Generating Masks from Boxes by Mining Spatio-Temporal Consistencies in Videos

Supplementary Material

Bin Zhao Goutam Bhat Martin Danelljan Luc Van Gool Radu Timofte
Computer Vision Lab, D-ITET, ETH Zurich, Switzerland
{bzhao,goutam.bhat,martin.danelljan,vangool,radu.timofte}@ethz.ch

In this supplementary material, we provide additional details and analysis of our approach. In Section 2, we provide details about the network architecture of our object encoder module. Section 3 provides additional analysis of the proposed approach. Details about the inference setting used to annotate the tracking datasets in Section 4.3 of the main paper are provided in Section 4. Detailed results on the GOT10k validation set are provided in Section 5. We evaluate our approach on the YouTube-VOS 2018 validation set in Section 6. Section 7 includes additional qualitative results on the DAVIS and GOT10k datasets. Additionally, we also include a video showing the results of our approach on the sequences from the DAVIS and GOT10k datasets.

1. Included video

We provide a video showing the masks produced by our approach for sequences from DAVIS and GOT10k datasets. Our approach generates high-quality masks even under difficult circumstances, such as fast motions, appearance change and shape change. The last sequence, *bike-packing* from DAVIS, shows a particularly challenging case where two objects are highly overlapping.

2. Structure of the object encoder

Here, we describe in detail the network structure employed for the object encoder, as shown in Fig. 1. Note that our object encoder is formulated as $(e_t, w_t, m_t) = B(x_t, b_t)$. The network first takes as input the mask representation of the bounding box b_t and then passes it to a convolution layer, a max pooling layer and two residual blocks. The intermediate mask features are then concatenated with deep features x_t and fed through another residual block, which reduces the feature dimension. Finally, two similar heads are utilized to produce abstract embedding e_t and weight w_t. Although the embedding and weight heads share common network to extract object representations, there is no need to share them for the single-frame

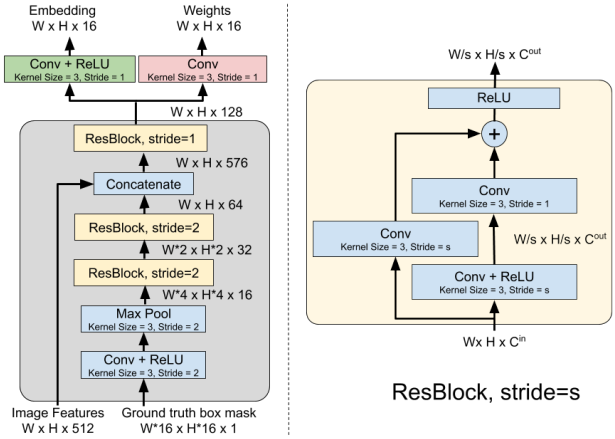

Figure 1. Architecture of object encoder.

encoding m_t. Single-frame encoding m_t is directly passed to the decoder and has no connection with embedding e_t. Thus, we use a different network to obtain the single-frame encoding m_t. This network has the same architecture as the network used to obtain e_t and w_t, with the only difference that it has a single head with a convoluton and ReLU layers to predict e_t.

3. Detailed analysis

Here, we provide a more detailed analysis of our approach for predicting object masks from bounding boxes in videos.

Impact of the number of steepest-descent iterations: We perform 5 steepest-descent algorithm iterations during training in order to save training time and speed up convergence. However, there is no need to only iterate 5 times during inference. We perform experiments to analyse the impact of iterating more times and give results in Tab. 1. Increasing the number of iterations from 5 to 15, the $\mathcal{J}$ score increases by 1.2 on DAVIS2017 validation set. The perfor-

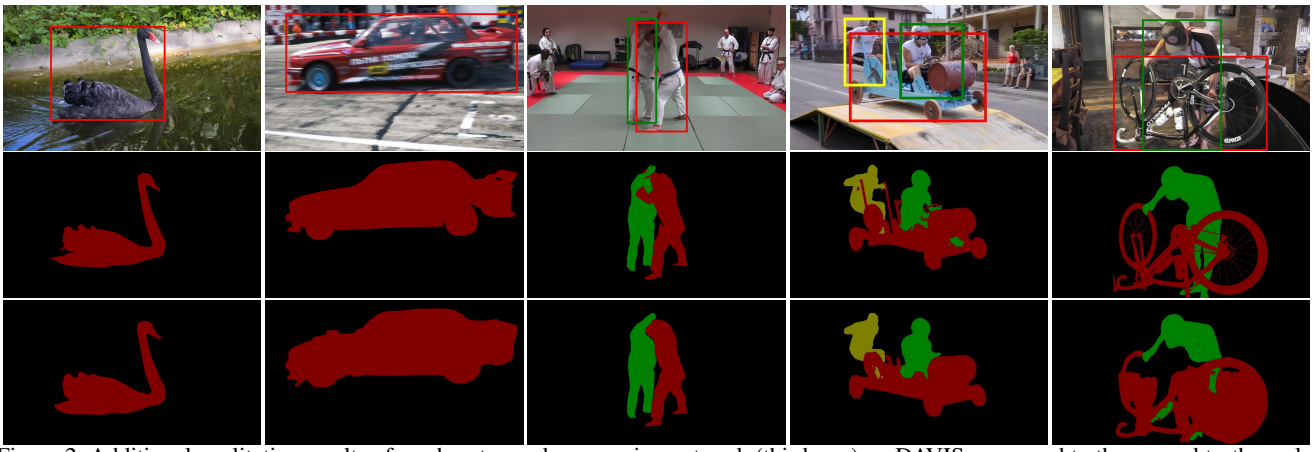

Figure 2. Additional qualitative results of our box to mask conversion network (third row) on DAVIS compared to the ground truth masks (second row).

Figure 3. Additional qualitative results of our box to mask conversion network on GOT10k.

Num. iterations	5	10	15	20
YT300	85.4	85.6	85.6	85.6
DAVIS2017 val	80.0	80.9	81.2	81.2

Table 1. Impact of the number of steepest-descent iterations. Results are shown in terms of Jaccard $\mathcal{J}$ index.

Sample scale relative to object	2	3	4	5
YT300	84.5	85.7	85.6	85.5
DAVIS2017 val	80.2	80.9	81.2	80.8

Table 2. Impact of the image sample size relative to the object bounding box. Results are reported in terms of Jaccard $\mathcal{J}$ score.

mance of our approach saturates when iterating more than 15 times on YouTube-VOS and DAVIS.

Impact of the sample size: During training, we crop a patch 5 times larger than the ground truth bounding box to exploit background consistency. However, we find that it may be harmful to include too much unnecessary environment information. Therefore, we perform experiments to analyse the results if we crop a smaller patch. The size of the crop is set to different scaling factors relative to the object bounding box size. The results on YT300 and DAVIS2017 validation set are shown in Tab. 2. There is a significant improvement of $+1.2$ and $+0.7$ in J score on YT300 and DAVIS when increasing the sample scale from 2 to 3, meaning including more background information leads to better results. We achieve best results when setting the sample scale as 3 on YouTube-VOS and 4 on DAVIS. A larger patch will contain too much noisy environment information, leading to degradation in performance. We set sample scale to 4 during inference in all our experiments.

Impact of the inter-frame interval: Here, we analyse the inter-frame interval used for DAVIS and YouTube-VOS. Results in terms of Jaccard $\mathcal{J}$ score are shown in Tab. 3 and Tab. 4. Compared to using consecutive frames, selecting every 5th frame gives an improvement of $+0.7$ in $\mathcal{J}$ score on DAVIS2017 validation set. The performance will decrease if we use larger inter-frame interval. While on YouTube-VOS, since the sequence is annotated every 5 frames, the minimal inter-frame interval that can be used is 5. There is no substantial difference of using different inter-frame intervals on YT300, so we generally set a value 15 for other experiments.

Performance w.r.t. quality of boxes: Here, we analyze

Inter-frame interval	1	5	10	15
DAVIS2017 val	80.5	81.2	80.9	80.5

Table 3. Impact of the inter-frame interval on DAVIS. Results are shown in terms of Jaccard $\mathcal{J}$ index.

Inter-frame interval	5	10	15	20	25
YT300	85.6	85.7	85.6	85.6	85.5

Table 4. Impact of the inter-frame interval on YT300. Results are shown in terms of Jaccard $\mathcal{J}$ index.

the impact of input bounding box accuracy on segmentation performance. We manually add Gaussian noise to the bounding box co-ordinates input to our network. The standard deviation of the Gaussian is set to *noise_level* times the object size. We evaluate the performance of our approach for different *noise_level* parameters. For each noise level, we train a separate model using identical noise level during training. The results of this analysis on YT300 and DAVIS is shown in Table 5. Adding Gaussian noise with a standard deviation of 1% the target size to the bounding box decreases accuracy by only $0.7\,\mathcal{J}$ on YT300 and $1.0\,\mathcal{J}$ on DAVIS.

4. Additional inference details

We describe details about how we annotate large-scale tracking datasets LaSOT and GOT10k. Generally, we use the same inference setting as used for YouTube-VOS, except for the number of frames. We select 5 frames for each testing frame in order to save inference time and ensure the high performance at the same time. We annotate every frame of the sequence on GOT10K, while on LaSOT, we only annotate every 5th frame. LaSOT contains very long sequences and the objects generally move slowly. There is no need to annotate adjacent frames because they are highly correlated. Moreover, we only annotate up to 200 frames from a video sequence to avoid generating too much data for the same object.

5. Detailed Results on GOT10k

In this section, we provide success plots on the GOT10k validation set. The success plots are obtained using the Overlap Precision (OP) score. The OP score at a threshold τ denotes the fraction of frames in which the intersection-over-union (IoU) overlap between the tracker prediction and the ground truth box is greater than τ. The OP scores for a range of thresholds in $[0, 1]$ are plotted to obtain the success plots. The trackers are ranked according the the average overlap (AO) score, which is computed as the average IoU overlap between the tracker prediction and the ground truth box over all frames in the dataset. The LWL model trained using our weakly annotated tracking data (LWL-Ours) obtains the best AO score, outperforming the standard LWL model 2.1%.

noise_level	0	0.005	0.01	0.02	0.03
YT300	85.6	85.0	84.9	85.0	84.4
DAVIS2017 val	81.2	80.4	80.2	80.6	79.6

Table 5. Impact of inaccuracies in the input bounding box on segmentation accuracy in terms of $\mathcal{J}$ index.

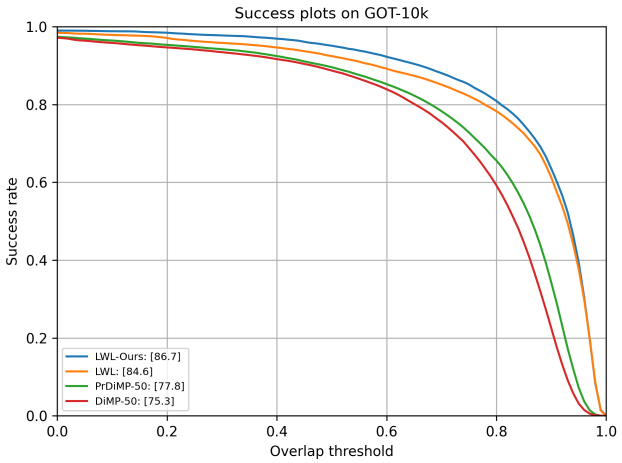

Figure 4. Success plots on the GOT10k validation set. The trackers are ranked using the average overlap (AO) score in percentage.

6. Results on YouTube-VOS

Here, we evaluate the LWL-Ours model trained using our pseudo-labelled tracking datasets, in addition to the YouTube-VOS and DAVIS datasets, on the YouTube-VOS 2018 validation set. A comparison with the state-of-the-art approaches is provided in Tab. 6. LWL-Ours obtains comparable results with standard LWL and outperforms other VOS methods significantly.

	OSVOS [1]	OnAVOS [4]	PreMVOS [2]	SiamRCNN [5]	STM [3]	LWL	**LWL-Ours**
$\mathcal{J}\&\mathcal{F}$ mean	58.8	55.2	66.9	73.2	79.4	81.5	80.9
$\mathcal{J}_{seen}$	59.8	60.1	71.4	73.5	79.7	80.4	79.6
$\mathcal{J}_{unseen}$	54.2	46.1	56.5	66.2	72.8	76.4	75.7
$\mathcal{F}_{seen}$	60.5	62.7	-	-	84.2	84.9	83.9
$\mathcal{F}_{unseen}$	60.7	51.4	-	-	80.9	84.4	84.3

Table 6. Comparison on YouTube-VOS 2018 validation set.

7. Qualitative results

We shown more qualitative results on DAVIS and GOT10k in Fig. 2. and Fig. 3, respectively. Compared to the ground truth masks on DAVIS, our approach gives high-quality results for frames containing only a single object.

For frames containing multiple objects, our approach can still delineate object boundary accurately if the overlapping problem is not severe.

References

[1] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 221–230, 2017.

[2] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580. Springer, 2018.

[3] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9226–9235, 2019.

[4] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017.

[5] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6588, 2020.