SFTrack++: A Fast Learnable Spectral Segmentation Approach for Space-Time Consistent Tracking – Pre-registration workshop –

Elena Burceanu*
Bitdefender
University of Bucharest, Romania
Institute of Mathematics of the Romanian Academy
eburceanu@bitdefender.com

Abstract

We propose an object tracking method, SFTrack++, that smoothly learns to preserve the tracked object consistency over space and time dimensions by taking a spectral clustering approach over the graph of pixels from the video, using a fast 3D filtering formulation for finding the principal eigenvector of this graph's adjacency matrix. To better capture complex aspects of the tracked object, we enrich our formulation to multi-channel inputs, which permit different points of view for the same input. The channel inputs could be, like in our experiments, the output of multiple tracking methods or other feature maps. After extracting and combining those feature maps, instead of relying only on hidden layers representations to predict a good tracking bounding box, we explicitly learn an intermediate, more refined one, namely the segmentation map of the tracked object. This prevents the rough common bounding box approach to introduce noise and distractors in the learning process. We test our method, SFTrack++, on seven tracking benchmarks: VOT2018, LaSOT, TrackingNet, GOT10k, NFS, OTB-100, and UAV123.

1 Introduction

Better using the temporal aspect of videos in visual tasks has been actively discussed for a rather long time, especially with the large and continuous progress in hardware. The first aspect we tackle in our approach is a seamless blending of space and time dimensions in visual object tracking. Current methods mostly rely on target appearance and frame-by-frame processing [1, 2, 3], with rather few taking explicit care of temporal consistency [4, 5]. In the spectral graph approach, nodes are pixels and edges are their local relations in space and time, while the strongest cluster in this graph, given by the principal eigenvector of the graph's adjacency matrix, represents the consistent main object volume over space and time.

A second observation challenges the rough bounding box (bbox) shape used for tracking. While it provides a handy way to annotate datasets, it is a rather imperfect label since it leads to errors that accumulate, propagate, and are amplified over time. Objects rarely look like boxes, and bboxes contain most of the time significant background information or distractors. Since having a good segmentation for the interest object directly influences the tracking performance, we constrain an intermediary representation, a segmentation map, which aims to reduce the quantity of noise transferred from a frame to the next one. We integrate it into our end-to-end flow, as shown in Fig. 1.

A third point we emphasize on is relying on multiple, independent characteristics of the same object or multiple modules specialized in different aspects. This comes with an improved ability to understand

*ilarele.github.io

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

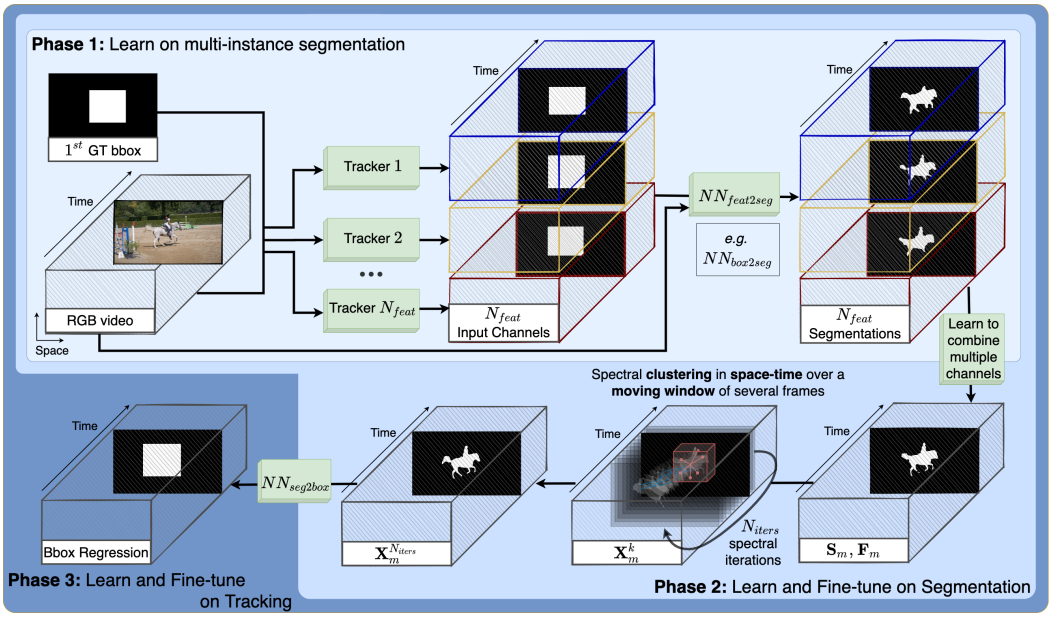

Figure 1: SFTrack++: We start from video's RGB and 1^{st} frame GT bbox of the tracked object. We run state-of-the-art trackers, in an online manner, while fine-tuning $NN_{feat2seg}$ network frame-by-frame (pretrained in Phase 1) to transform the extracted feature maps (*e.g.* bboxes) to segmentation maps. Next, we learn to combine multiple segmentation inputs and refine the final mask using a spectral approach, applied also online over a moving window containing the previous N frames, for N_{iter} spectral iterations (Phase 2). In Phase 3, we learn a bbox regressor from the final segmentation mask, $NN_{seg2box}$ and fine-tune all our parameters on the tracking task.

complex objects while increasing the robustness [4, 6]. We, therefore, adjust the SFSeg [7] spectral approach, enhancing its formulation to support learning on top of multiple input channels. They could be general features for the input frame coming from different approaches or, more specifically, tracking outputs from multiple solutions, as we test in Sec. 4.

The **main contributions** of our approach are:

- SFTrack++ brings to tracking a natural, contiguous, and efficient approach for integrating space and time components, using a fast 3D spectral clustering method over the graph of pixels from the video, to strengthen the tracked object's model.
- We explicitly learn intermediate fine-grained segmentation as opposed to rough bounding boxes in our three phases end-to-end approach to a more robust tracking solution.
- We integrate into our formulation a way of learning to combine multiple input channels, offering a wider view of the objects, harmonizing different perceptions, for a powerful and robust approach.

2 Related work

General Object Tracking. Out of the three main trackers families, **Siamese based trackers** gained a lot of traction in recent years for their high speed and end-to-end capabilities [8, 9, 2, 10]. Most approaches focus on exhaustive offline-training, failing to monitor changes w.r.t. the initial template [11, 12, 13], while others update their model online [14, 15]. Nevertheless, the robustness towards unseen objects and transformations at training time remains a fundamental problem for Siamese trackers. **Meta-learning approaches for tracking** [16, 17, 18] come with an interesting way of adapting to the current object of interest, while keeping a short inference time, by proposing a target-independent tracking model. One major limitation for both those approaches, Siamese and meta-learning trackers, is that they fail to adapt continuously to the real-time changes in the tracked object, using rather a history of several well-chosen patches or even just the initial one. In contrast,

our method naturally integrates the temporal dimension, by continuously enforcing the local temporal and spatial object consistency. **Discriminative methods** [3, 19, 20] on the other hand are classic approaches, focusing more on changes in the tracked object [21] (background, distractors, hard negatives), better integrating the temporal dimension [4] in the method flow. They prove to be robust, but they mostly rely on hand-crafted observations or modules not trainable end-to-end. SFTrack++ provides an end-to-end approach while minimizing the distractors and background noise using the intermediary segmentation map. Our method distances itself from a certain family of trackers, introducing the space and time consistency endorsement via clustering component as an additional dimension of the algorithm.

With a few notable exceptions [22, 23], most tracking solutions use internally hidden layer representations extracted from the previous frame's rough bbox prediction [4, 19, 3, 12], rather than a fine-grained segmentation mask as in our approach. Also, most of them do not take into account multiple perceptions for the input frame and operate over a unique feature extractor [3, 10, 8]. There are a few trackers though that combine two models for adapting to sudden changes while remaining robust to background noise, by explicitly model the different pathways [4, 6]. In contrast, our end-to-end multi-channel formulation learns over 10 input channels, a significantly larger number.

Graph representations. Images and videos were previously represented as graphs, where the nodes are pixels, super-pixels, or regions [24]. This choice directly impacts the running time and performance. Regarding edges, they are usually undirected, modeled by symmetric similarity functions, but there are also several works that use directed ones [25, 26]. **Spectral clustering** approaches [27, 28, 29] search for the leading or the smallest eigenvector for the graph's adjacency matrix to solve the clusters' assignments. Spectral clustering was previously used in pixel-level image segmentation [30], with a high burden on the running time and in building space-time correspondences between video patches [24]. Graph Cuts is a common approach for spectral clustering, having many variations [30, 31, 32]. SFSeg [7] proposes a 3D filtering technique for efficiently finding the spectral clustering solution without explicitly computing the graph's adjacency matrix. Inspired by this method, we integrate an improved version with learning over multi-channel inputs, as an intermediate component in our tracker, as detailed in Sec. 3.

3 Our approach

SFTrack++ algorithm has three phases, as we visually present them in Fig. 1. In **Phase 1**, we learn a neural net, $NN_{feat2seg}$, that transforms the RGB and a frame-level feature map extracted using a tracker (*e.g.* bbox from a tracker prediction) into a segmentation mask. Using only the RGB as input is not enough, because frames can contain multiple objects and instances, and we also need a pointer to the tracked object to predict its segmentation. Next, in **Phase 2**, we run multiple state-of-the-art trackers frame-by-frame over the input as an online process and extract input channels from them (*e.g.* bboxes). We transform those feature maps to segmentation maps with the previously recalled module, $NN_{feat2seg}$. Next, we learn to combine and refine the outputs for the current frame using a spectral solution for preserving space-time consistency, adapted to learn over multiple channels. Note that, when applying the spectral iterations, we use a sliding window approach over the previous N frames in the video volume. For supervising this path, we use segmentation ground-truth. **Phase 3** learns a neural net as a bbox regressor over the final segmentation map from the previous phase, $NN_{seg2box}$, while fine-tuning all the other trainable parameters in the model, using tracking GT.

Spectral approach to segmentation. We go next through the following aspects, briefly explaining the connection between them: segmentation $\rightarrow$ leading eigenvector $\rightarrow$ power iteration $\rightarrow$ 3D filtering formulation $\rightarrow$ multi-channel. Image segmentation was previously formulated as a graph partitioning problem, where the segmentation solution [30] is the leading eigenvector of the adjacency matrix. It was used in a similar way for video [7]. Power iteration algorithm can compute the leading eigenvector: $\mathbf{x}_i^{k+1} \leftarrow \sum_{j \in \mathcal{N}(i)} \mathbf{M}_{i,j} \mathbf{x}_j^k$, where $\mathbf{M}$ is the $N \times N$ graph's adjacency matrix, N is the number of nodes in the graph (pixels in the video space-time volume in our case), $\mathcal{N}(i)$ is the space-time neighbourhood of node i and each step k is followed by normalization. The adjacency matrix used in power iteration usually depends on two types of terms: unary ones are about individual node properties and pairwise ones describe relations between two nodes (pairs).

Following this approach, SFSeg [7] rewrites power iteration using 3D filtering for an approximated adjacency matrix. The solution is described next in Eq. 1:

$$\mathbf{X}^{k+1} \leftarrow \text{normalized}(\mathbf{S}^p \cdot (\alpha^{-1}\mathbf{1} - \mathbf{F}^2) \cdot G_{3D} * (\mathbf{S}^p \cdot \mathbf{X}^k) - \mathbf{S}^p \cdot G_{3D} * (\mathbf{F}^2 \cdot \mathbf{S}^p \cdot \mathbf{X}^k) + 2\mathbf{S}^p \cdot \mathbf{F} \cdot G_{3D} * (\mathbf{F} \cdot \mathbf{S}^p \cdot \mathbf{X}^k)), \quad (1)$$

where $*$ is a 3D convolution with Gaussian filter G_{3D} over space-time volume, $\cdot$ is an element-wise multiplication, $\mathbf{S}$ and $\mathbf{F}$ are unary and pairwise terms in matrix form with p and α controlling their importance, k is the current spectral iteration and $\mathbf{X}, \mathbf{S}, \mathbf{F}$ matrices have the original video shape ($N_{frames} \times H \times W$).

Multi-channel learning formulation. We extend the single-channel formulation in SFSeg such that it can learn how to combine several input channels, S_i and F_i, for unary and pairwise terms respectively: $\mathbf{S}_m \leftarrow \sigma(\sum_{i=1}^{N_{cs}} w_{s,i}\mathbf{S}_i + b_s\mathbf{1})$, $\mathbf{F}_m \leftarrow \sigma(\sum_{i=1}^{N_{cf}} w_{f,i}\mathbf{F}_i + b_f\mathbf{1})$, where $\mathbf{S}_m$ and $\mathbf{F}_m$ are the multi-channel unary and pairwise maps, respectively, σ is the sigmoid function, N_{cs} and N_{cf} are the number of input channels, $\mathbf{1}$ is an all-one matrix for the bias terms and $w_{s,i}, w_{f,i}, b_s, b_f$ are their corresponding learnable weights. We replace $\mathbf{S}$ and $\mathbf{F}$ in Eq. 1 with their multi-channel versions $\mathbf{S}_m$ and $\mathbf{F}_m$, respectively. We learn $w_{s,i}, w_{f,i}, b_s, b_f$ parameters both over segmentation and tracking tasks. More, SFTrack++ can learn end-to-end, from the original input frames all the way to final output, in the case of end-to-end learnable feature extractors.

4 Experimental protocol

We test if SFTrack++ brings in a complementary dimension to tracking by having an intermediary fine-grained representation, extracted over multiple state-of-the-art trackers' outputs, and smoothed in space and time. We guide our experiments such that we evaluate the least expensive pathways first. For reducing the hyper-parameters search burden, we use AdamW [33], with a scheduler policy that reduces the learning rate on a plateau. For efficiency and compactness, we use the same channels to construct both the unary and pairwise maps: $\mathbf{S}_i = \mathbf{F}_i$. Their learned weights are also shared $w_{s,i} = w_{f,i}$. We use as input channels bboxes extracted with top single object trackers. We choose 10 top trackers: SiamR-CNN [14], LTMU [15], KYS [4], PrDiMP [19], ATOM [3], Ocean [10], D3S [20], SiamFC++ [12], SiamRPN++ [2], SiamBAN [13], which differ in architecture, training sets and overall in their approaches, but all achieves top results on tracking benchmarks.

Training. In Phase 1 we train our $NN_{feat2seg}$ network on DAVIS-2017 [34] and Youtube-VIS [35] trainsets, for each individual object. It receives the current RGB and the output of a tracking method (bbox), randomly sampled at training time. We use the U-Net architecture, validating the right number of parameters (100K - 1 mil) and the number of layers. We use DAVIS-2017 and Youtube-VIS evaluation sets to stop the training. Following the curriculum learning approach, before introducing tracking methods intro the pipeline, we use at the beginning of the tracking GT bboxes (extracted from segmentation GT, as straight bboxes). This allows the $NN_{feat2seg}$ component to get a good initialization, before introducing faulty bbox extractors, namely the top 10 tracking methods mentioned before. For Phase 2, we train on the segmentation task. We learn the second part of our method to have an intermediary fine-grained representation, extracted over multiple channels, and smoothed in space and time. We validate here N_{iters}, the number of spectral iterations (1-5). We train on DAVIS-2016 [36] and Youtube-VIS datasets. In Phase 3, training for tracking, we learn a regression network, $NN_{seg2box}$ (with 50K-500K parameters), to transform the final segmentation to bbox. We train on TrackingNet [37], LaSOT [38] and GOT-10k [39] training splits.

Baselines Comparison. Experiment for comparing with other methods focus on the improvements SFTrack++ could bring over state-of-the-art and other competitive approaches for general object tracking: single method state-of-the-art solutions, a basic ensemble over the trackers, SFTrack++ applied only over the best tracker, SFTrack++ applied over the basic ensemble and the best learned neural net ensemble we could get out of several configurations (2D and 3D versions for U-Net [40] and shallow nets, having a different number of parameters: 100K, 500K, 1 mil, 5 mil, 15 mil). All methods receive the same input from top 10 trackers and train on TrackingNet, LaSOT and GOT-10k train sets as previously described. We evaluate our solution against all baselines on seven tracking benchmarks: **VOT2018** [41], **LaSOT**, **TrackingNet**, **GOT-10k**, **NFS** [42], **OTB-100** [43] and **UAV123** [44]. For the main conclusion of the paper, we will provide statistical results (mean and variance over several runs) to better indicate a strong positive/negative result, or an inconclusive one.

Ablative studies. We vary several components of our end-to-end model to better understand their role and power. We train our **Phase 1** component, $NN_{feat2seg}$ net, not only for bbox input features but also for other earlier features, extracted from each tracker architecture. We test the overall tracking performance for this case. We remove from the pipeline the spectral refinement in **Phase 2** and report the results. We test the performance of our tracker without the **Phase 3** neural net, $NN_{seg2box}$, by replacing it with a straight box and rotated box extractors from OpenCV [45]. We test several losses to optimize for both segmentation and tracking tasks: a linear combination between the weighted diceloss [46] and binary cross-entropy, Focal-Tversky [47], and Focal-Dice [48]. For the ablative experiments, we evaluate only on OTB-100, UAV123, and NFS tracking datasets.

References

[1] S. Caelles, K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. *CVPR*, 2017.

[2] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

[3] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: accurate tracking by overlap maximization. In *CVPR*, 2019.

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, 2020.

[5] Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. *CoRR*, 2020.

[6] Elena Burceanu and Marius Leordeanu. Learning a robust society of tracking parts using co-occurrence constraints. In *ECCV Workshops*, 2018.

[7] Elena Burceanu and Marius Leordeanu. A 3d convolutional approach to spectral object segmentation in space and time. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 495–501. ijcai.org, 2020.

[8] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In Gang Hua and Hervé Jégou, editors, *ECCV Workshops*, 2016.

[9] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[10] Zhipeng Zhang and Houwen Peng. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020.

[11] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders. Siamese instance search for tracking. In *CVPR*, 2016.

[12] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, 2020.

[13] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, 2020.

[14] Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: visual tracking by re-detection. In *CVPR*, 2020.

[15] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In *CVPR*, 2020.

[16] Eunbyung Park and Alexander C. Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. In *ECCV*, 2018.

[17] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip H. S. Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *NIPS*, 2016.

[18] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In *CVPR*, 2020.

[19] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020.

[20] Alan Lukezic, Jiri Matas, and Matej Kristan. D3S - A discriminative single shot segmentation tracker. In *CVPR*, 2020.

[21] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, 2017.

[22] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H. S. Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.

[23] 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.

[24] Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. *CVPR*, 2020.

[25] Andrea Torsello, Samuel Rota Bulò, and Marcello Pelillo. Grouping with asymmetric affinities: A game-theoretic perspective. In *CVPR*, 2006.

[26] Stella X. Yu and Jianbo Shi. Grouping with directed relationships. In *EMMCVPR 2001*, 2001.

[27] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[28] Marina Meila and Jianbo Shi. A random walks view of spectral segmentation. In *AISTATS*, 2001.

[29] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. *ICCV*, 2005.

[30] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *PAMI*, 2000.

[31] Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *ICDM*, 2001.

[32] Sudeep Sarkar and Padmanabhan Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *PAMI*, 2000.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[34] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[35] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019.

[36] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[37] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.

[38] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019.

[39] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[41] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman P. Pflugfelder, Luka Cehovin Zajc, Tomás Vojír, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, Gustavo Fernández, and et al. The sixth visual object tracking VOT2018 challenge results. In *ECCV 2018 Workshops*, 2018.

[42] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, 2017.

[43] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.

[44] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, 2016.

[45] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[46] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *MICCAI*, 2017.

[47] Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation. In *ISBI*, 2019.

[48] Pei Wang and Albert C. S. Chung. Focal dice loss and image dilation for brain tumor segmentation. In Danail Stoyanov, Zeike Taylor, Gustavo Carneiro, Tanveer F. Syeda-Mahmood, and et al., editors, *MICCAI*, 2018.