SearchTrack: Multiple Object Tracking with Object-Customized Search and Motion-Aware Features – Supplementary Material

Zhong-Min Tsai[1*]
vtsai01@cmlab.csie.ntu.edu.tw
Yu-Ju Tsai[1*]
r06922009@cmlab.csie.ntu.edu.tw
Chien-Yao Wang[2]
kinyiu@iis.sinica.edu.tw
Hong-Yuan Liao[2]
liao@iis.sinica.edu.tw
Youn-Long Lin[3]
ylin@cs.nthu.edu.tw
Yung-Yu Chuang[1]
cyy@csie.ntu.edu.tw

[1] National Taiwan University
Taipei, Taiwan
[2] Institute of Information Science,
Academia Sinica
Taipei, Taiwan
[3] National Tsing Hua University
Hsinchu, Taiwan

1 Model architecture

Fig 1 illustrates our model architecture for generating the response map. The output feature map from DLA-34 [10] has the size of $W \times H \times 64$, where $W = W_{Image}/R$ and $H = H_{Image}/R$ with a downsampling ratio of 4 ($R = 4$) in our implementation.

The choices of C_{search} relate to the number of parameters of the dynamic weight θ for the dynamic searcher $\mathcal{T}$. We set the channels of the search branch C_{search} to 16 and the related number of parameters for the dynamic weight θ is 593, which is $\#weights + \#bias$ ($\#weights = (16+2) \times 16(conv1) + 16 \times 16(conv2) + 16 \times 1(conv3)$ and $\#bias = 16(conv1) + 16(conv2) + 1(conv3)$).

2 Datasets and metrics for evaluation

KITTI MOTS [10]. KITTI MOTS is a popular MOTS benchmark, which consists of 12 sequences for training, 9 for validation, and 29 for testing from the KITTI Tracking Evaluation 2012 benchmark [7]. The dataset provides instance segmentation annotations for cars and pedestrians. The videos in the dataset are captured at 10 FPS and contain large inter-frame

*These authors contributed equally to this work.

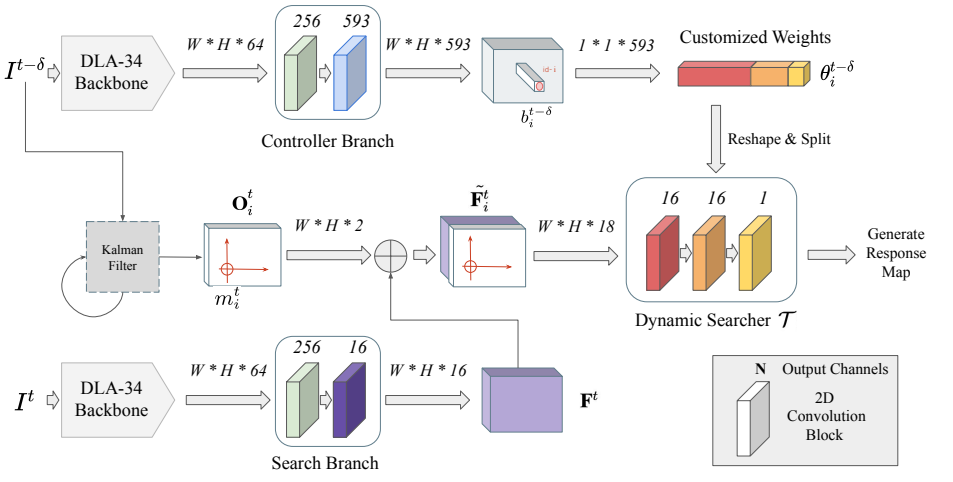

Figure 1: **Detailed SearchTrack architecture for generating the response map.** A shared backbone network (DLA-34 [41]) extracts features from the current frame I^t and the previous frame $I^{t-\delta}$. The features from the current frame I^t is fed to the search branch to form a search feature map $\mathbf{F}^t$. In addition, the features from the previous frame $I^{t-\delta}$ is fed to the controller branch to generate customized weights. For a query object $b_i^{t-\delta}$ in the previous frame, we obtain its customized weight $\theta_i^{t-\delta}$ and predicted location m_i^t from its Kalman filter. The location m_i^t is encoded in a motion map $\mathbf{O}_i^t$. Combining $\mathbf{O}_i^t$ and $\mathbf{F}^t$ gives us the motion-aware motion map $\tilde{\mathbf{F}}_i^t$. The customized weight $\theta_i^{t-\delta}$ is reshaped and distributed to become the corresponding convolutional weight in the dynamic searcher $\mathcal{T}$ for the object $b_i^{t-\delta}$. By taking $\tilde{\mathbf{F}}_i^t$ as input, $\mathcal{T}$ outputs a response map indicating where the object could locate.

motions for objects. The KITTI benchmark does not provide the detection results, and all the methods on the benchmark use their own detection results, also called private detection.

MOT17 [27]. MOT17 is a widely used benchmark for the MOT task. It contains 7 training sequences and 7 test sequences. The dataset videos are captured at 25-30 FPS with heavy occlusion. Only pedestrians are annotated and evaluated. Because this MOT dataset does not provide the official validation set, we follow the setting in CenterTrack and split each training sequence into two halves, one for training and the other for validation, for the experiments.

Evaluation metrics. We adopt HOTA [33, 34] as the primary evaluation metric for our experiments. HOTA is also the primary evaluation metric used in KITTI MOTS and MOT17 for comparisons and ranking since 2021. HOTA achieves a proper balance between accurate detection, association, and localization. In contrast to other evaluation metrics, HOTA gives equal weight to both detection and association. It is intended to calculate an overall score that accounts for both detection and association. Through HOTA, we can gain a better understanding of the underlying behavior of trackers and compare them effectively. Additionally, we also consider other evaluation metrics when appropriate, including sMOTSA, MOTA, IDF1, DetA, AssA, and LocA, to evaluate different aspects of tracker performance.

3 Datasets for pre-training

COCO [5]. MS COCO dataset is a large-scale dataset that can be utilized for various tasks, such as object detection, segmentation, key-point detection, and captioning. The dataset contains 118k training images (train2017), and the detection annotations are utilized in our experiment to pretrain our network.

CrowdHuman [8]. The CrowdHuman dataset contains 15k images with rich human instance annotations for training. The dataset features a variety of kinds of occlusions, with an average of 23 subjects per image. Three types of annotations are applied to each human instance in the image: a head bounding box, a human visible-region bounding box, and a human full-body bounding box. The human full-body bounding-box annotation is used in our experiment to possess the same setting as MOT17 [7].

4 Pre-training experiments

With respect to the KITTI MOTS [10] task, we use the pre-trained weights provided by CenterNet [11] as our backbone detector DLA-34 with the COCO dataset pre-training. For the MOT17 [7] task, we modify the settings in CenterTrack [12] and pre-train our network using the static images of the CrowdHuman dataset [8]. Parameters used are input resolution 512×512, random scaling ratio 0.05, and random translation ratio 0.05. These parameters are used to simulate the previous frame by randomly scaling and translating the current frame for generating the tracking effect.

5 Inference

With the dynamic weights $\theta_i^{t-\delta}$ stored in the detected object, the customized searcher generates the response map $\mathbf{R}_i^t$. The peak location $\mathbf{p}$ of the map indicates the center of the candidate object b_i^t. By looking up the size map S of the detection branch, we obtain the size $S_\mathbf{p}$ of the object's bounding box. Now, we have the complete bounding box of the object b_i^t. After finding all bounding boxes for all detected objects $b_i^{t-\delta}$, we collect all boxes into the set T^t. At the same time, the detection branch outputs a set of bounding boxes $\tilde{T}^t$. Our goal is to match boxes between $\tilde{T}^t$ from the detection branch and T^t from the search (tracking) branch. This is achieved by computing spatial distances (IoU) between boxes and association confidence v_i^t using a greedy algorithm. Following the above procedure, we define a continuous trajectory if the detection confidence w of the object is above γ. A new trajectory will be created for a non-matched detection with confidence w above κ.

For tracking objects which disappear for a while, probably due to occlusion, the tracker maintains a tracker pool which contains all alive objects' dynamic weights. An object will be removed from the tracker pool if it is not matched for τ consecutive frames. New objects will be added, and matched objects will be updated in the tracker pool.

6 Implementation details

We use a variant of DLA-34 with deformable convolution layer [1] proposed in CenterNet [11] as our backbone. The optimizer is Adam [4] with the learning rate as $1.25e-4$ and the batch size as 12. Data augmentations include random horizontal flipping, random resized

C_{search}	HOTA↑	DetA↑	AssA↑
8	57.9 %	56.8 %	59.4 %
16	**59.4** %	56.0 %	**63.5** %
32	57.2 %	**57.2** %	57.4 %

Table 1: **Evaluation on the output dimension of the search branch using the KITTI MOTS validation set.**

τ	MOTA↑	IDF1↑	IDS↓
1	66.2	64.0 %	610
5	66.8	67.3 %	**366**
15	66.8	69.1 %	394
30	**66.9**	**69.7** %	388
45	66.9	69.5 %	395

Table 2: **Results on MOT17 that trajectories are removed from tracker pool after they unseen within τ frames.** The tracking performance increases with τ in $IDF1$ and saturates after τ is larger than 30.

cropping, and color jittering. The model parameters are pre-trained on the COCO dataset [5] for the KITTI MOTS benchmark and pre-trained on CrowdHuman [8] with static images for the MOT17 benchmark. For all experiments, we train the network for 140 epochs. The learning rate is decayed by a factor of 10 at the 80th and 110th epochs. We measure the runtime on a machine with an Intel Core Silver 4110 @ 2.3 GHz and NVIDIA Tesla V100S.

For the KITTI dataset [10], the resolution of input images remains 1280×384 in the training and testing phases. We set the continued trajectory threshold of object association confidence to $\gamma = 0.4$ and the newly generated trajectory threshold to $\kappa = 0.4$. The trajectory will be removed if it is not visible for $\tau = 9$ consecutive frames.

For the MOT17 dataset [7], the resolution of input images is set to 960×544 with resizing and padding. The hyperparameter for maintaining the continued trajectory is set to $\gamma = 0.4$, and the threshold of the newly generated trajectory to $\kappa = 0.3$. Also, the trajectory will be removed if it is not visible for $\tau = 30$ consecutive frames. The number of channels is $C_{search} = 16$ in the search branch for both tasks.

7 More ablation studies

The output dimension of the search branch. We analyze the impact of the output dimension C_{search} for the search branch. As shown in Table 1, the association accuracy improves as C_{search} increases from 8 to 16, but it drops when increasing from 16 to 32. Empirically, $C_{search} = 16$ gives the best HOTA and AssA, and we take it as the default setting for all experiments. The choices of C_{search} also relate to the number of parameters of the dynamic searcher. When $C_{search} = 16$, the number of parameters is 593.

Inference of SearchTrack Like CenterTrack [12] and SiamMOT [9], we focus on improving tracking as long as the instance is visible. However, the instance (e.g., a person) may be invisible due to occlusion, especially in crowded scenes. Therefore, SearchTrack maintains a trajectory pool that keeps unmatched trajectories for τ consecutive frames. In Table 2, we evaluate the impact of τ on MOT17, in which most videos are captured at 30 FPS. It can be observed that tracking performance improves with τ in IDF1 metrics, and it saturates after τ is larger than 30. Thus, our tracker keeps a trajectory in the pool as long as it does not disappear longer than one second.

References

[1] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[3] Arne Hoffhues Jonathon Luiten. Trackeval. https://github.com/JonathonLuiten/TrackEval, 2020.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, 2014.

[6] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 2020.

[7] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[8] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[9] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. Siammot: Siamese multi-object tracking. In *CVPR*, 2021.

[10] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *CVPR*, 2019.

[11] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[12] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, 2020.