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

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

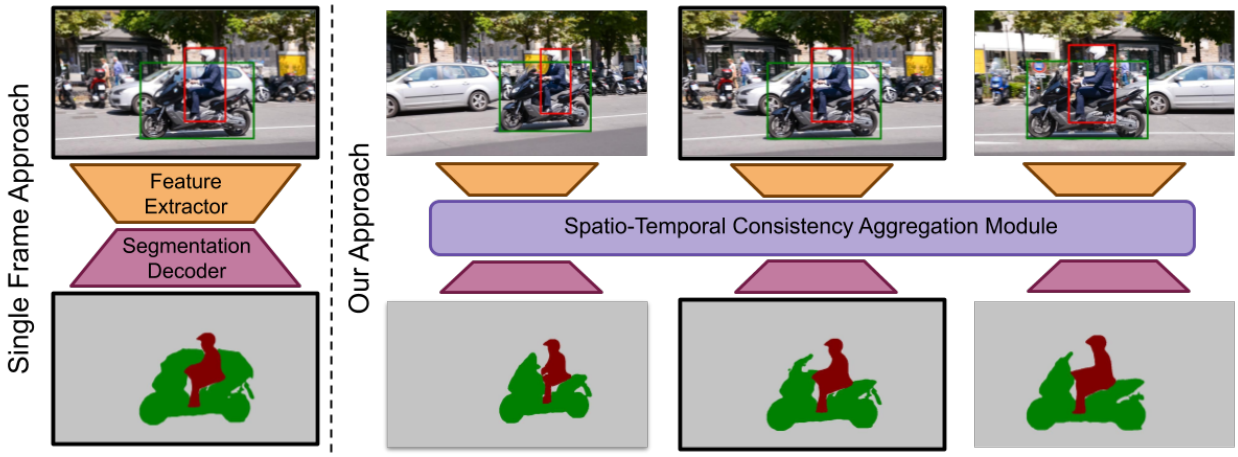

Figure 1. When only using a single frame, predicting object masks from bounding boxes often leads to failures (left), since the outline of an object is difficult to resolve. By using video, our approach aggregates information across several frames. In this example, it identifies the car as background through the neighboring frames, while the scooter remains within the box, allowing it to be accurately segmented.

Abstract

Segmenting objects in videos is a fundamental computer vision task. The current deep learning based paradigm offers a powerful, but data-hungry solution. However, current datasets are limited by the cost and human effort of annotating object masks in videos. This effectively limits the performance and generalization capabilities of existing video segmentation methods. To address this issue, we explore weaker form of bounding box annotations.

We introduce a method for generating segmentation masks from per-frame bounding box annotations in videos. To this end, we propose a spatio-temporal aggregation module that effectively mines consistencies in the object and background appearance across multiple frames. We use our predicted accurate masks to train video object segmentation (VOS) networks for the tracking domain, where only manual bounding box annotations are available. The additional data provides substantially better generalization performance, leading to state-of-the-art results on standard tracking benchmarks. The code and models are available at https://github.com/visionml/pytracking.

1. Introduction

Segmenting objects in videos is an important but challenging task with many applications in autonomous driving [51, 55], surveillance [11, 15] and video editing. The field has been driven by the astonishing performance of deep learning based approaches [7, 44, 58]. However, these methods require large amount of training images with pixel-wise annotations. Manually annotating segmentation masks in videos is an extremely time-consuming and costly task. Existing video datasets with segmentation labels [49, 63] therefore do not provide the large-scale diversity desired in deep learning. This effectively limits the potential of current state-of-the-art approaches.

To address this issue, it is tempting to consider weaker forms of human annotations. In particular, object bounding boxes offer an interesting alternative. Boxes provide horizontal and vertical constraints on the extent of the segmentation mask, while also being substantially faster to annotate. Hence, a method of effectively leveraging bounding box annotations for training video segmentation models would greatly simplify the process of deploying these models for novel domains. Ideally, simply converting the video

box annotations to object masks would allow existing video segmentation approaches to integrate these annotations using standard supervised techniques, without requiring any modification of losses or architectures. Such a conversion network can itself be trained using available mask annotated data. We therefore investigate the problem of generating object segmentations from box-annotations in videos.

Generating masks from box-annotated videos is a deceptively challenging task. The background scene is often cluttered or contains similar objects. Objects can change appearance rapidly and often undergo heavy occlusions. Existing approaches [38, 60] only address the single frame case, where these ambiguities are difficult, or sometimes impossible to resolve due to the limited information. However, the aforementioned problems can be greatly alleviated if we can utilize multiple frames in the video. As the object moves relative to the background, we can find consistencies over several example views of the object and background. While object regions should consistently stay inside the box, background patches can move from inside to outside the object box over the duration of the video sequence. For instance, in Fig. 1 the single frame approach fails to properly segment the scooter due to the background car. In contrast, our video-based approach can identify the car as background in earlier and later frames while the scooter is consistently within the box for all frames. The car is therefore easily excluded from the final segmentation in all frames.

Effectively exploiting the information encoded in the temporal information is however a highly challenging problem. Since the object and background moves and changes in each frame, standard fusion operations cannot extract the desired consistencies and relations. Instead, we propose a spatio-temporal aggregation module by taking inspiration from the emerging direction of deep declarative networks [17]. Our module is formulated as an optimization problem that aims to find the underlying object representation that best explains the observed object and background appearance in each frame. It allows our approach to mine spatio-temporal consistencies by jointly reasoning about all image patches in all input frames. The resulting mask embeddings for each frame are then processed by a decoder to generate the final segmentation output.

Contributions: Our main contributions are as follows. **(i)** We propose a method for predicting object masks from bounding boxes in videos. **(ii)** We develop a spatio-temporal aggregation module that effectively mines the object and background information over multiple frames. **(iii)** Through an iterative formulation, we can further refine the masks through a second aggregation module. **(iv)** We utilize our method to annotate large-scale tracking datasets with object masks, which are then utilized to extend Video Object Segmentation (VOS) to the tracking domain.

We perform extensive experiments, demonstrating the effectiveness of our approach in the limited data domain. Moreover, we show that the data generated by our approach allows VOS methods to cope with challenges posed by the tracking setting. An existing VOS approach [7] trained on our pseudo-annotated tracking videos achieves state-of-the-art performance on standard tracking benchmarks, achieving an EAO score of 0.510 on VOT2020, and 86.7 AO on GOT-10k validation set. Code, models and generated annotations will be made publicly available.

2. Related Work

Semi-supervised video object segmentation: Semi-supervised video object segmentation (VOS) is the task of classifying all pixels in a video sequence into foreground and background, given a target ground-truth mask in the first frame. A number of different approaches have been proposed for VOS in recent years, including detection based methods [8, 40, 59], propagation based approaches [27, 35, 46, 62], feature matching techniques [10, 22, 44, 58], and meta-learning based methods [3, 7, 50]. A crucial factor that has enabled the recent advancements in VOS has been the release of high quality datasets such as DAVIS [49] and YouTube-VOS [63]. However, performing pixel-wise mask annotations for VOS datasets is an extremely time consuming task. As a result, VOS datasets are still relatively smaller in size and contain limited number of object classes, motion types, *etc*. compared to other fields such as object detection and tracking. This poses significant challenges in training general VOS models for real-world applications.

Weakly supervised segmentation: Due to the high cost of collecting pixel-wise labels, different types of weak annotations have been utilized to guide segmentation tasks recently, such as image-level supervision [1, 24, 29, 47, 66], points [2, 41], scribbles [37, 57] and bounding boxes [12, 21, 26, 31, 45]. Recent work [21] designs a multiple instance learning (MIL) loss by leveraging the tightness property of bounding boxes. Our work is more related to bounding box supervised segmentation. In [12, 26], pseudo segmentation masks for training are generated using GrabCut [52] and the MCG proposals [48]. Voigtlaender *et al*. [61] employed the box to mask conversion model introduced in [38] to generate pseudo segmentation masks for box annotated videos, using a single mask annotation per video. In a similar spirit to [61], our approach utilizes a pool of mask annotated videos to train a video box to mask conversion network, which is then used to generate pseudo-labels using box annotations. However, unlike in [61], our approach does not require any additional mask annotations when labelling box annotated videos. Moreover, the masks are generated by utilizing spatio-temporal consistencies across video frames

Converting boxes to segmentation masks: Generating

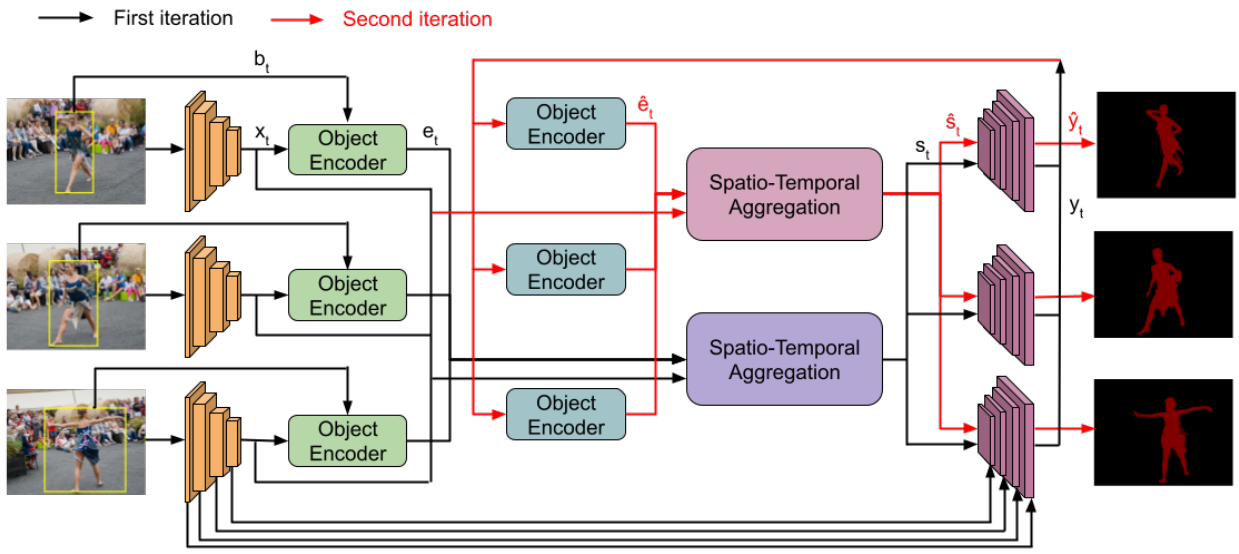

Figure 2. An overview of our architecture for segmenting an object from a box-annotated video. We extract deep features from each frame. Features x_t and boxes b_t are then given to the object encoder (Sec. 3.1) to generate an object-aware representation e_t. The spatio-temporal aggregation module (Sec. 3.2) inputs object encodings and deep features from all frames. Its output s_t is decoded to an object mask y_t. We refine the masks by iterating the process (Sec. 3.3) with a secondary object encoder and aggregation module to generate the final output $\hat{y}_t$.

a segmentation mask from the given object bounding box is an essential sub-task in instance segmentation, especially detection-based methods [13, 18, 19, 36]. These approaches follow a multi-task learning strategy where a backbone network first extracts deep features and generates a set of proposals. Then detection and segmentation heads are used separately to predict an accurate bounding box and segmentation mask for the proposal. ShapeMask [32] takes a bounding box detection as the initial shape estimate and refines it gradually, using a collection of shape priors. Luiten *et al*. [38] train a modified DeepLabv3 [9] model to output a mask, given a crop containing the object as input. In contrast to the previous approaches which only operate on single images, we address the task generating masks given box annotated videos as input. Our approach can exploit the additional temporal information in the video to predict more accurate masks, as compared to single image approaches.

Object Co-segmentation: Object co-segmentation is the task of segmenting the common objects from a set of images. This concept was first introduced by Rother *et al*. in [53], which minimizes an energy function containing an MRF smoothness prior and a histogram matching term. Subsequent work [54] combines visual saliency and dense SIFT matching to capture the sparsity and visual variability of the common object in a group of images. The work [34] integrates a mutual correlation layer into a CNN-based Siamese architecture to perform co-segmentation. Similar to co-segmentation method, we segment objects using multiple images. However, our images are obtained from the same video. This enables exploiting strong temporal consistency in videos to improve segmentation accuracy.

3. Method

We propose an end-to-end trainable architecture for the problem of segmenting an object in a video, given its bounding boxes in each frame. Our complete architecture is shown in Fig. 2. To fully exploit the temporal dimension, we aim to use detailed information of not only the target, but also the background context. Our backbone feature network F therefore first separately encodes video frames $\{I_t\}_1^T$ containing the object as well as substantial background. The extracted deep features $x_t = F(I_t)$ along with the corresponding bounding box b_t are given to the object encoder B, which provides an object-aware representation e_t of each individual frame. By integrating the object bounding box, it provides information about hypothetical object and background regions.

The object encodings and deep features x_t from all frames are input to the spatio-temporal aggregation module S. The goal of this module is to generate an encoding of the object segmentation s_t for each frame. The module S aggregates appearance and box information from all frames and locations through an efficient and differentiable optimization processes. The iterative procedure fuses the different observations of the appearance $\{(x_t, e_t)\}$ by find an underlying representation of the object. This representation then generates the segmentation encoding s_t, which is processed by the segmentation decoder D to predict preliminary object masks as $y_t = D(s_t, x_t)$. Our flexible architecture allows us to further improve the masks by feeding the results into a second spatio-temporal aggregation module, which predicts a set of refined segmentation encodings

$\hat{s}_t$. The final segmentation masks $\hat{y}_t = D(\hat{s}_t, x_t)$ are then generated by the same decoder network. Our entire architecture is trained end-to-end in a fully supervised manner. In the next section, we first detail our object encoder.

3.1. Object Encoder

Accurately segmenting a specified object given only a single frame is a challenging problem. Since our goal is generic object segmentation, the type of object specified during inference may not even be represented in the training set. In general, it is therefore difficult to assess which image region inside a single bounding box belongs to the object in question. This is further complicated by cluttered scenes or presence of distractor regions in the background that are similar to the object itself. In these cases, determining object boundaries is particularly difficult. Moreover, multiple objects often overlap, making even the decision of which object to segment given the bounding box an ambiguous task. All aforementioned issues are greatly alleviated if we can exploit several frames from a video sequence. As the object moves relative to the background, we can search for consistencies over several views of the object appearance. While object regions should consistently stay inside the box, background patches can move from inside to outside the box over the duration of the sequence. This search for consistency is performed by our spatio-temporal aggregation S through an iterative optimization process. It operates on information extracted from the individual frames by the object encoder, which we first detail here.

Directly extracting the segmentation from a single frame is difficult and prone to errors. We can however generate an object encoding from a single frame, capturing multiple *possible* segmentation hypotheses. Each frame gives detailed information about image patches, structures and patterns that are certainly not part of the object itself. These are image regions that are strictly *outside* the bounding box, which provide important cues when combined with other frames in the sequence. To extract such frame-wise object information, we integrate an object encoder B. It takes information available in frame t by inputting the deep image representation $x_t = F(I_t)$ along with the object bounding box b_t. We first convert the box b_t to a corresponding rectangular mask representation in the input image coordinates. This is then processed by several convolutional and pooling layers, which increase the dimensionality while reducing the spatial resolution to be the same of the deep features x_t. The resulting activations are then concatenated with the features x_t and further processed by several residual blocks.

Through the deep features x_t, the object encoder B can extract candidate object shapes, which are used when searching for consistency over several frames. Specifically, the object encoder has three outputs,

$$(e_t, w_t, m_t) = B(x_t, b_t)\,, \quad e_t, w_t, m_t \in \mathbb{R}^{H\times W\times C}\,. \tag{1}$$

All outputs have the same spatial resolution $H \times W$ as the features x_t. The abstract embedding e_t holds information about the candidate object shapes, and background regions in image I_t. Intuitively, at spatial location (i, j), the activation vector $e_t[i, j] \in \mathbb{R}^C$ encodes probable segmentations of the corresponding image region. Note that the certainty of this encoding $e_t[i, j]$ can vary spatially and over the feature channels. For instance, regions outside the bounding box are certainly not part of the object, while regions inside the box can be ambiguous. To model this uncertainty in $e_t[i, j]$, we also predict a corresponding confidence weight w_t for each element in e_t. The output m_t also contains single-frame object encoding, similar to e_t. However, m_t is directly input to the segmentation decoder D. In contrast, the encoding e_t and its confidence w_t are given to the spatio-temporal aggregation module, detailed next.

3.2. Spatio-Temporal Aggregation

It is the task of the spatio-temporal aggregation module to mine the object information over multiple frames. However, designing a neural network module capable of effectively integrating information from multiple frames is a challenging and intricate problem as the object changes location and pose in each frame. As a result, temporal pooling, concatenation, or convolutions cannot find the desired consistencies. Moreover, these operations do not consider detailed global information. When deciding whether a patch corresponds to foreground or background, we need to find and reason about all similar patches in the given frames.

The main idea of our formulation is to find an underlying object representation z that best *explains* the observed object embedding e_t. That is, the representation z should indicate *consistent* local correlations between the deep image features x_t and the corresponding object embedding e_t. We formulate this as the problem of finding the best fitting local linear mapping from features vectors $x_t[i, j] \in \mathbb{R}^D$ to corresponding embedding vectors $e_t[i, j] \in \mathbb{R}^C$. This is most conveniently expressed as a convolution with the filter $z \in \mathbb{R}^{K\times K\times D\times C}$, where K is the kernel size. Using the squared error to measure the fit, our temporal aggregation module is formulated as

$$\{s_t\}_1^T = S\big(\{(x_t, e_t, w_t)\}_1^T\big) = \{x_t * z^*\}_1^T \quad \text{where} \tag{2a}$$

$$z^* = \arg\min_z \frac{1}{T}\sum_{t=1}^{T} \big\|w_t \cdot (x_t * z - e_t)\big\|^2 + \lambda\|z\|^2. \tag{2b}$$

The filter z is thus optimized to predict the embedding e_t from the features x_t. In order to minimize the objective, the filter must focus on consistent local correlations between x_t and e_t, while ignoring accidental relations that do not reoccur. The predicted confidence w_t actively weights the error at each spatio-temporal location and channel dimension through an element-wise multiplication. Our network can

therefore learn to ignore information in e_t that is deemed uncertain by predicting a low weight w_t, while emphasizing other information by giving a large importance weight. The regularization weight λ is learned during training.

During both inference and training, the optimization problem (2b) needs to be solved for every forward pass of the network. The solver thus needs to be efficient in order to ensure practical training and inference times. Moreover, the solution z^* needs to be differentiable w.r.t. to the inputs $\{(x_t, e_t, w_t)\}_1^T$ and λ. While it is possible to directly compute the closed-form solution of (2b), it involves large-scale matrix operations which are computationally heavy. Thus, we employ unrolled steepest-descent based optimization strategy utilized in [5, 7] to yield a simple and fast solution. As the algorithm employs iterative updates to z through a differentiable closed-form expression, backpropagation is automatically achieved through the standard auto-differentiation implemented in deep learning libraries.

After mining for spatio-temporal consistencies through the iterative minimization of (2b), the filter z^* contains a strong representation of the object. It encapsulates the consistent patterns and correlations of the object, integrating both spatial and temporal information. The output segmentation encoding s_t of the spatio-temporal aggregation module is achieved by applying the optimized representation z^* to the deep features x_t of each frame in (2a) as $s_t = x_t * z^*$. This is then input to our decoder $y_t = D(s_t, m_t, x_t)$, which generates a final object segmentation y_t.

Relation to [7]: Our approach can be seen as an extension of the internal learner employed in the LWL VOS method [7]. LWL tackles a few-shot learning problem where the goal is to learn a parametric target model for VOS, using the mask annotation provided in the first frame. The model is then applied on subsequent test frames to segment the target. In contrast, our formulation (2) does not assume access to any segmentation annotation. Instead, we exploit the spatio-temporal consistencies within input frames to output segmentation encoding s_t for each frame, using only the box annotation. Thus, formulation (2) serves the purpose of spatio-temporal fusion in our approach, as opposed to a few-shot learning objective in LWL.

3.3. Iterative Refinement

In this section, we describe a method to further refine the object segmentations using existing components in our architecture. The decoder module learns powerful segmentation priors by integrating deep features from different levels. It is able to extract accurate object boundaries and filter out potential errors. The segmentation embedding s_t predicted by the spatio-temporal aggregation module (2) is thus enriched with these priors in order to generate the output segmentation y_t. Note that this represents new knowledge not seen by the aggregation module in the first pass. We can therefore utilize this information by feeding the output segmentation masks back into the aggregation step.

To this end, we create a secondary object encoder $\hat{B}$, taking the predicted mask y_t. Since the preliminary mask y_t already encapsulates a detailed representation of the object extent, we found it to be sufficient for generating the object embedding $\hat{e}_t$ and confidence weights $\hat{w}_t$ used by the aggregation module. Thus for each frame we predict,

$$(\hat{e}_t, \hat{w}_t) = \hat{B}(y_t)\,, \qquad e_t, w_t \in \mathbb{R}^{H\times W\times C}\,. \tag{3}$$

Note that we do not re-generate the single-frame information m_t later used by the decoder. Instead, we employ the one stemming from the original object encoder (1).

The object encoding $\hat{e}_t$ and corresponding weights $\hat{w}_t$ now include new and more accurate information about the object. We integrate this for mask prediction by inputting it to our spatio-temporal aggregation module (2) to generate new segmentation encodings $\{\hat{s}_t\}_1^T = S\big(\{(x_t, \hat{e}_t, \hat{w}_t)\}_1^T\big)$. Note that this implies solving a new optimization problem (2b), which mines spatio-temporal consistencies. The final segmentation mask $\hat{y}_t$ is obtained using the same decoder module as $\hat{y}_t = D(\hat{s}_t, m_t, x_t)$. While the process could be repeated several times, we did not observe noticeable improvement from a third iteration. This is however expected as the strong segmentation priors of the decoder is already exploited by the aggregation module in the second iteration.

3.4. Training

Our complete model is fully differentiable and hence can be trained end-to-end using existing mask annotated video datasets. From a ground truth mask y_t^{GT}, we extract a corresponding bounding box b_t by taking smallest axis-aligned box containing the mask y_t^{GT}. Our network is trained on sub-sequences of length T by minimizing the loss,

$$L = \frac{1}{T}\sum_{t=1}^{T} \ell(y_t, y_t^{\text{GT}}) + \frac{1}{T}\sum_{t=1}^{T} \ell(\hat{y}_t, y_t^{\text{GT}})\,. \tag{4}$$

Here, y_t and $\hat{y}_t$ are the segmentation outputs generated by the initial prediction and the refinement respectively. Further, ℓ denotes a generic segmentation loss.

For our experiments, we use the YouTube-VOS [63] and DAVIS 2017 [49] datasets. We sample sequences from both datasets using a 6 times higher probability for YouTube-VOS compared to DAVIS 2017 training set. We then randomly sample sub-sequences of length $T = 3$ frames within a temporal window of length 100. For each frame, we first crop a patch that is 5 times larger than the ground-truth bounding box, while ensuring the maximal size to be equal to the image itself. We then resize the cropped patch to 832×480 with the same aspect ratio. Only random horizontal flipping is employed for data augmentation.

We initialize our backbone ResNet-50 with Mask R-CNN weights from [42]. All the remaining modules are initialized using [20]. We use the Lovasz [4] loss as our segmentation loss ℓ in (4). The network parameters are learned using the ADAM [28] optimizer with a batch size of 4. We train our network for 80k iterations with the backbone weights fixed. The learning rate is initialized as 10^{-2} and then reduced by a factor of 5 after 30k and 60k iterations. The entire training takes 32 hours on a single GPU.

3.5. Implementation Details

Architecture: Here, we give further details about our architecture. We use a ResNet-50 backbone network as feature extractor. For the object encoder B and spatio-temporal aggregation module S, we employ the third residual block and add another convolutional layer which reduces the dimensionality to 512. The object encoder generates outputs (1) with a dimension $C = 16$. We adopt the segmentation decoder used in [7, 50]. We first concatenate the segmentation embedding s_t from (2) with the single-frame information m_t from (1). The decoder then progressively increases the resolution while integrating deep features from different levels in F. For the spatio-temporal aggregation module (2), we first initialize the object representation z to zero. We then apply 5 steepest descent iterations [7] to optimize (2b) during training. The kernel size of z is set to $K = 3$.

Inference: For a given input video, we extract a sequence of T frames. Since our method benefits from using *different* views of the target and background, we do not extract directly subsequent frames as they are highly correlated. Instead, we take T frames with an inter-frame interval of Δ. In order to segment all frames, we simply proceed by shifting the sub-sequence one step each time. We generally employ $T = 9$ and $\Delta = 15$. We analyze the impact of the sequence length T in Sec. 4.1. For the spatio-temporal aggregation (4), we found it beneficial to increase the number of steepest-descent iterations to 15 during inference.

4. Experiments

We perform comprehensive experiments to validate our contributions. A detailed ablative analysis of our architecture is provided in Sec. 4.1. We demonstrate the effectiveness of our approach for partially supervised training of VOS in Sec 4.2. Finally, in Sec. 4.3, we use our network to annotate large-scale tracking datasets and show improved tracking performance using the generated annotations.

4.1. Ablation Study

We perform a detailed ablative study, analysing the impact of key components in our approach. The analysis is performed on DAVIS 2017 validation set, as well as YT300 set which has been previously utilized in [7, 25]. YT300 consists of 300 sequences which are sampled randomly from the YouTube-VOS 2019 training set and not used for training our models. The methods are evaluated using the mean Jaccard $\mathcal{J}$ index. Unless specified, inference is performed using the settings described in Sec. 3.5. We only employ a different $\Delta = 5$ for DAVIS due to faster motions.

Num. Frames	1	3	5	7	9	11
YT300	84.2	85.2	85.5	85.6	85.6	85.6
DAVIS2017 val	78.7	80.4	80.9	81.1	81.2	81.2

Table 1. Impact of using multiple frames for box to mask conversion. Results are shown in terms of Jaccard $\mathcal{J}$ index.

Impact of using multiple frames: We investigate the impact of exploiting information from multiple frames to convert boxes to masks by evaluating our approach using different number of input frames. The result of this comparison is shown in Table 1, and qualitative examples are provided in Fig. 3. When using a single frame as input, our approach obtains a $\mathcal{J}$ score of 84.2 and 78.7 on YT300 and DAVIS 2017 validation set, respectively. The performance of our approach improves substantially when using multiple input frames. The best results are obtained when using 9 frames, with a $\mathcal{J}$ score of 81.2 on the DAVIS 2017. These results clearly demonstrate the advantages of using multiple frames to perform accurate box to mask conversion.

Analysis of architecture: Here, we analyse the impact of different components in our architecture. We evaluate four variants of our method; i) **SingleImage:** A single image baseline which only uses the single-frame object representation m_t to independently convert boxes to masks in each frame. ii) **MultiFrame:** Our spatio-temporal aggregation module is used to obtain the segmentation encoding s_t by exploiting multiple frames. iii) **MultiFrame+:** The single-frame object representation m_t is passed to the segmentation decoder, in addition to the segmentation encoding s_t. iv) **MultiFrameIterative:** We employ the iterative refinement strategy described in Sec 3.3 to refine the initial segmentation prediction obtained using **MultiFrame+**. Results are shown in Tab. 2. The SingleImage achieves a $\mathcal{J}$ score of 83.3 and 77.2 on YT300 and DAVIS 2017 validation set, respectively. The MultiFrame model, which exploits object information from multiple frames achieves significantly better results, with an improvement of $+2.6$ in $\mathcal{J}$ score on DAVIS 2017. This demonstrates the effectiveness of our spatio-temporal aggregation module in effectively combining the information from multiple frames. Using the single-frame object representation m_t in combination with the segmentation encoding s_t provides a slight improvement. Finally, performing iterative refinement of the initial segmentation prediction provides a further improvement of $+1.1$ in $\mathcal{J}$ score on DAVIS 2017. This shows that the segmentation decoder contains rich prior information which can comple-

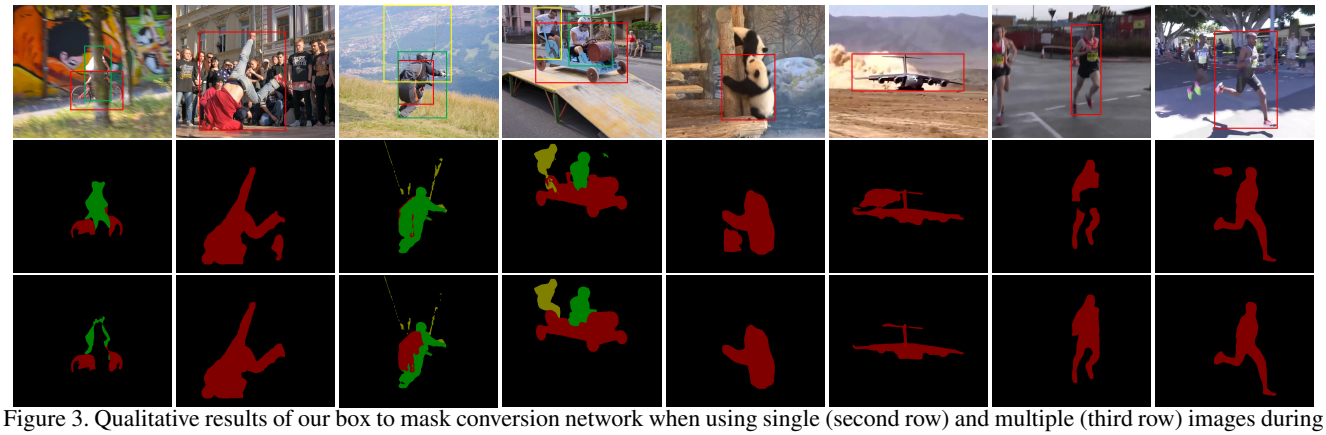

Figure 3. Qualitative results of our box to mask conversion network when using single (second row) and multiple (third row) images during inference. Our approach can effectively exploit multiple frames to handle challenging cases by mining spatio-temporal consistencies.

	m_t	s_t	Iter.	YT300	DAVIS2017 val
SingleImage	✓			83.3	77.2
MultiFrame		✓		84.6	79.8
MultiFrame+	✓	✓		84.8	80.1
MultiFrameIterative	✓	✓	✓	85.6	81.2

Table 2. Impact of different components in the proposed approach. Results are reported in terms of Jaccard $\mathcal{J}$ score.

ment our spatio-temporal aggregation module.

Spatio-temporal aggregation: Here, we compare our approach with alternate strategies for aggregating information from multiple frames. We replace our aggregation module with two different approaches; i) **STA:** We perform dense space-time matching utilized in [44] to aggregate information over frames, using the features x_t as keys, and encodings e_t as values; ii) **Concat:** We concatenate e_t over all frames along the channel dimension and pass it through a small network to obtain the fused segmentation encoding s_t. Additionally, we include our SingleImage variant, as well as an off-the-shelf single frame box to mask conversion network Box2Seg (proposal refinement network from [38]) for comparison. Results are shown in Tab. 3. Both **STA** and **Concat** approaches fail to effectively fuse information from multiple frames, providing only minor improvement over the SingleImage baseline. In contrast, our approach outperforms the single frame Box2Seg network by $+1.9$ in $\mathcal{J}$ score on DAVIS2017 val, demonstrating that it can effectively integrate information from multiple frames.

4.2. Partially Supervised VOS Training

In this section, we validate the effectiveness of our approach to generate pseudo-labels for partially supervised training of VOS models. We consider the scenario where pixel-wise segmentation labels are only available for a small number of training sequences, while the rest of the sequences have bounding box annotation for the objects. This is a highly practical scenario as generating bounding box labels is significantly faster compared to obtaining pixel-wise

	Box2Seg [38]	SingleImage	Concat.	STA	**Ours**
YT300	-	83.3	83.4	83.2	85.6
DAVIS2017 val	79.3	77.2	77.8	78.5	81.2

Table 3. Comparison with alternate approaches of integrating information from multiple frames, in terms of Jaccard $\mathcal{J}$ score. Results for Box2Seg are from [60].

	Only A	MIL	MIL+CRF	**Ours**	FS
$\mathcal{J}$	76.9	77.7	77.8	78.9	79.8

Table 4. Comparison with other partially supervised training methods on the YT300 dataset, in terms of Jaccard $\mathcal{J}$ Index.

mask annotations. In such cases, it is desirable to exploit the bounding box annotations to perform partially supervised training in order to benefit from more training data. In order to evaluate our approach for this setting, we simulate the training scenario using YouTube-VOS 2019 training set. We randomly split YouTube-VOS training set into two subsets A and B in the ratio 1:9. The segmentation labels are available for set A, while only the bounding box annotations are made available for videos in set B.

We use the mask annotated videos from set A to train our video box to mask conversion network. The trained model is then used to generate pseudo-labels for video in set B, using only the bounding box annotation. A VOS model is then trained using the combined datasets A and B. We compare our approach of generating pseudo labels using video with two alternative; i) **MIL** We use the recently introduced multiple instance learning (MIL) loss [21] to compute training loss on the box annotated videos from set B. ii) **MIL+CRF** We use the MIL loss in combination with the CRF regularizer introduced in [56] to compute training loss. Additionally, we also report the results obtained when using only the fully annotated set A for training (**Only A**), as well as the upper bound attained when using the complete YouTube VOS training set with mask annotations (**FS**). We use the recently introduced LWL [7] approach as our VOS model for this experiment. The LWL network is trained using each of the partially supervised methods, for 100k iterations. The

	STM[44]	AlphaRef[64]	OceanPlus[65]	RPT[39]	LWL	**LWL-Ours**
EAO	0.308	0.482	0.491	0.530	0.463	0.510
Accuracy	0.751	0.754	0.685	0.700	0.719	0.732
Robustness	0.574	0.777	0.842	0.869	0.798	0.824

Table 5. State-of-the-art comparison on VOT2020 in terms of expected average overlap (EAO), accuracy, and robustness.

	SiamRPN++ [33]	DiMP-50 [5]	PrDiMP-50 [14]	LWL	**LWL-Ours**
$SR_{0.5}$ (%)	82.8	88.7	89.6	92.4	95.1
$SR_{0.75}$ (%)	-	68.8	72.8	82.2	85.2
AO (%)	73.0	75.3	77.8	84.6	86.7

Table 6. State-of-the-art comparison on the GOT-10k validation set in terms of average overlap (AO) and success rates (SR) at overlap thresholds 0.5 and 0.75

result of this comparison is shown in Table 4, on the YT300 set. Both the **MIL** and **MIL+CRF** approaches obtain an improvement of around $+0.9$ in $\mathcal{J}$ score, compared to the naïve baseline using only the mask annotated videos from set A for training. Our approach of generating pseudo labels obtains the best results, achieving a substantial improvement of over $+1$ in $\mathcal{J}$ score over the MIL baselines. These results demonstrate the quality and effectiveness of the masks generated from our approach for performing partially supervised training for VOS.

4.3. VOS in the Tracking Domain

We utilize the capability of performing partially supervised VOS training using box annotations to train a VOS method on large-scale tracking datasets in order to obtain improved tracking performance. We use our network to annotate tracking datasets LaSOT [16] and GOT10k [23] containing 1120 and 9340 training sequences, respectively. These datasets contain a variety of object classes and motion types which are not often included in standard VOS datasets [49, 63]. The pseudo annotated tracking sequences, along with the fully annotated YouTube-VOS and DAVIS datasets are then used to fine-tune a VOS model. We start with the LWL [7] model trained with fixed backbone weights. The complete model, including the backbone feature extractor, is then trained on combined YouTube-VOS, DAVIS, LaSOT, and GOT-10k datasets for 120k iterations. We compare this model, denoted **LWL-Ours**, with the state-of-the-art on VOT2020 [30], GOT10K [23], and TrackingNet [43] datasets. For comparison, we also report results for the standard LWL model fine-tuned using only the YouTube-VOS and DAVIS datasets.

VOT2020 [30]: We evaluate our LWL-Ours model on VOT2020 dataset consisting of 60 challenging sequences. Similar to semi-supervised VOS, the trackers are provided an initial object mask. In order to obtain robust performance measures, the trackers are evaluated multiple times on each sequence, using different starting frames. The trackers are compared using the accuracy, robustness, and expected average overlap EAO measure. Accuracy denotes the average overlap between tracker prediction and the ground truth over the successfully tracked frames, while robustness measures the fraction of sequence tracked on average before tracking loss. Both these measures are combined to obtain the EAO score. LWL-Ours fine-tuned on tracking datasets, obtains a relative improvement of over 10% in EAO score, compared to the LWL baseline (see Tab. 5). Furthermore, despite performing vanilla VOS, LWL-Ours outperforms existing tracking approaches, achieving the second best EAO score. These results show that the masks generated from our approach can be utilized to improve the generalization of VOS model on generic tracking datasets.

	SiamRPN++ [33]	DiMP-50 [5]	KYS [6]	SiamRCNN [60]	LWL	**LWL-Ours**
Precision (%)	69.4	68.7	68.8	80.0	78.4	79.1
Norm. Prec. (%)	80.0	80.1	80.0	85.4	84.4	84.7
Success (AUC) (%)	73.3	74.0	74.0	81.2	80.7	81.2

Table 7. State-of-the-art comparison on the TrackingNet test set in terms of precision, normalized precision, and success.

GOT10k [23]: We evaluate LWL-Ours on the validation split of GOT10k dataset, consisting of 180 videos. Unlike VOT2020, trackers are only provided an initial box and required to output a target box for each frame. Thus, we use our box to mask conversion network to obtain the initial segmentation mask. The VOS model is then run using the generated mask. In each subsequent frame, we simply compute the target box using the extreme points of the predicted segmentation mask, without performing any post-processing. Fine-tuning LWL on our pseudo-annotated tracking videos provides an improvement of 2.1% in AO over the baseline LWL model (see Tab. 6). Moreover, LWL-ours significantly outperforms existing trackers with an AO score of 86.7%.

TrackingNet [43]: We report results on the test split of TrackingNet dataset consisting of 511 videos, using the same evaluation strategy employed for GOT10k dataset. Using our generated masks on tracking datasets for fine-tuning improves the results of the LWL model by 0.5% in terms of success score (see Tab. 7). Moreover, LWL-Ours obtains the best results among all methods in terms of Success score, along with SiamRCNN [60].

5. Conclusion

We propose an end-to-end trainable method for predicting object masks from bounding boxes in videos. Our approach can effectively mine object and background information over multiple frames using a novel spatio-temporal aggregation module. The predicted masks are further refined using an iterative formulation. Our approach obtains superior segmentation accuracy, compared to single image baselines. We further demonstrate the usefulness of our method for partially supervised VOS training for tracking.

Acknowledgments: This work was supported by a Huawei Technologies Oy (Finland) project, the ETH Zürich Fund (OK), an Amazon AWS grant, and Nvidia.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018. 2

[2] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision*, pages 549–565. Springer, 2016. 2

[3] Harkirat Singh Behl, Mohammad Najafi, Anurag Arnab, and Philip HS Torr. Meta learning deep visual words for fast video object segmentation. *arXiv preprint arXiv:1812.01397*, 2018. 2

[4] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. 6

[5] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6182–6191, 2019. 5, 8

[6] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 8

[7] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 5, 6, 7, 8

[8] 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

[9] Liang-Chieh Chen, Y. Zhu, G. Papandreou, Florian Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 3

[10] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1189–1198, 2018. 2

[11] Isaac Cohen and Gerard Medioni. Detecting and tracking moving objects for video surveillance. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 319–325. IEEE, 1999. 1

[12] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015. 2

[13] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016. 3

[14] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7192, 2020. 8

[15] Adám Erdélyi, Tibor Barát, Patrick Valet, Thomas Winkler, and Bernhard Rinner. Adaptive cartooning for privacy protection in camera networks. In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 44–49. IEEE, 2014. 1

[16] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5374–5383, 2019. 8

[17] Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks: A new hope. *CoRR*, abs/1909.04866, 2019. 2

[18] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 3

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 3

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. 6

[21] Cheng-Chun Hsu, Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised instance segmentation using the bounding box tightness prior. In *Advances in Neural Information Processing Systems*, pages 6586–6597, 2019. 2, 7

[22] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 54–70, 2018. 2

[23] 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*, page 1–1, 2019. 8

[24] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018. 2

[25] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2019. 6

[26] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 876–885, 2017. 2

[27] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for object tracking. In *The DAVIS Challenge on Video Object Segmentation*, 2017. 2

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[29] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, pages 695–711. Springer, 2016. 2

[30] Matej Kristan, Alan Lukezic, Martin Danelljan, Luka Cehovin Zajc, and Jiri Matas. The new vot2020 short-term tracking performance evaluation protocol and measures. 8

[31] Viveka Kulharia, Siddhartha Chandra, Amit Agrawal, Philip Torr, and Ambrish Tyagi. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[32] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9207–9216, 2019. 3

[33] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019. 8

[34] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *Asian Conference on Computer Vision*, pages 638–653. Springer, 2018. 3

[35] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 90–105, 2018. 2

[36] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367, 2017. 3

[37] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016. 2

[38] 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. 2, 3, 7

[39] Ziang Ma, Linyuan Wang, Haitao Zhang, Wei Lu, and Jun Yin. RPT: Learning Point Set Representation for Siamese Visual Tracking. *arXiv e-prints*, page arXiv:2008.03467, Aug. 2020. 8

[40] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6):1515–1530, 2018. 2

[41] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–625, 2018. 2

[42] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. Accessed: 28/10/2020. 6

[43] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 300–317, 2018. 8

[44] 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. 1, 2, 7, 8

[45] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1742–1750, 2015. 2

[46] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017. 2

[47] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015. 2

[48] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):128–140, 2016. 2

[49] 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. 1, 2, 5, 8

[50] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7406–7415, 2020. 2, 6

[51] German Ros, Sebastian Ramos, Manuel Granados, Amir Bakhtiary, David Vazquez, and Antonio M Lopez. Vision-based offline-online perception paradigm for autonomous

driving. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 231–238. IEEE, 2015. 1

[52] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004. 2

[53] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 993–1000. IEEE, 2006. 3

[54] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1939–1946, 2013. 3

[55] Khaled Saleh, Mohammed Hossny, and Saeid Nahavandi. Kangaroo vehicle collision detection using deep semantic segmentation convolutional neural network. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7. IEEE, 2016. 1

[56] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 507–522, 2018. 7

[57] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7158–7166, 2017. 2

[58] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9481–9490, 2019. 1, 2

[59] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 2

[60] 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. 2, 7, 8

[61] P. Voigtlaender, L. Luo, C. Yuan, Yong Jiang, and B. Leibe. Reducing the annotation effort for video object segmentation datasets. In *WACV*, 2021. 2

[62] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018. 2

[63] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1, 2, 5, 8

[64] Bin Yan, Xinyu Zhang, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Alpha-refine: Boosting tracking performance by precise bounding box estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5289–5298, June 2021. 8

[65] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *European Conference on Computer Vision (ECCV)*, August 2020. 8

[66] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3791–3800, 2018. 2