the output of the feature extraction network has a stride of 16, we use bilinear interpolation to further down-sample the flow field and scale the field by half, to match the resolution of extracted feature maps. The down-sample process is non-learnable as the bilinear interpolation is a parameter-free layer in the network and is also differentiated during training.

3.2.3 Task heads. For the task heads, we follow architectures presented in Faster R-CNN [72], Mask RCNN [34], and TrackR-CNN [81] for different tasks. For detection, we craft Faster R-CNN, a two-stage detector for object classification and bounding box regression. For mask prediction, we follow Mask RCNN by adding a fully convolutional mask prediction branch. For the tracking task, we include an association layer [81] to calculate the distance of 128-D identity vectors to track different objects across frames. Among these heads, there are

associative connections to collect object-aware features and propagate them across these tasks. Introducing associative connections among different task heads to facilitate information flow interactively among several tasks is also the main contribution of this work. Compared to previous proposal-based features, this work uses object-aware features by leveraging associative connections, which shows better performance in predicting masks and tracking the same identities.

3.3 Feature Extraction Guided by Optical Flow

Given an input frame I_t at time t , the feature extraction process can be expressed as $F_t = \mathbf{N}_{fm}(I_t)$. Note that F_t are only intermediate feature maps, which will be enhanced with features from previous frames based on the estimated object movements by \mathbf{N}_{fl} . It is the enhanced feature maps being passed into the following heads regarding different tasks. Although various backbone networks can be used here for feature extraction, we use ResNet-101 [35] in this paper due to its popularity.

To enhance the feature representation of different objects on the input frame by leveraging on its previous frames, we exploit the temporal visual cues with the help of an optical flow network \mathbf{N}_{fl} . Thus, the enhanced feature representation is also called flow-guided features. The warped feature from time $t - i$ to t is denote as:

$$F_{t-i \rightarrow t} = \mathcal{WP}(F_{t-i}, \Delta D_{t-i \rightarrow t}), \quad (1)$$

where \mathcal{WP} is the feature warping function to predict the feature maps at t based on the feature maps at $t - i$ (i.e., F_{t-i}) and the estimated object movement $\Delta D_{t-i \rightarrow t}$ between t and $t - i$ frames. The movement is predicted by the optical flow network, i.e., $\Delta D_{t-i \rightarrow t} = \mathbf{N}_{fl}(I_{t-i}, I_t)$. By modeling the feature map movements from nearby frames, we hope to improve the extracted features that may have been originally compromised by motion blur, defocus, or occlusion that often happened in the video domain.

Since in real scenarios processing a specific video frame can only rely on its previous frames, given an input frame I_t , we obtain a set of warped feature maps for each of its previous frames to compute the enhanced feature maps of the current frame I_t . Specifically, with a predefined temporal range n , each feature map of the previous frames in this range is warped into frame t according to Eq. (1), resulting in a set of predicted feature maps $\{F_{t-i \rightarrow t} | i \in [1, n]\}$. Note that we warp previous features to every current frame for feature fusion instead of keyframes or using dense aggregation [106, 107]. Due to the high efficiency of the used flow network (i.e., FlowNet [26]), this enables our method to extract strong features but still with affordable computational cost. Runtime analysis is provided in Section 3.5.2.

Given the set of warped feature maps $\{F_{t-i \rightarrow t} | i \in [1, n]\}$, the fused feature maps F'_t at the frame I_t is then computed by the weighted average of these warped features and F_t ,

$$F'_t = \sum_{k \in [t-n, t-1]} (\omega_{k \rightarrow t} \cdot F'_{k \rightarrow t}) + F_t, \quad (2)$$

where the weight $\omega_{k \rightarrow t}$ is adaptively computed based on the similarity between $F_{k \rightarrow t}$ and F_t . Among many choices for defining the similarity between these feature maps, we follow the implementation in [54] to use a shallow fully convolutional network to output embedding vectors for similarity computation. That is, $\omega_{k \rightarrow t}$ is computed as,

$$\omega_{k \rightarrow t} = \exp\left(\frac{F_{k \rightarrow t}^e \cdot F_t^e}{|F_{k \rightarrow t}^e| |F_t^e|}\right), \quad (3)$$

where F^e is the embedding vector of F outputted by the shallow fully convolutional network. Note that the dot productions in Eq. (2) and Eq. (3) is element-wise multiplication and all the obtained weights are normalized so that $\sum_{k \in [t-1, t-n]} \omega_{k \rightarrow t} = 1$.

Compared to the existing architectures like TrackR-CNN [81] which naively uses heavy 3D convolutions for feature extraction and fusion, our method models the spatial movements of objects in successive video frames in

a more reasonable fashion to enhance the extracted features, could improve the accuracy for upper-stream tasks. With the help of a lightweight optical flow network (i.e., FlowNet [26]) to predict the object movements, our method for feature extraction has much fewer parameters than 3D convolutions. Therefore, our method keeps a faster runtime than TrackR-CNN and its variants. In addition, by tuning the temporal range n , we can actively control the tradeoff between inferring speed and accuracy.

3.4 Associative Connections Across Tasks

After the above-mentioned flow-guided feature fusion, the obtained feature representation could be significantly enhanced. These enhanced feature maps are then fed into the individual upper-stream task heads. These tasks include object detection, instance segmentation, and object tracking. Although these tasks are intrinsically related, existing methods (e.g., TrackR-CNN [81] and CAMOT [67]) simply add different learning objectives together. This paper explores the relations among these tasks for improving multiple object tracking and segmentation accuracy. Under this motivation, we propose collaborative multi-task learning by adding associative connections among different tasks to facilitate information flow through other heads. Different from previous methods where different task heads are independent, the three heads in our method jointly worked together. To be concrete, as shown in Fig. 1, there are three associative connections across the detection, segmentation, and tracking heads. The first one is the associative connection between the output of bounding box regression in the detection head and the input of the mask prediction head. The second one is a connection to link the outputs of detection and mask heads. The third one is the associative connection between the combined outputs of detection and mask heads and the input of the tracking head. With this implementation, we achieve much lower computational complexity per instance than the mask head in TrackR-CNN, where the ROI-based operations are repeatedly performed for final dense predictions. For comparison, the associative connections facilitate information sharing across different task heads to keep a low computational cost. Meanwhile, owing to the interleaved information flow among all three heads, we can jointly optimize their objectives as a whole, while previous works optimize each task's objective independently since all the task heads are independent in their methods.

3.4.1 Connection from Detection to Segmentation. Due to the good performance of Mask R-CNN [34] for instance segmentation, we adopt its head architecture for object detection and segmentation. The detection and instance segmentation are based on region proposals, and our network training procedure contains two stages Mask R-CNN and Faster R-CNN [72]. In the first stage, a Region Proposal Network (RPN) takes the extracted features from the backbone and outputs a set of region proposals for object detection. In the second stage, for the detection head, an ROI pooling layer is used to generate region features for each proposal, which are then used to predict the object class and the related bounding box by the classification \mathbf{N}_{cl} and regression \mathbf{N}_{bb} heads respectively. Then, for the mask head, an ROI pooling layer is used to generate object-aware features based on the bounding boxes outputted from the detection head. These object-aware features are used by the following layers to conduct instance segmentation for producing object masks.

Adding a connection from the output of the detection head to the ROI pooling layer of the mask head (i.e., the red arrow between the bounding box and mask heads in Fig. 1) can explore more accurate features focused on objects compared to using the region proposals that have been widely used in the literature [81, 94]. This is because the output of the detection head is the object bounding boxes that are finer than the region proposals regarding the object locations. Therefore, our mask head can accurately fire on the pixels of instances. In other words, by adding an associative connection between the output of the detection head and the input of the mask head, we achieve a better behavior for mask generation since the used features could be more focused on the object itself.

The learning objective of our detection head is identical to that of Faster R-CNN,

$$\mathcal{L}_{cls} + \mathcal{L}_{box} = \frac{1}{D} \sum_{d \in D} \mathcal{L}'(p_d, p_d^*) + \frac{1}{D} \sum_{d \in D} \mathcal{L}''(t_d, t_d^*), \quad (4)$$

where D is the total number of detections in a video sequence, p_d is the predicted probability of an object, p_d^* is the corresponding ground-truth label, t_d is the coordinates of the predicted bounding box, t_d^* is the ground-truth for the corresponding bounding box.

Similarly, the instance segmentation head is trained using the following loss function,

$$\mathcal{L}_{mask} = \frac{1}{D} \sum_{d \in D} \mathcal{L}'''(m_d, m_d^*), \quad (5)$$

where m_d and m_d^* are predicted mask and the ground-truth mask.

Note that the novelty of our method lies in introducing the associative connections to interact among different tasks, not the individual learning objective for each task.

3.4.2 Connection from Detection and Segmentation to Tracking. The tracking head aims to establish correspondences of the same identities across successive frames. Existing works, such as TrackR-CNN, extensively extract ROI features for all region proposals, which are then used to compute identity vectors. Those identity vectors from positive proposals are used to link the same identities across frames based on their similarities. Such a tracking head computes many redundant proposal features and identity vectors. In addition, extracting ROI features from region proposals may not be accurate enough to facilitate good tracking results. To address these issues, we propose to extract ROI features from the detection and instance segmentation results. This not only avoids the redundant feature computation but also provides a more accurate description of the tracked objects. Such an idea inspires the associative connection from the outputs of detection and segmentation heads to the input of the tracking head.

Usually, the bounding boxes generated by the detection head are larger than the ground truth boxes, while the bounding boxes computed from the masks produced by the mask head are often tighter than the ground truth ones due to the pixel-level prediction of the mask head. To effectively combine two kinds of bounding boxes for a better ROI feature extraction in the tracking head, we propose an adaptive fusion strategy to obtain a weighted bounding box as,

$$b_{wb}^i = \alpha_1 \cdot b^i + \alpha_2 \cdot b_{mk}^i, \quad (6)$$

where b^i is the bounding box outputted by the detection head and b_{mk}^i is the bounding box drawn from the predicted mask for the i^{th} object, α_1, α_2 are adaptive weights fulfilling $\alpha_1 + \alpha_2 = 1$. The parameters α_1 and α_2 are hyperparameters but they do not vary for each scenario right now. We will investigate how to adjust them for each video sequence in future work.

In this way, the weighted bounding boxes b_{wb} will be smaller than the bounding boxes b generated by the detection head, while larger than the mask-based bounding boxes b_{mk} . If the object is on a large scale, we expect the weighted bounding box to be tighter for tracking. However, if the object is small, the mask-based bounding box may not be accurate enough so we expect the weighted bounding box to approach the detected bounding box to include more information. Under this consideration, α_1 is set proportional to the reciprocal of the scale of the object, to be specific, the area of bounding box b^i . Therefore, we have a small α_1 (α_2 is large) and b_{mk}^i has a larger impact on the weighted bounding box b_{wb}^i when the object is large. On the contrary, b^i will have a larger impact on b_{wb}^i when the object is small.

Owing to the good properties of these adaptive bounding boxes, we add an associative connection to link these boxes into the tracking head for ROI feature extraction. In other words, we use b_{wb} to pool feature maps to be more object-aware. Following the idea of [77, 81] that use the embedding vectors to re-identify persons, our

tracking head uses one fully connected layer to map ROI features into identity vectors v , each of which indicates a unique identified instance. These identity vectors v are used to link all detections across frames. Since our ROI features are extracted from object-aware feature maps and could be more discriminative for tracking multiple objects, our tracking head can predict more distinguishable identity vectors.

In the training stage, we minimize the distances of all vectors belonging to the same object while maximizing the distances of vectors belonging to different objects. To this end, the tracking head is trained with the batch hard triplet ranking loss. For each detected object, we sample its hard positive detections and negative detections for network training. Let us denote D as all the detections in a video. At frame, t , each detection $d \in D$ is associated with a tracking vector v_d and a ground truth track ID which is used to determine its overlap with the ground truth over frames. Thus, for a video sequence including T frames, the tracking loss is computed by,

$$\mathcal{L}_{track} = \frac{1}{D} \sum_{d \in D} \max \left(m + \max_{\substack{n \in D \\ id_d \neq id_n}} \mathcal{S}(v_d, v_n) - \min_{\substack{p \in D \\ id_d = id_p}} \mathcal{S}(v_d, v_p), 0 \right), \quad (7)$$

where the subscript p and n indicate the positive and negative detections respectively and \mathcal{S} represents the similarity between the input vectors. By default, we use cosine similarity.

3.5 Inference and Training

In this section, we first elaborate on the inference algorithm, then analyze the runtime complexity, and finally discuss the training objective of the proposed method.

3.5.1 Inference algorithm. We summarize the inference of our proposed method in Algorithm 1. The algorithm of our proposed method can be divided into two major stages. In the first stage, given a video frame I_t , the feature extraction network \mathbf{N}_{fm} takes it as input and produces a set of intermediate feature maps F_t (line 2 in Algorithm 1). Based on the light-weight optical flow network \mathbf{N}_{fl} , we warp the temporal feature maps in previous frames to the current frame (line 3 to 4) according to Eq. (1). In the meantime, we calculate the similarities of the warped feature maps to that of the current frame. These similarities are used as adaptive weights (line 5) to perform flow-guided feature fusion over the temporally aligned features, to enhance the feature representation for the current frame. In this way, the enhanced feature maps F'_t (lines 6 to 7) can be obtained for different head tasks. In the second stage, the enhanced feature maps F'_t are fed into the classification and bounding box heads to predict object class $\{c_t\}$ and bounding box $\{b_t\}$ respectively (line 8). Using the predicted bounding box from $\{b_t\}$, we further obtain the object-aware feature maps $F'_{(bb)_t}$ which are more accurate than the proposal-based features that are widely used in Mask RCNN [34] and its variants [81, 94]. Next, the feature maps $F'_{(bb)_t}$ are fed into the mask head \mathbf{N}_{mk} (line 11). Since the mask head \mathbf{N}_{mk} takes the object-aware feature maps as inputs, it tends to generate masks (denoted as $\{m_t\}$) with higher quality compared to those masks predicted from proposal based features. We draw a close rectangular over each instance in $\{m_t\}$ to obtain a mask-based bounding box $\{b_{(mk)_t}\}$. Considering the predictions from both $\{b_t\}$ and $\{b_{(mk)_t}\}$ together, we compute adaptive weights to get a weighted bounding box $\{b_{(wk)_t}\}$ for each detected object (line 12) which could be more accurate than the original bounding boxes $\{b_t\}$ produced by the bounding box head. Intuitively, more accurate bounding boxes can benefit following object tracking heads as they only focus on the major component of objects, reducing the influence of background. The final video recognition result R_t consists of a set of object classes $\{c_t\}$, bounding-box locations $\{b_t\}$, object masks $\{m_t\}$, and tracked identity labels $\{t_t\}$.

3.5.2 Time complexity. Based on Algorithm 1, we give an analysis of the time complexity of the proposed method. Besides the backbone feature extraction network \mathbf{N}_{fm} , there are five modules: (1) The optical flow network \mathbf{N}_{fl} which includes the bilinear warping and feature embedding functions; (2) The classification head \mathbf{N}_{cl} ; (3) The bounding-box regression head \mathbf{N}_{bb} ; (4) The mask head \mathbf{N}_{mk} ; (5) The tracking head \mathbf{N}_{tr} . Since our associative

Algorithm 1 Online inference of the proposed method per frame

1: input: frame $\{I_t\}$	◀ Video frame at time t
2: $F_t = \mathbf{N}_{fm}(I_t)$	◀ Produce feature maps
3: for $j = t - n$ to $t - 1$ do	◀ Use temporal features
4: $F_{j \rightarrow t} = \mathcal{WP}(F_j, \mathbf{N}_{fl}(I_j, I_t))$	◀ Feature warping to time t
5: $\omega_{j \rightarrow t} = \exp\left(\frac{F_{j \rightarrow t}^e \cdot F_t^e}{ F_{j \rightarrow t}^e F_t^e }\right)$	◀ Calculate adaptive weights
6: end for	
7: $F'_t = \sum_{j=t-n}^{t-1} (\omega_{j \rightarrow t} \cdot F_{j \rightarrow t}) + F_t$	◀ Flow-guided feature fusion
8: $\{c_t\} = \mathbf{N}_{cl}(F'_t)$	◀ Object detection results
9: $\{b_t\} = \mathbf{N}_{bb}(F'_t)$	
9: $F'_{(bb)t} = \{b_t\} \rightarrow F'_t$	◀ bounding-box features maps
10: $\{m_t\} = \mathbf{N}_{mk}(F'_{(bb)t})$	◀ Produce mask results
11: $b_{(mk)t} \leftarrow \{m_t\}$	◀ Mask-based bounding box
12: $\{b_{(wb)t}\} = \sum_{i \in \mathbb{R}^n} \alpha_1 \cdot b_t^i + \alpha_2 \cdot b_{(mk)t}^i$	◀ Weighed bounding box
13: $F'_{(tb)t} = \{b_{(wb)t}\} \rightarrow F'_t$	◀ Object-aware features maps
14: $\{t_t\} = \mathbf{N}_{tr}(F'_{(tb)t})$	◀ Produce tracking results
13: $R_t = (\{b_t\}, \{c_t\}, \{m_t\}, \{t_t\})$	◀ Final recognition at t
15: output: Video recognition result R_t .	

connections are parameters-free and only work as information flow which is fast, we exclude it in our analysis. Given a temporal range of n in flow-guided feature fusion, the optical flow network \mathbf{N}_{fl} loops n times per frame for feature warping. Accordingly, the runtime complexity for the proposed method is,

$$O_{ours} = O(\mathbf{N}_{fm}) + n \cdot O(\mathbf{N}_{fl}) + O(\mathbf{N}_{cl}) + O(\mathbf{N}_{bb}) + O(\mathbf{N}_{mk}) + O(\mathbf{N}_{tr}) \quad (8)$$

We compare the time complexity of our method to its predecessor, TrackR-CNN [81] as both of them have similar architecture and TrackR-CNN is the state-of-the-art multi-object tracking and segmentation. With the same symbols, TrackR-CNN has the following runtime complexity,

$$O_{tr} = O(\mathbf{N}_{fm}) + m \cdot O(\mathbf{N}_{3D}) + O(\mathbf{N}_{cl}) + O(\mathbf{N}_{bb}) + O(\mathbf{N}_{mk}) + O(\mathbf{N}_{tr}), \quad (9)$$

where $O(\mathbf{N}_{3D})$ is the 3D convolutions and m is the feature fusion length used in TrackR-CNN for per frame feature extraction.

Typically, the backbone feature extraction network has a heavier architecture than the task heads because it contains many more layers for feature abstraction. Therefore, it is reasonable to assume $O(\mathbf{N}_{cl}) \ll O(\mathbf{N}_{fm})$, $O(\mathbf{N}_{bb}) \ll O(\mathbf{N}_{fm})$, $O(\mathbf{N}_{mk}) \ll O(\mathbf{N}_{fm})$, $O(\mathbf{N}_{tr}) \ll O(\mathbf{N}_{fm})$, and $O(\mathbf{N}_{mk}) \ll O(\mathbf{N}_{fm})$. Given these, the ratio of runtime complexity of the proposed method to TrackR-CNN [81] can be computed as:

$$C = \frac{O_{ours}}{O_{tr}} \approx \frac{O(\mathbf{N}_{fm}) + n \cdot O(\mathbf{N}_{fl})}{O(\mathbf{N}_{fm}) + m \cdot O(\mathbf{N}_{3D})}. \quad (10)$$

Compared to \mathbf{N}_{3D} used in TrackR-CNN, we use \mathbf{N}_{fl} which is more efficient while still being effective for object movement modeling. Assuming $n = m$, C in Eq. (10) is less than 1. Therefore, our method is faster than TrackR-CNN. Experimental results in Table 2 also demonstrate that our method is two times faster than TrackR-CNN.

3.5.3 Training objective. Since our method simultaneously considers object detection, segmentation, and tracking, its training objective contains multiple losses accordingly,

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask} + \mathcal{L}_{track}, \quad (11)$$

where \mathcal{L}_{cls} , \mathcal{L}_{box} , \mathcal{L}_{mask} , and \mathcal{L}_{track} denotes the loss for classification, bounding box regression, mask segmentation, and object tracking respectively. Following the implementation from Mask R-CNN[34], we modify the classification loss \mathcal{L}_{cls} and bounding-box loss \mathcal{L}_{box} based those from Faster R-CNN[72]. In detail, we add extra losses before and after the associative connections across tasks for full supervision. Therefore, both the predictions before and after the associative connections are optimized. As we use a set of $m \times m$ for mask generation, the mask head has a cm^2 -dimensional output for each ROI, which encodes c binary masks. To achieve this goal, we use a per-pixel sigmoid, and \mathcal{L}_{mask} is defined as the average binary cross-entropy loss. For an ROI associated with ground-truth class k , we only define \mathcal{L}_{mask} on the k^{th} mask while ignoring other mask outputs that do not contribute to the loss. The details of \mathcal{L}_{track} are discussed in Section 3.4.2.

As the entire framework of the proposed method is multiple-stage, we rely on predictions from the classification head for the label of object detection, instance segmentation (mask), and tracking. Using the classification result for each ROI, we allow each task head to generate its results. Namely, the bounding-box head, the mask head, and the tracking head only need to focus on their specific task based on ROI without competition among classes. With the associative connections, critical ROI information can forward pass from the bounding box head to the mask head and eventually to the tracking head respectively. In the same vein, the ground truth for instance segmentation and object tracking can be backpropagated to each task head in the training procedure.

4 EXPERIMENTS

This section first describes the datasets and evaluation metrics that we use to assess our method. The implementation details with training parameter settings are then elaborated. Next, We supply ablation studies to investigate the improvements of our method from a strong baseline and its variants. We also compare our method with the state-of-the-art methods on two benchmarks. Finally, we conclude with the accuracy and runtime evaluation.

Table 2. Comparison with the state-of-the-art methods on the KITTI MOTS dataset.

Method	Detect + Segment	Speed (s)	FPS	Cars / Pedestrians			
				sMOTSA↑	MOTSA↑	MOTSP↑	IDS↓
CAMOT [67]	TRCNN	0.76	1.32	67.4 / 39.5	78.6 / 57.6	86.5 / 73.1	220 / 131
CIWT[66]	TRCNN	0.28	3.57	68.1 / 42.9	79.4 / 61.0	86.7 / 75.7	106 / 42
ReMOTS[98]	TRCNN + BB2SegNet	3.33 / -	0.30	70.4 / -	84.4 / -	- / -	231 / -
TrackR-CNN[81]	TRCNN	0.50	2.00	76.2 / 46.8	87.8 / 65.1	87.2 / 75.7	93 / 78
BePix [74]	RRC[71] + TRCNN	0.36	2.77	76.9 / -	89.7 / -	86.5 / -	88 / -
Stem-Seg [2]	TRCNN	0.32	3.12	72.7 / 50.4	83.8 / 66.1	87.2 / 77.7	76 / 14
Ours	TRCNN	0.27	3.70	76.7 / 47.9	88.2 / 65.3	88.5 / 76.1	62 / 32

4.1 Datasets and Evaluation Metrics

KITTI MOTS [81] and MOTS Challenge [64] are used to evaluate the effectiveness of the proposed method. KITTI MOTS has 21 video sequences of 8,008 frames, while MOTSChallenge has four video sequences of 2,862 frames. For KITTI MOTS, it includes 11,420 pedestrian and 26,899 car instances. There are 26,894 pedestrian instances in the MOTS Challenge.

Following TrackR-CNN [81], we use **MOTSA** to measure the accuracy for multi-object tracking and segmentation, **MOTSP** to measure the precision of mask-based multi-object tracking and segmentation results, and **sMOTSA** to evaluate the soft multi-object tracking and segmentation accuracy. They are defined as follows,

$$\begin{aligned} MOTSA &= \frac{|TP| - |FP| - |IDS|}{|M|}, \\ MOTSP &= \frac{|\widetilde{TP}|}{|TP|}, \\ sMOTSA &= \frac{\widetilde{TP} - |FP| - |IDS|}{|M|}, \end{aligned} \tag{12}$$

where TP , \widetilde{TP} , FP , and IDS are true positive, soft true positive, false positive, and tracking ID switch score respectively. We refer readers to [81] for more details about each notation. These metrics collectively measure the performance of a multi-object tracking and segmentation system by considering three tasks (i.e., object detection, instance segmentation, and multi-object tracking) together.

4.2 Implementation Details

The proposed method is implemented on a workstation with one NVIDIA RTX GPU. To have a fair comparison with existing methods, we follow the same experimental setup as in TrackR-CNN [81]. To be specific, the backbone feature extraction network ResNet-101 [35] is pretrained on COCO [51] and Mapillary [65] datasets. The optical flow network FlowNet is pretrained on the Flying Chair dataset [26]. During the training process, the weights of ResNet-101 and FlowNet are fixed, and the other weights related to different task heads (i.e., \mathbf{N}_{bb} , \mathbf{N}_{cl} , \mathbf{N}_{mk} and \mathbf{N}_{tr}) are updated by learning on the target dataset, i.e. KITTI MOTS or MOTS Challenge. We train our model for 40 epochs with a learning rate of 5×10^{-7} using the Adam [48] optimizer and mini-batch size of eight. The temporal range n in Eq. (2) is set to eight, i.e., eight adjacent frames are used for the flow-guided feature extraction.

For the KITTI MOTS benchmark, there are 21 videos in total and we use 12 of them for training and the remaining for testing, following the practice in TrackR-CNN. We randomly keep some training data for validation and choose the best model on the validation set for testing. For the MOTS Challenge benchmark, since there are only four video sequences in total, we use cross-validation to test the performance of different methods. To be specific, we leave one video sequence for evaluation and train the model on the three others on the MOTS Challenge. This process is repeated four times and the average result is reported.

4.3 Comparison with the State-of-the-art Methods

Results on KITTI MOTS. The results compared with the state-of-the-art methods on KITTI MOTS dataset are shown in Table 2. For a fair comparison, we list the leading methods with detection and segmentation architectures based on TrackR-CNN, which is the same as ours. The best result for each metric is highlighted in the table. The results of our method in Table 2 are quite encouraging. For car recognition, our method achieves the best result on MOTSP and IDS. Especially, our method improves IDS over other methods by a significantly large margin, which leads to the second-best method by 26. sMOTSA and MOTSA of our method are on par with the best model, a.k.a., BePix [74]. Regarding speed, our method is faster than BePix. The fast inference speed of our method is due to its flow-guided feature extraction in which a lightweight flow network is used. Finally, for pedestrian recognition, our method is better than all the compared methods on these metrics.

To qualitatively demonstrate the improvements of our method, we visualize our results and compare them with TrackR-CNN. As shown in Fig. 2, our method is less prone to false predictions. In the left two columns of Fig. 2, TrackR-CNN in the top row produces a "car" prediction on the red minivan while our method in the

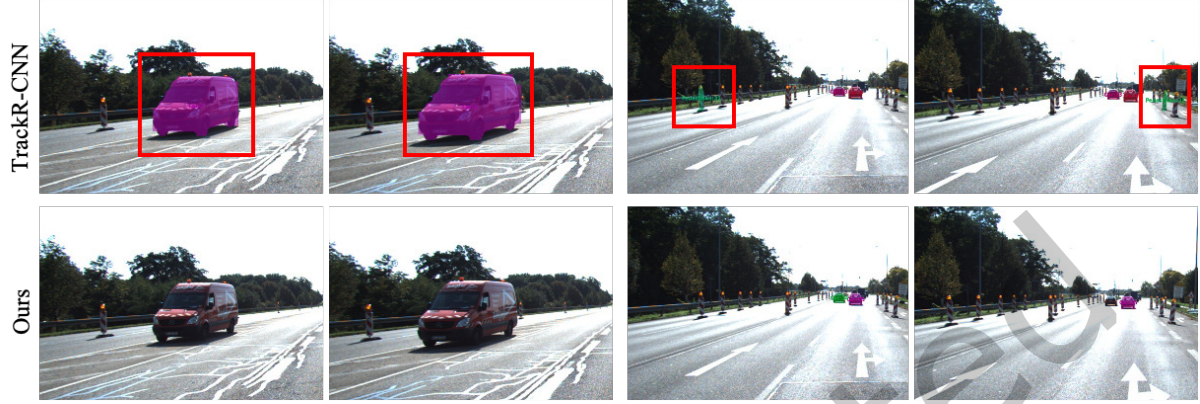


Fig. 2. Qualitative comparison between TrackR-CNN and our method regarding false prediction. The false predictions are marked by red bounding boxes. Minivan is not an object in the KITTI MOTs dataset. In the first two rows, TrackR-CNN has a false prediction on the minivan as a car class. In the same vein, TrackR-CNN has a false recognition of traffic signs and predicts them as "pedestrian". For both scenarios, our method works correctly without false predictions. The images are cropped for better visualization.

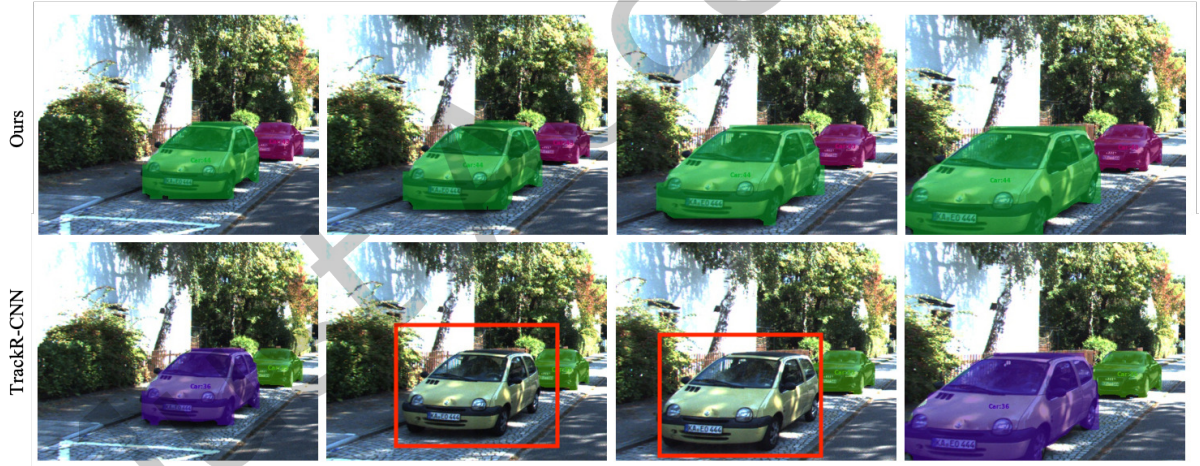


Fig. 3. Comparison of TrackR-CNN and our results on missed recognition. The missed recognition is marked in red bounding boxes. Our method can constantly perform recognition on video frames while TrackR-CNN encounters recognition lost for the middle two frames. The demonstrated examples are cropped for better visualization.

bottom row does not. A minivan is not a labeled object in the KITTI MOTs dataset. For the right two columns of Fig. 2, TrackR-CNN produces a false detection on traffic signs, predicting them as "pedestrian". In these examples, our method performs accurate predictions for both scenarios. What is more, TrackR-CNN frequently encounters missing detection. For comparison, our method can constantly detect objects through video frames

as demonstrated in Fig. 3. The improvement of our method over TrackR-CNN mainly stems from the reliable object-aware features, which are brought from our associative connections. With the help of these connections, our method produces better feature representations than TrackR-CNN before the task heads produce individual predictions, thus achieving better results.

We also observe that our method has improved performance on tracking and mask quality. We argue that the proposed associative connections benefit the mask and tracking predictions as for the two tasks, we use refined ROI features for the final prediction. In contrast, TrackR-CNN produces predictions using coarse features based on proposals.

Results on MOTS Challenge. Table 3 reports the results on the MOTS Challenge dataset. Our method outperforms all the other methods for all metrics. The improvements over the second best method (TrackR-CNN), are 0.5 on sMOTSA, 0.2 on MOTSA, and 0.7 on MOTSP respectively. Besides TrackR-CNN, our method achieves significantly better results than other methods even if they use a domain-finetuned Mask R-CNN.

Table 3. Comparison with the state-of-the-art methods on MOTS Challenge dataset. +MG denotes mask generation with a domain finetuned Mask R-CNN.

Method	sMOTSA↑	MOTSA↑	MOTSP↑
MOTDT[13] + MG	47.8	61.1	80.0
MHT-DAM[46] + MG	48.0	62.7	79.8
jCC[45] + MG	48.3	63.0	79.9
FWT[36] + MG	48.3	64.0	79.7
TrackR-CNN[81]	52.7	66.9	80.2
Ours	53.2	67.1	80.9

4.4 Ablation Study

To demonstrate the effectiveness of our design, we conduct ablation studies from TrackR-CNN since our method is proposed for the same purpose as TrackR-CNN and shares a similar architecture to it. Specifically, we first replace the computationally expensive conv3d in TrackR-CNN with more efficient flow-guided feature extraction proposed in Section 3.3, which leads to Method A. Then, we add an associative connection from the bounding box head to the mask head as described in Section 3.4.1 and obtain Method B. By further adding an associative connection from the bounding-box head to track, this leads to Method C in which bounding box features are used for both segmentation and tracking, rather than the proposal based features. A variant of Method C is to connect the mask-based bounding box to the tracking head, which we call Method D. We also test two kinds of strategies to fuse bounding boxes and mask-based bounding boxes, which are then input to the tracking head. Method E uses fixed weight $\alpha = 0.5$ to fuse these two kinds of bounding boxes, while Method F is the proposed one that fuses them with adaptive weights.

Table 4 demonstrates the results of each method on the KITTI MOTS dataset. For reference, we also include the results of TrackR-CNN. By comparing A to TrackR-CNN, we can find that using flow-guided feature extraction leads to slightly worse accuracy, however, it leads to faster runtime as shown in Table 2. Based on A, B obtains better results, proving our associative connection that using predicted bounding boxes is better than region proposals in pooling features for the mask head. C and D further improve B, demonstrating the effectiveness of introducing an associative connection to the tracking head. If these two kinds of bounding boxes together, better results are obtained. More importantly, the proposed adaptive box fusion strategy (Method F) achieves significant improvement over the hard fusion (Method E), especially for the tracking consistency. These results

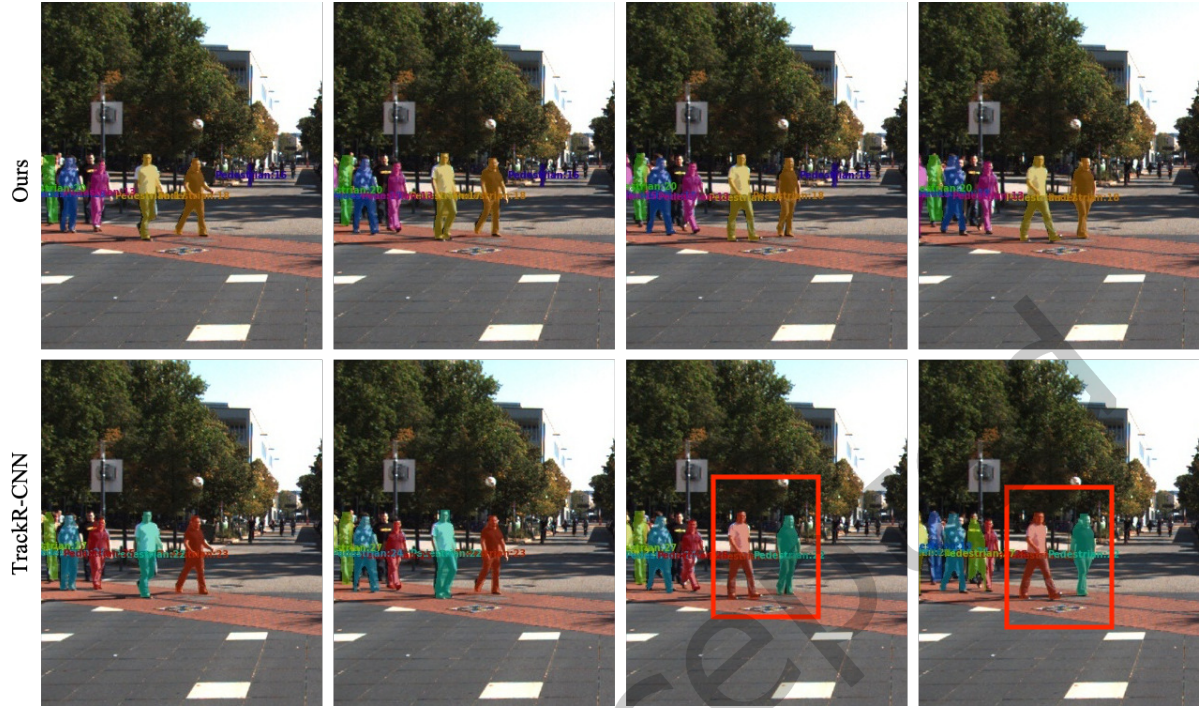


Fig. 4. Comparison of TrackR-CNN and our results on ID switch. The incorrect cases for tracking are marked in red bounding boxes. Our method can constantly perform accurate tracking of the same instances on video frames while TrackR-CNN frequently produces incorrect predictions. The demonstrated examples are cropped for better visualization.

show that the proposed method effectively explores the properties of these two kinds for multi-object tracking and segmentation.

Table 4. Ablation study on the KITTI MOTS dataset. Please see the texts for details about methods A to E, and F is the proposed method.

Method		TrackR-CNN	A	B	C	D	E	F
Cars	sMOTSA	76.2	75.6	76.0	76.1	76.5	76.3	76.5
	MOTSA	87.8	87.0	87.1	87.5	88.1	87.9	88.2
	MOTSP	87.2	86.9	87.1	87.4	88.4	88.3	88.5
	IDS	93	95	87	76	74	73	62
Pedestrians	sMOTSA	46.8	45.2	45.5	46.1	46.1	46.2	47.9
	MOTSA	65.1	63.5	63.8	64.3	64.3	64.4	65.3
	MOTSP	75.7	75.0	75.1	75.8	75.9	76.0	76.1
	IDS	78	76	43	39	38	36	32

4.5 Fusion Length and Accuracy

We use eight frames as the default temporal range for feature fusion based on the object movements across frames. To evaluate the impact of this fusion length on the accuracy, we change the number of input frames for our method and test our method on the KITTI MOTS dataset. Table 5 reports the results concerning the number of input frames. For both cars and pedestrians, the recognition accuracy increases when the temporal range increases from 2 to 8. After that, the performance starts to degrade. Such results are reasonable, because only a few frames may not accumulate enough information to enhance the feature of the current frame while too many frames are also damageable as the long-term movements estimated by the optical flow network are unstable. In addition, it is also worth pointing out that a larger temporal range requires more time to process feature extraction, thus will result in a lower speed. Therefore, we need to pay more attention to the temporal range in practice to achieve a desirable recognition accuracy as well as the runtime speed.

Table 5. Influence of the temporal range used in the flow-guided feature fusion. The results are on the KITTI MOTS dataset.

temporal range n		2	4	8	12	16
Cars	sMOTSA↑	75.2	75.7	76.7	75.4	75.3
	MOTSA↑	86.5	87.6	88.2	86.8	86.7
	MOTSP↑	87.3	87.4	88.5	87.8	87.2
	IDS↓	87	75	62	73	78
Pedestrians	sMOTSA↑	46.2	46.9	47.9	46.5	46.3
	MOTSA↑	63.5	64.3	65.3	63.7	63.6
	MOTSP↑	75.4	75.7	76.1	75.5	75.4
	IDS↓	39	36	32	38	35

5 CONCLUSION

In this work, we propose a novel algorithm capable of associatively detecting, segmenting, and tracking multiple objects for video analysis. By adding associative connections across detection, segmentation, and tracking heads in an end-to-end learnable CNN, our method enables information propagation through different tasks, which could benefit all the considered tasks. Therefore, our method achieves state-of-the-art performance on various metrics regarding object detection, instance segmentation, and multi-object tracking, according to our evaluation of two benchmarks. We also conducted extensive ablation studies to demonstrate the effectiveness of each associative connection.

REFERENCES

- [1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. 2018. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5167–5176.
- [2] Ali Athar, Sabarinath Mahadevan, Aljoša Ošep, Laura Leal-Taixé, and Bastian Leibe. 2020. STEm-Seg: Spatio-temporal Embeddings for Instance Segmentation in Videos. *arXiv preprint arXiv:2003.08429* (2020).
- [3] Gedas Bertasius and Lorenzo Torresani. 2020. Classifying, Segmenting, and Tracking Object Instances in Video with Mask Propagation. In *CVPR. IEEE, Virtual*, 9739–9748.
- [4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).
- [5] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. 2019. YOLACT: Real-time Instance Segmentation. In *ICCV. IEEE, Seoul, Korea*, 9157–9166.

- [6] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. 2020. YOLACT++: Better Real-time Instance Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).
- [7] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. 2020. SipMask: Spatial Information Preservation for Fast Image and Video Instance Segmentation. In *ECCV*. Springer, Glasgow, UK, 1–18.
- [8] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. 2020. SipMask: Spatial Information Preservation for Fast Image and Video Instance Segmentation. *arXiv preprint arXiv:2007.14772* (2020).
- [9] Zhiwen Cao, Zongcheng Chu, Dongfang Liu, and Yingjie Chen. 2021. A vector-based representation to enhance head pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE, virtual, 1188–1197.
- [10] Zhiwen Cao, Dongfang Liu, Qifan Wang, and Yingjie Chen. 2022. Towards unbiased label distribution learning for facial pose estimation using anisotropic spherical gaussian. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*. Springer, Springer, Tel Aviv, Israel, 737–753.
- [11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. In *ECCV*. Springer, Glasgow, UK, 213–229.
- [12] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. 2019. Hybrid task cascade for instance segmentation. In *CVPR*. IEEE, Long Beach, CA, USA, 4974–4983.
- [13] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. 2018. Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification. In *ICME*.
- [14] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. 2020. Memory enhanced global-local aggregation for video object detection. In *CVPR*. IEEE, Virtual, 10337–10346.
- [15] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. 2021. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764* (2021).
- [16] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. 2017. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*. IEEE, Honolulu, Hawaii, USA, 686–695.
- [17] Peng Chu and Haibin Ling. 2019. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *ICCV*. IEEE, Long Beach, CA, USA, 6172–6181.
- [18] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2015. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, Vol. 2.
- [19] Yiming Cui. 2022. DFA: Dynamic Feature Aggregation for Efficient Video Object Detection. *arXiv preprint arXiv:2210.00588* (2022).
- [20] Yiming Cui. 2023. Feature Aggregated Queries for Transformer-Based Video Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6365–6376.
- [21] Yiming Cui, Zhiwen Cao, Yixin Xie, Xingyu Jiang, Feng Tao, Yingjie Victor Chen, Lin Li, and Dongfang Liu. 2022. Dg-labeler and dgl-mots dataset: Boost the autonomous driving perception. In *WACV*. IEEE, Waikoloa, HI, USA, 58–67.
- [22] Yiming Cui, Liqi Yan, Zhiwen Cao, and Dongfang Liu. 2021. TF-Blender: Temporal Feature Blender for Video Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8138–8147.
- [23] Yiming Cui, Linjie Yang, and Ding Liu. 2022. Dynamic proposals for efficient object detection. *arXiv preprint arXiv:2207.05252* (2022).
- [24] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*. MIT Press, Barcelona, Spain, 379–387.
- [25] Zheng Dong, Yan Lu, Guangmo Tong, Yuanhao Shu, Shuai Wang, and Weisong Shi. 2023. Watchdog: Real-time vehicle tracking on geo-distributed edge nodes. *ACM Transactions on Internet of Things* 4, 1 (2023), 1–23.
- [26] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philipp Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. FlowNet: Learning optical flow with convolutional networks. In *ICCV*. IEEE, Santiago, Chile, 2758–2766.
- [27] Badr Ben Elallid, Sara El Hamdani, Nabil Benamar, and Nabil Mrani. 2022. Deep learning-based modeling of pedestrian perception and decision-making in refuge island for autonomous driving. In *Computational Intelligence in Recent Communication Networks*. Springer, 135–146.
- [28] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. 2021. Instances As Queries. In *ICCV*. IEEE, Virtual, 6910–6919.
- [29] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3354–3361.
- [30] Ross Girshick. 2015. Fast r-cnn. In *ICCV*. IEEE, Santiago, Chile, 1440–1448.
- [31] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*. IEEE, 580–587.
- [32] J.W. Han, D.W. Zhang, X.T. Hu, L. Guo, J.C. Ren, and F. Wu. 2015. Background prior-based salient object detection via deep reconstruction residual. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 8 (2015), 1309–1321.
- [33] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. 2016. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465* (2016).

- [34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *ICCV*. IEEE, Venice, Italy, 2961–2969.
- [35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. IEEE, Honolulu, Hawaii, USA, 770–778.
- [36] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. 2018. Fusion of head and full-body detectors for multi-object tracking. In *CVPR workshops*. IEEE, Salt Lake City, UT, USA, 1428–1437.
- [37] Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. 2022. Vita: Video instance segmentation via object token association. *Advances in Neural Information Processing Systems* 35 (2022), 23109–23120.
- [38] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. 2018. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 954–960.
- [39] Zhenhua Huang, Xin Xu, Haibo He, Jun Tan, and Zhenping Sun. 2019. Parameterized Batch Reinforcement Learning for Longitudinal Control of Autonomous Land Vehicles. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49, 4 (2019), 730–741.
- [40] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. 2021. Video instance segmentation using inter-frame communication transformers. *Advances in Neural Information Processing Systems* 34 (2021), 13352–13363.
- [41] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. 2017. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 10 (2017), 2896–2907.
- [42] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2016. Object detection from video tubelets with convolutional neural networks. In *CVPR*. IEEE, Las Vegas, Nevada, USA, 817–825.
- [43] Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. 2022. Mask transfiner for high-quality instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4412–4421.
- [44] Lei Ke, Henghui Ding, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. 2022. Video mask transfiner for high-quality video instance segmentation. In *European Conference on Computer Vision*. Springer, 731–747.
- [45] Margret Keuper, Siyu Tang, B. Andres, T. Brox, and B. Schiele. 2020. Motion Segmentation and Multiple Object Tracking by Correlation Co-Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2020), 140–153.
- [46] Chanh Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. 2015. Multiple Hypothesis Tracking Revisited. In *ICCV*. IEEE, Santiago, Chile.
- [47] Chanh Kim, Fuxin Li, and James M Rehg. 2018. Multi-object tracking with neural gating using bilinear lstm. In *ECCV*. Springer, 200–215.
- [48] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. (2014). arXiv:1412.6980 [cs.LG]
- [49] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. 2017. Fully convolutional instance-aware semantic segmentation. In *CVPR*. IEEE, Honolulu, Hawaii, USA, 2359–2367.
- [50] Huajia Lin, Xiaojuan Qi, and Jiaya Jia. 2019. Agss-vos: Attention guided single-shot video object segmentation. In *ICCV*. IEEE, Seoul, Korea, 3949–3957.
- [51] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft COCO: Common Objects in Context. (2014). arXiv:1405.0312 [cs.CV]
- [52] Dongfang Liu, Yiming Cui, Zhiwen Cao, and Yingjie Chen. 2020. Indoor navigation for mobile agents: A multimodal vision fusion model. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, IEEE, virtual, 1–8.
- [53] Dongfang Liu, Yiming Cui, Zhiwen Cao, and Yingjie Chen. 2020. A large-scale simulation dataset: Boost the detection accuracy for special weather conditions. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
- [54] Dongfang Liu, Yiming Cui, Yingjie Chen, Jiyong Zhang, and Bin Fan. 2020. Video Object Detection For Autonomous Driving: Motion-aid Feature Calibration. *Neurocomputing* (2020), 1–11.
- [55] Dongfang Liu, Yiming Cui, Xiaolei Guo, Wei Ding, Baijian Yang, and Yingjie Chen. 2021. Visual localization for autonomous driving: Mapping the accurate location in the city maze. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 3170–3177.
- [56] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. 2021. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*. IEEE, Virtual, 9816–9825.
- [57] Dongfang Liu, Yiming Cui, Liqi Yan, Christos Mousas, Baijian Yang, and Yingjie Chen. 2021. Densernet: Weakly supervised visual localization using multi-scale feature aggregation. In *AAAI*. AAAI, Virtual, 6101–6109.
- [58] Jia Liu, Maoguo Gong, and Haibo He. 2019. Deep associative neural network for associative memory based on unsupervised representation learning. *Neural Networks* 113 (2019), 41 – 53.
- [59] Liangkai Liu, Zheng Dong, Yanzhi Wang, and Weisong Shi. 2022. Prophet: Realizing a Predictable Real-time Perception Pipeline for Autonomous Vehicles. In *2022 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 305–317.
- [60] Xiao Liu, Dacheng Tao, Mingli Song, Luming Zhang, Jiajun Bu, and Chun Chen. 2015. Learning to Track Multiple Targets. *IEEE Transactions on Neural Networks and Learning Systems* 26, 5 (2015), 1060–1073.

- [61] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. 2020. Unovost: Unsupervised offline video object segmentation and tracking. In *WACV*. IEEE, Snowmass Village, CO, USA, 2000–2009.
- [62] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan. 2018. Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network. *IEEE Transactions on Intelligent Transportation Systems* 19, 4 (2018), 1100–1111.
- [63] Santiago Manen, Michael Gygli, Dengxin Dai, and Luc Van Gool. 2017. Pathtrack: Fast trajectory annotation with path supervision. In *Proceedings of the IEEE International Conference on Computer Vision*. 290–299.
- [64] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831* (2016).
- [65] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kotschieder. 2017. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *ICCV*. IEEE, Venice, Italy.
- [66] Aljoša Osep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. 2017. Combined image-and world-space tracking in traffic scenes. In *ICRA*. IEEE, IEEE, Marina Bay Sands, Singapore, 1988–1995.
- [67] Aljoša Osep, Wolfgang Mehner, Paul Voigtlaender, and Bastian Leibe. 2018. Track, then decide: Category-agnostic vision-based multi-object tracking. In *ICRA*. IEEE, Brisbane, Australia, 1–8.
- [68] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. 2019. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*. IEEE, Long Beach, CA, USA, 821–830.
- [69] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. 2017. Learning video object segmentation from static images. In *CVPR*. IEEE, Honolulu, Hawaii, USA, 2663–2672.
- [70] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.
- [71] Jimmy Ren, Xiaohao Chen, Jianbo Liu, Wenxiu Sun, Jiahao Pang, Qiong Yan, Yu-Wing Tai, and Li Xu. 2017. Accurate single stage detector using recurrent rolling convolution. In *CVPR*. IEEE, Honolulu, Hawaii, USA, 5420–5428.
- [72] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*. MIT Press, Montreal, Canada, 91–99.
- [73] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Jun 2017).
- [74] Sarthak Sharma, Junaid Ahmed Ansari, J. Krishna Murthy, and K. Madhava Krishna. 2018. Beyond Pixels: Leveraging Geometry and Shape Cues for Online Multi-Object Tracking. In *ICRA*. IEEE, Brisbane, Australia.
- [75] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. 2021. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*. IEEE, Virtual, 14454–14463.
- [76] Shijie Sun, Naveed Akhtar, HuanSheng Song, Ajmal S Mian, and Mubarak Shah. 2019. Deep affinity network for multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).
- [77] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. 2017. Multiple people tracking by lifted multicut and person re-identification. In *CVPR*. IEEE, Honolulu, Hawaii, USA, 3539–3548.
- [78] Zhi Tian, Chunhua Shen, and Hao Chen. 2020. Conditional Convolutions for Instance Segmentation. In *ECCV*. Springer, Glasgow, UK.
- [79] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *ICCV*. IEEE, Seoul, Korea, 9627–9636.
- [80] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NeurIPS* 30 (2017).
- [81] Paul Voigtlaender, Michael Krause, Aljoša Osep, Jonathon Luiten, Berin Balachandrar Gnana Sekar, Andreas Geiger, and Bastian Leibe. 2019. MOTs: Multi-object tracking and segmentation. In *CVPR*. IEEE, Long Beach, CA, USA, 7942–7951.
- [82] Guang Wang, Zhou Qin, Shuai Wang, Huijun Sun, Zheng Dong, and Desheng Zhang. 2022. Towards Accessible Shared Autonomous Electric Mobility with Dynamic Deadlines. *IEEE Transactions on Mobile Computing* (2022).
- [83] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. 2019. Fast online object tracking and segmentation: A unifying approach. In *CVPR*. IEEE, Long Beach, CA, USA, 1328–1338.
- [84] Shiyao Wang, Yucong Zhou, Junjie Yan, and Zhidong Deng. 2018. Fully motion-aware network for video object detection. In *Proceedings of the European conference on computer vision (ECCV)*. 542–557.
- [85] Xinchao Wang, Bin Fan, Shiyu Chang, Zhangyang Wang, Xianming Liu, Dacheng Tao, and Thomas S. Huang. 2017. Greedy Batch-based Minimum-cost Flows for Tracking Multiple Objects. *IEEE Transactions on Image Processing* 26, 10 (2017), 4765–4776.
- [86] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. 2020. SOLO: Segmenting Objects by Locations. In *ECCV*. Springer, Glasgow, UK, 649–665.
- [87] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. 2020. SOLOv2: Dynamic and Fast Instance Segmentation. In *NeurIPS*. MIT Press, virtual, 17721–17732.
- [88] Yaqin Wang, Dongfang Liu, Hyewon Jeon, Zhiwei Chu, and Eric T Matson. 2019. End-to-end Learning Approach for Autonomous Driving: A Convolutional Neural Network Model. In *ICAART (2)*. IEEE, Prague, Czechia, 833–839.

- [89] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. 2021. End-to-end video instance segmentation with transformers. In *CVPR*. IEEE, Virtual, 8741–8750.
- [90] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. 2015. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv preprint arXiv:1511.04136* (2015).
- [91] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. 2019. Sequence level semantics aggregation for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9217–9225.
- [92] Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai. 2022. Seqformer: Sequential transformer for video instance segmentation. In *European Conference on Computer Vision*. Springer, 553–569.
- [93] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. 2022. In defense of online models for video instance segmentation. In *European Conference on Computer Vision*. Springer, 588–605.
- [94] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Xiangbo Su, Yuchen Yuan, Hongwu Zhang, Shilei Wen, Errui Ding, and Liusheng Huang. 2020. PointTrack++ for Effective Online Multi-Object Tracking and Segmentation. *arXiv preprint arXiv:2007.01549* (2020).
- [95] Liqi Yan, Yiming Cui, Yingjie Chen, and Dongfang Liu. 2021. Hierarchical attention fusion for geo-localization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, IEEE, virtual, 2220–2224.
- [96] Liqi Yan, Siqi Ma, Qifan Wang, Yingjie Chen, Xiangyu Zhang, Andreas Savakis, and Dongfang Liu. 2022. Video captioning using global-local representation. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 10 (2022), 6642–6656.
- [97] Liqi Yan, Qifan Wang, Yiming Cui, Fuli Feng, Xiaojun Quan, X. Zhang, and Dongfang Liu. 2022. GL-RG: Global-Local Representation Granularity for Video Captioning. In *International Joint Conference on Artificial Intelligence*.
- [98] Fan Yang, Xin Chang, Chenyu Dang, Ziqiang Zheng, Sakriani Sakti, Satoshi Nakamura, and Yang Wu. 2020. ReMOTS: Self-Supervised Refining Multi-Object Tracking and Segmentation. *arXiv e-prints* (2020), arXiv–2007.
- [99] Linjie Yang, Yuchen Fan, and Ning Xu. 2019. Video instance segmentation. In *ICCV*. IEEE, Seoul, Korea, 5188–5197.
- [100] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. 2018. Efficient video object segmentation via network modulation. In *CVPR*. IEEE, Salt Lake City, UT, USA, 6499–6507.
- [101] Shusheng Yang, Xinggang Wang, Yu Li, Yuxin Fang, Jiemin Fang, Wenyu Liu, Xun Zhao, and Ying Shan. 2022. Temporally efficient vision transformer for video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2885–2895.
- [102] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2636–2645.
- [103] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. 2019. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems* 30, 11 (2019), 3212–3232.
- [104] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. 2018. Towards high performance video object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7210–7218.
- [105] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*. <https://openreview.net/forum?id=gZ9hCDWe6ke>
- [106] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Flow-guided feature aggregation for video object detection. In *ICCV*. IEEE, Venice, Italy, 408–417.
- [107] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Deep feature flow for video recognition. In *CVPR*. IEEE, Honolulu, Hawaii, USA, 2349–2358.