Classifying, Segmenting, and Tracking Object Instances in Video with Mask Propagation

Gedas Bertasius, Lorenzo Torresani
Facebook AI

Abstract

We introduce a method for simultaneously classifying, segmenting and tracking object instances in a video sequence. Our method, named MaskProp, adapts the popular Mask R-CNN to video by adding a mask propagation branch that propagates frame-level object instance masks from each video frame to all the other frames in a video clip. This allows our system to predict clip-level instance tracks with respect to the object instances segmented in the middle frame of the clip. Clip-level instance tracks generated densely for each frame in the sequence are finally aggregated to produce video-level object instance segmentation and classification. Our experiments demonstrate that our clip-level instance segmentation makes our approach robust to motion blur and object occlusions in video. MaskProp achieves the best reported accuracy on the YouTube-VIS dataset, outperforming the ICCV 2019 video instance segmentation challenge winner despite being much simpler and using orders of magnitude less labeled data (1.3M vs 1B images and 860K vs 14M bounding boxes). The project page is at: https://gberta.github.io/maskprop/.

1. Introduction

In this paper, we tackle the recently introduced video instance segmentation problem [42]. This task requires segmenting all instances of a predefined set of object classes in each frame, classifying them, and linking individual instances over the entire sequence.

In recent years, convolutional networks have obtained remarkable results in still-image object detection [16, 33, 14, 15], and segmentation [27, 45, 8, 4]. However, extending these models to video instance segmentation is challenging. In order to localize objects precisely, these methods have to operate at very large spatial resolution. As a result, detectors based on the popular ResNet-101 or ResNet-152 backbones [17] can rarely fit more than one image per GPU during training. In the context of video instance segmentation this is problematic because tracking objects over time requires analyzing multiple video frames simultaneously.

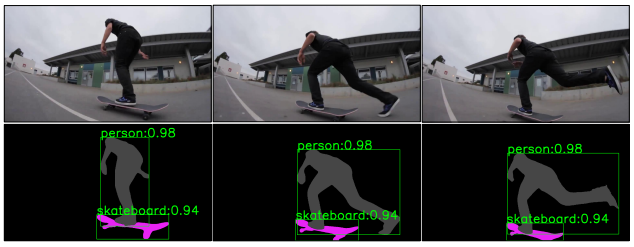

Figure 1: In this paper, we tackle the problem of video instance segmentation, which requires classifying, segmenting, and tracking object instances in a given video sequence. Our proposed Mask Propagation framework (MaskProp) provides a simple and effective way for solving this task.

To address this issue, one could reduce the spatial resolution of the input and fit more video frames in a GPU. However, doing so typically leads to a significant drop in segmentation or detection performance. Alternatively, one could perform high-resolution instance segmentation on individual frames and then link segmentations temporally in a separate post-processing stage. However, performing instance segmentation and tracking in two disjoint steps often produces suboptimal results, because these two tasks are closely intertwined. The key challenge then becomes designing a unified model that can track objects in video while maintaining strong detection accuracy.

Currently, the best method for video instance segmentation is the ICCV 2019 challenge winner [28]. It tackles video instance segmentation by dividing it into four problems: 1) detection, 2) classification, 3) segmentation, and 4) tracking. These four problems are solved independently using several off-the-shelf components and their respective solutions are combined and adapted to the video instance segmentation task. However, despite effective performance, such an approach is disadvantageous because it requires designing and tuning a separate model (or, in some cases, an ensemble of models) for each of the four tasks. This renders the approach costly and cumbersome. On the other end of the complexity spectrum, MaskTrack R-CNN [42] is a simple unified approach trained end-to-end but it achieves significantly lower performance (30.3 vs 44.8 video mAP).

		MaskTrack R-CNN [42]	ICCV19 Challenge Winner [28]	MaskProp
Model	Classification	cls head	Mask R-CNN [16], ResNeXt-101 32x48d [30]	cls head
	Localization	bbox head	Mask R-CNN [16]	bbox head
	Segmentation	mask head	DeepLabv3 [9], Box2Seg [29]	mask head
	Tracking	tracking head	UnOVOST [47], ReID Net [18, 31]	mask propagation head
	Optical Flow	-	PWC-Net [35]	-
Pre-training Datasets	ImageNet [34] (1.3M images)	✓	✓	✓
	COCO [25] (860K bboxes)	✓	✓	✓
	Instagram [30] (1B images)	-	✓	-
	OpenImages [23] (14M bboxes)	-	✓	-
Performance	video mAP	30.3	44.8	**46.6**
	video AP@75	32.6	48.9	**51.2**

Table 1: A table comparing our work to prior video instance segmentation methods [42, 28]. The ICCV 2019 Challenge Winner [28] decomposes video instance segmentation into four different problems, solves each of them independently using ensembles of different models, and then combines these solutions. In contrast, our approach relies on a single unified model trained end-to-end. Despite being simpler, and using several orders of magnitude less pretraining data (1.3M vs 1B images and 860K vs 14M bounding boxes) our model achieves higher accuracy. Furthermore, compared to MaskTrack R-CNN [42], our work yields a 16.3% gain in mAP (46.6% vs 30.3%).

To address the shortcomings of these prior methods we introduce MaskProp, a simple mask propagation framework for simultaneously classifying, segmenting and tracking object instances in video. Our method adapts the popular Mask R-CNN [16] to video by adding a branch that propagates frame-level instance masks from each video frame to other frames within a temporal neighborhood (which we refer to as a clip). This allows our method to compute clip-level instance tracks centered at each individual frame of the video. These densely estimated clip-level tracks are then aggregated to form accurate and coherent object instance sequences for the entire video, regardless of its length. This renders our approach capable of handling challenging cases of occlusions, disocclusions, and motion blur. Our method achieves the best reported accuracy on the YouTube-VIS dataset [42], outperforming the ICCV 2019 challenge winner [28] despite being much simpler and using significantly less labeled data (1000x fewer images and 10x fewer bounding boxes). In Table 1, we compare our approach vs these prior methods in terms of accuracy and other characteristics.

2. Related Work

Instance Segmentation in Images. Compared to instance segmentation in images [11, 16, 44, 1, 2, 22, 26], the problem considered in this paper requires not only to segment object instances in individual frames, but also to determine instance correspondences across multiple frames. We leverage the Mask R-CNN model [16] for still-image instance segmentation and adapt it to track object instances in video.

Object Detection in Video. Object detection in video requires classifying and localizing objects in every frame of a given video. Most modern video object detection systems [46, 5, 40, 13] implement some form of spatiotemporal feature alignment for improving object detection accuracy in individual video frames. However, these systems are typically not designed for tracking object instances. In contrast, our mask propagation produces clip-level instance segmentations rather than frame-level bounding boxes.

Video Object Segmentation. The task of video object segmentation requires segmenting foreground objects in a class-agnostic fashion [43, 21, 36, 38], often by leveraging ground truth masks available for the first frame during inference [6, 32, 10, 19, 37]. Instead, video instance segmentation requires finding all instances of a predefined set of object classes in each frame, classifying them and linking them over the entire sequence.

Video Instance Segmentation. The recently introduced video instance segmentation task [42] requires classifying, segmenting and tracking object instances in videos. This is the task considered in this work. There are only a few video instance segmentation methods we can compare our approach to. The MaskTrack R-CNN [42] presents a unified model for video instance segmentation. It augments the original Mask R-CNN [16] with a tracking branch that establishes associations among object instances segmented in separate frames. Furthermore, we include the ICCV 2019 video instance segmentation challenge winner [28] in our comparison. This approach divides video instance segmentation into four separate subproblems: classification, detection, segmentation, and tracking. A separate model (or an ensemble of models) is used to solve each of these subproblems, and these solutions are then combined to produce video instance segmentation results. For brevity, from now on we refer to it as EnsembleVIS to indicate that it is an ensemble approach designed for video instance segmentation.

Our MaskProp framework provides advantages over both of these methods [42, 28]. Similarly to MaskTrack R-CNN [42], our method is a unified and simple approach. However, our mask propagation branch is much more effective than the tracking branch of MaskTrack R-CNN, achieving much higher accuracy relative to this baseline. Furthermore, compared to EnsembleVIS [28], our method 1)

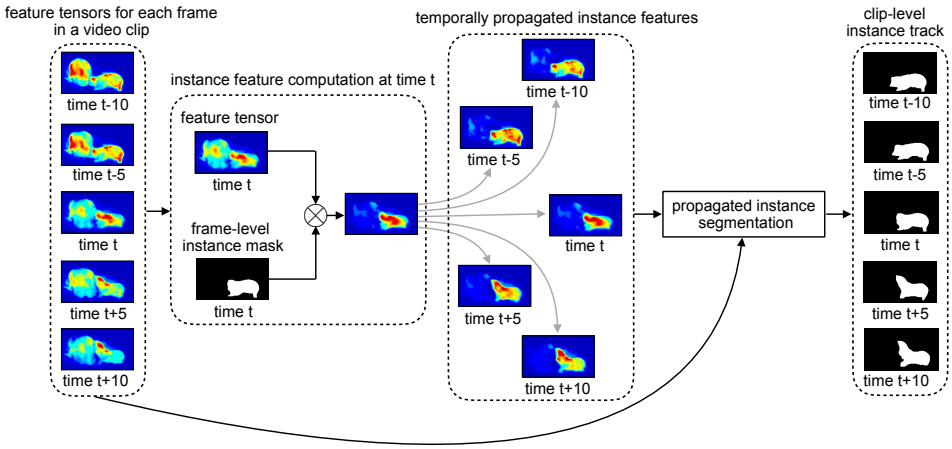

Figure 2: An illustration of our MaskProp system, which takes as input a video clip centered around frame t, and outputs a clip-level instance track. Our mask propagation framework can be summarized in three high-level steps: 1) An instance-specific feature map at time t is computed by masking the frame features at time t with the given instance segmentation for frame t (one for each instance detected in frame t). 2) Next, we use our mask propagation mechanism to temporally propagate instance-specific features from frame t to all the other frames in the clip. 3) Lastly, our model predicts instance-specific segmentations in every frame of the clip by implicitly matching the propagated instance features with the frame-level features computed at each time step. This last step yields clip-level instance tracks centered around frame t.

is much simpler, 2) uses significantly less labeled data, and 3) produces higher accuracy on YouTube-VIS [42].

3. Video Instance Segmentation

Problem Definition. Let us denote with $V \in \mathcal{R}^{L\times 3\times H\times W}$ an input video consisting of L RGB frames of spatial size $H\times W$. The aim of our system is to segment and temporally link all object instances that are visible for at least one frame in V and that belong to a predefined set of categories $\mathcal{C} = \{1, ..., K\}$. To achieve this goal, our model outputs a video-level instance mask track $M^i \in \mathcal{R}^{L\times H\times W}$ with a category label $c^i \in \{1, ..., K\}$ and a confidence score $s^i \in [0, 1]$ for each object instance i detected in the video.

Evaluation Metric. Video instance segmentation is evaluated according to the metrics of average precision (AP) and average recall (AR). Unlike in the image domain, these metrics are evaluated over the video sequence. Thus, to evaluate spatiotemporal consistency of the predicted mask sequences, the video Intersection over Union (IoU) between a predicted object instance i and a ground truth object instance j is computed as:

$$IoU(i, j) = \frac{\sum_{t=1}^{T} |M^i(t) \cap \tilde{M}^j(t)|}{\sum_{t=1}^{T} |M^i(t) \cup \tilde{M}^j(t)|} \qquad (1)$$

where $\tilde{M}^j(t)$ is the ground-truth segmentation of object j in frame t. To achieve a large IoU, a model must not only accurately classify and segment object instances at a frame-level, but also reliably track them over the video sequence.

As in the COCO benchmark for image segmentation [25], the metrics of AP and AR are computed separately for each object category, and then averaged over ten IoU thresholds from 50% to 95% at increments of 5%. Lastly, the resulting AP and AR metrics are averaged over the category set, which yields the final evaluation metric.

4. Mask Propagation

MaskProp takes a video V of arbitrary length L as input and outputs video-level instance segmentation tracks M^i, category labels c^i and confidence scores s^i for all objects i detected in the video. In order to achieve this goal, our method first builds clip-level object instance tracks $M^i_{t-T:t+T} \in \mathcal{R}^{(2T+1)\times 1\times H\times W}$ for each individual clip $V_{t-T:t+T} \in \mathcal{R}^{(2T+1)\times 3\times H\times W}$ of length $(2T + 1)$ in the video, i.e., for $t = 1, 2, \ldots, L$ (clips at the beginning and the end of the video will include fewer frames).

We want to use clips that are long enough to allow us to jointly solve instance segmentation and tracking while handling challenging cases of occlusion and motion blur. At the same time, the clip should be short enough to allow us to fit it at high spatial resolution in the memory of a GPU.

The resulting clip-level instance masks $M^i_{t-T:t+T}$ produced densely for all overlapping clips $t = 1, \ldots, L$ are then aggregated to produce video-level instance masks M^i.

Our approach for clip-level instance segmentation is described in subsections 4.1 and 4.2. We also illustrate it in Figure 2. The subsequent clip-level instance mask aggregation method is presented in subsection 4.3.

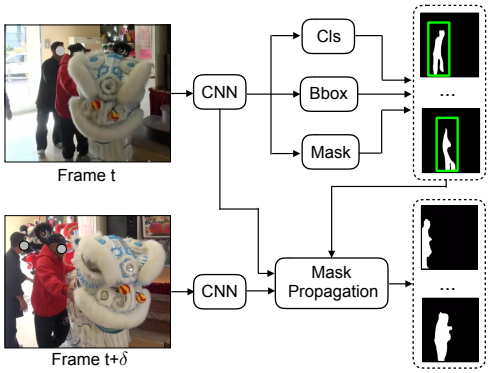

Figure 3: We adapt Mask R-CNN [16] to video by adding a mask propagation branch, for tracking object segmentation instances in video. Given a video clip centered around frame t, our system outputs a clip-level instance segmentation track as well as a classification score and a bounding box for each object instanced detected in frame t. For compactness, in this figure, we illustrate our system processing a pair of frames but typically the propagation is applied from the middle frame to all the other frames in the clip.

4.1. Video Mask R-CNN

Our video instance segmentation system is based on the Mask R-CNN [16] model, which we adapt to video by adding a mask propagation branch (See Figure 3). We train our system with a multi-task loss $L_t = L_t^{cls} + L_t^{box} + L_t^{mask} + L_t^{prop}$ where t denotes a time-step of a center frame. We use identical loss terms $L_t^{cls}, L_t^{box}, L_t^{mask}$ as in Mask R-CNN. The mask propagation loss is defined as:

$$L_t^{prop} = \sum_{i}^{\tilde{N}_t} \sum_{t'=t-T}^{t+T} 1 - sIoU(M^i_{t-T:t+T}(t'), \tilde{M}^i_{t-T:t+T}(t')) \quad (2)$$

where $M^i_{t-T:t+T}(t') \in [0,1]$ is the segmentation at time t' for an instance i predicted from a clip centered at t and $\tilde{M}^i_{t-T:t+T}(t')$ is the corresponding ground truth mask at time t'. $\tilde{N}_t$ is the number of ground truth object instances in frame t, and $sIoU$ is defined as:

$$sIoU(A,B) = \frac{\sum_p A(p)B(p)}{\sum_p A(p) + B(p) - A(p)B(p)} \quad (3)$$

where the summations in numerator and denominator are performed over every pixel location p. The loss above is a soft IoU loss, which we observed to work slightly better than the standard cross entropy loss for our task.

4.2. Mask Propagation Branch

Overview. Our main technical contribution is the design of a mask propagation branch, that allows our method to track object instances. Given a video clip $V_{t-T:t+T}$ centered at frame t, our system outputs clip-level instance masks $M^i_{t-T:t+T}$ for each predicted object instance i in frame t. Our mask propagation branch can be described in three high-level steps: 1) instance-specific feature computation, 2) temporal propagation of instance features, and 3) propagated instance segmentation. We will now describe each of these steps in more detail. We introduce our mask propagation with the example of propagating object instance masks from frame t to frame $t+\delta$ where $\delta \in [-T:T]$.

Computing Instance Specific Features. The mask branch of our model predicts frame-level instance masks $M^i_t \in \mathcal{R}^{1\times H'\times W'}$ from single frame inputs. We then use these frame-level instance masks to compute instance-specific features for frame t. Specifically, for each object instance i, we compute an element-wise product between M^i_t and the feature tensor from the backbone network f_t. This then yields a set of new feature tensors $f^i_t \in \mathcal{R}^{C\times H'\times W'}$, where $i = 1, .., N_t$, and N_t is the number of object instances detected in frame t. In other words, for each object instance i, we are zeroing out the feature values in f_t that correspond to pixels not belonging to that object instance.

Temporally Propagating Instance Features. Given frame-level features $f_t, f_{t+\delta} \in \mathcal{R}^{C\times H'\times W'}$ and instance-specific feature tensor f^i_t, our method generates a propagated instance feature tensor $g^i_{t,t+\delta}$. Intuitively, $g^i_{t,t+\delta}$ represents the features predicted by our model for object instance i in frame $t+\delta$ from an instance-specific feature tensor f^i_t. The tensor $g^i_{t,t+\delta}$ is generated by warping features f^i_t using the alignment computed from frame-level features f_t and $f_{t+\delta}$. We implement the propagation mechanism via a deformable convolution [12], which has previously been used for aligning features computed from separate frames of a video [5, 3]. Specifically, we compute the element-wise difference of tensors $f_t, f_{t+\delta}$ and feed it through a simple residual block [17], which predicts motion offsets $o_{t,t+\delta} \in \mathcal{R}^{2k^2\times H'\times W'}$. These offsets contain (x,y) sampling locations for each entry of a $k\times k$ deformable convolution kernel [12]. The propagation step takes as inputs 1) the offsets $o_{t,t+\delta}$ and 2) the instance feature tensor f^i_t, and then applies deformable convolution to output the propagated instance feature tensor $g^i_{t,t+\delta}$ for each instance i . We use subscript $t, t+\delta$ to denote the propagated instance feature because, although g is obtained by propagating the feature tensor f^i_t, the offset computation uses both frame t and frame $t+\delta$. We stress that no explicit ground truth alignment is available between frames. The deformable convolutional kernels are supervised implicitly by optimizing Eq. 2.

Segmenting Propagated Instances. Lastly, we use our propagated feature map $g^i_{t,t+\delta}$ for predicting a corresponding object instance mask in frame $t+\delta$. To do this we first, construct a new feature tensor $\phi^i_{t,t+\delta} = g^i_{t,t+\delta} + f_{t+\delta}$. The addition effectively overimposes the tensor $g^i_{t,t+\delta}$ predicted from time t for object instance i in frame $t+\delta$, with

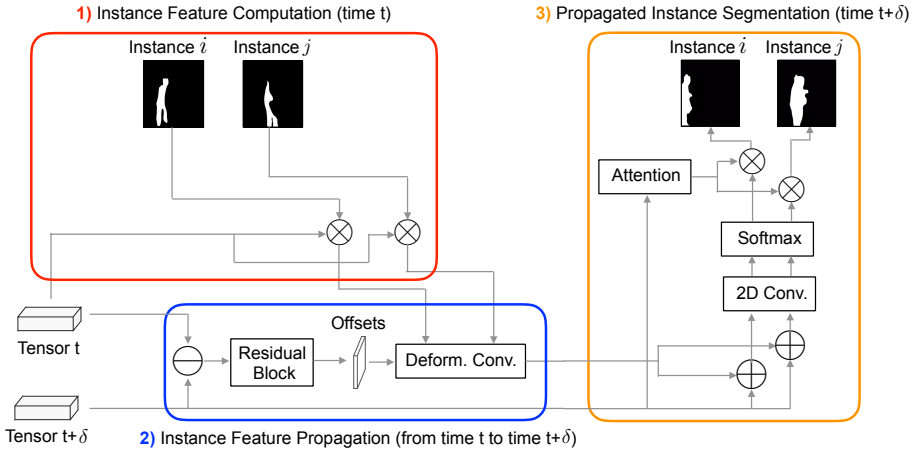

Figure 4: An illustration of the 3 steps implemented by our mask propagation branch: **1)** For every detected instance in frame t, we compute an instance-specific feature tensor via element-wise multiplication between tensor t and the given frame-level instance mask at frame t . **2)** Next, the element-wise difference of the feature tensors associated with frames t and $t+\delta$ is used to predict motion offsets between frames t and $t+\delta$. The resulting offsets are used to propagate the instance-specific tensors from time t to time $t+\delta$ via deformable convolution. The propagated tensors represent instance-specific features predicted for time $t+\delta$ using the tensors computed at time t. **3)** Lastly, we add the propagated instance feature tensors to the tensor effectively computed at $t+\delta$. A convolutional layer applied to these tensors predicts instance masks in frame $t+\delta$. The pixels that do not belong to any object instances are zeroed out using an instance-agnostic attention map.

the tensor $f_{t+\delta}$ effectively computed from the frame at time $t+\delta$. If the object instance prediction is consistent with the feature computation, the feature tensors will be aligned and thus, they will reinforce each other in the predicted region.

Finally, the resulting feature tensors $\phi^i_{t,t+\delta}$ are fed into a 1×1 convolution layer that outputs instance masks for each object instance i in frame $t+\delta$. The masks are normalized with the softmax nonlinearity across all N_t instances. To zero-out pixels that do not belong to any object instance, we use a single 3×3 convolution that computes an instance-agnostic attention map $A_{t+\delta}$ from feature tensor $f_{t+\delta}$. We then multiply $A_{t+\delta}$ with each of our predicted instance masks. A detailed illustration of our mask propagation branch is presented in Figure 4.

4.3. Video-Level Segmentation Instances

Given a video of length L, our goal is to produce video-level segmentation instances $M^i \in \mathcal{R}^{L \times H \times W}$. Conceptually, this requires linking clip-level instance tracks $M^i_{t-T:t+T}$ and $M^j_{t'-T:t'+T}$ when i and j represent the same object instance, i.e., when the instances are matching. We achieve this by assigning a video-level instance ID to each of our predicted clip-level instance tracks. Matching instance tracks are assigned the same video-level instance ID.

Matching Clip-Level Instance Tracks. Consider a pair of clip-level instance tracks $M^i_{t-T:t+T}$ and $M^j_{t'-T:t'+T}$, that are centered around frames t and t' respectively. They overlap in time if $|t-t'| < 2T+1$. Let us denote their overlapping time interval as $\cap_{t,t'}$. Given two overlapping instance tracks, we can check if they match by comparing their predicted instance masks in the overlapping frames. If the masks of the two instance tracks align well, then they are likely to encode the same object instance. Otherwise, each track represents a different object instance. We compute a matching score $m^{i,j}_{t,t'}$ between two clip-level instance tracks using our previously defined soft IoU metric as:

$$m^{i,j}_{t,t'} = \frac{1}{|\cap_{t,t'}|} \sum_{\tilde{t} \in \cap_{t,t'}} sIoU(M^i_{t-T:t+T}(\tilde{t}), M^j_{t'-T:t'+T}(\tilde{t})) \quad (4)$$

Video-Level Instance ID Assignment. We denote with $\mathcal{Y}$ the set of video-level instance IDs. The set $\mathcal{Y}$ is built incrementally by matching clip-level instance tracks $M^i_{t-T:t+T}$ in order from time $t=1$ to time $t=L$. Initially, we set $\mathcal{Y} = \{1, 2, \ldots, N_1\}$ where N_1 is the number of object instances detected at time $t=1$ (i.e., in the first clip of the video). Let $y^i_t \in \mathcal{Y}$ denote the video-level instance ID assigned to clip-level instance track $M^i_{t-T:t+T}$. As we move forward in time $t>1$, the video-level instance ID y^i_t is assigned by matching clip-level instance $M^i_{t-T:t+T}$ to all previously processed instance tracks $M^j_{t'-T:t'+T}$ that overlap with this clip, i.e., such that $\cap_{t,t'} \neq \emptyset$. For each video-level instance ID $y \in \mathcal{Y}$ already in the ID set, we compute a score $q^i_t(y)$ capturing how well $M^i_{t-T:t+T}$ matches the tracks that have been already assigned video-level ID y:

$$q^i_t(y) = \frac{\sum_{t' \text{s.t. } \cap_{t,t'} \neq \emptyset} \sum_{j=1}^{N_{t'}} 1\{y^j_{t'} = y\} \cdot m^{i,j}_{t,t'}}{\sum_{t' \text{s.t. } \cap_{t,t'} \neq \emptyset} \sum_{j=1}^{N_{t'}} 1\{y^j_{t'} = y\}} \quad (5)$$

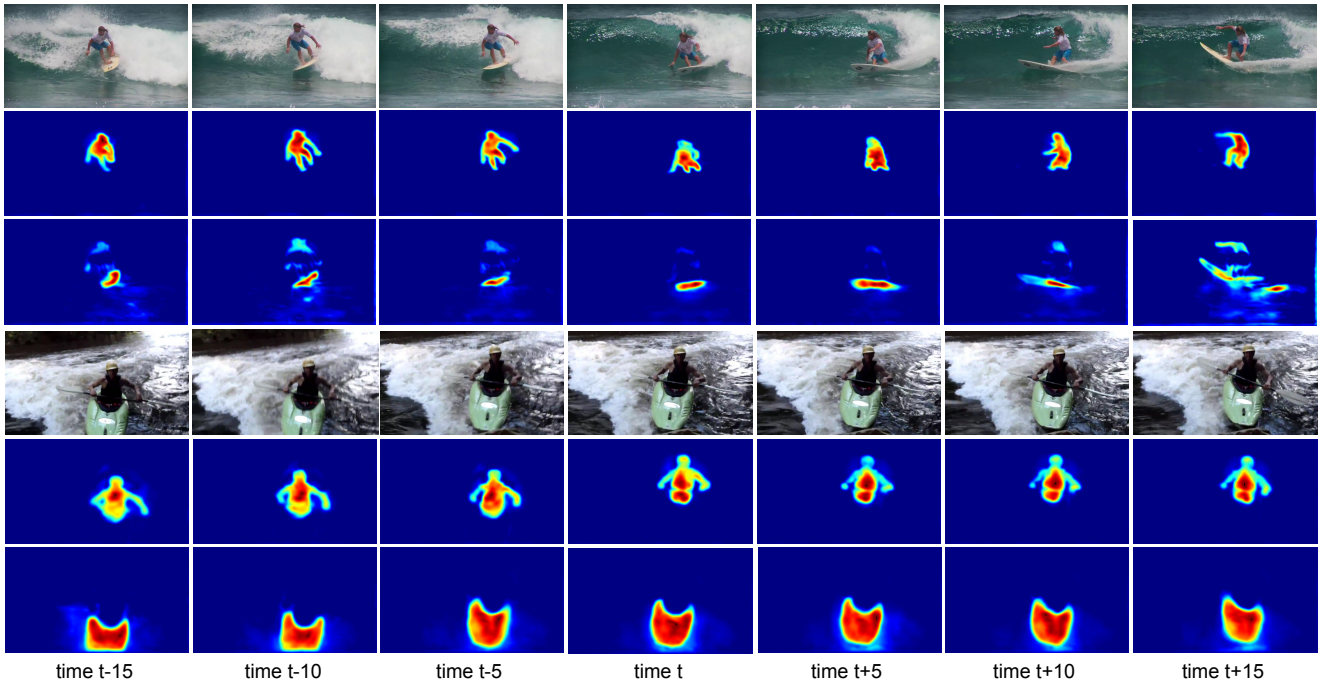

Figure 5: An illustration of instance-specific features propagated from frame t to other frames in the given video clip. Here, we visualize propagated activations from one randomly selected feature channel. The activations in the two rows correspond to two different object instances detected at time t. Our visualizations suggest that MaskProp reliably propagates features that are specific to each instance even when instances appear next to each other, and despite the changes in shape, pose and the nuisances effects of deformation and occlusion.

where $1\{y^j_{t'} = y\}$ is an indicator function that is equal to 1 when $y^j_{t'} = y$, and 0 otherwise. $N_{t'}$ is the number of detected instances at time t', and $m^{i,j}_{t,t'}$ is the previously defined matching score between instance tracks $M^i_{t-T:t+T}$ and $M^j_{t'-T:t'+T}$. The score $q^i_t(y)$ is effectively an average of the soft IoU computed between instance track $M^i_{t-T:t+T}$, and all its overlapping instance tracks that have been assigned video-level instance ID y.

Let $q^* = \max_{y \in \mathcal{Y}} q^i_t(y)$ be the maximum score obtained by considering all possible video-level instance IDs $y \in \mathcal{Y}$. If q^* is greater than a certain threshold, the current instance track $M^i_{t-T:t+T}$ is assigned the video-level instance ID $y^* = \arg\max_{y \in \mathcal{Y}} q^i_t(y)$. Otherwise, the clip-level track does not match any of the current video-level instances. In such case we create a new video-level instance ID and assign it to the the clip-level track while also expanding the set $\mathcal{Y}$, i.e., $y^i_t = |\mathcal{Y}| + 1$ and $\mathcal{Y} = \mathcal{Y} \cup \{|\mathcal{Y}| + 1\}$.

Finally, for each video-level instance ID $y \in \mathcal{Y}$, we generate the final sequence of segmentation instance masks $M^y \in \mathcal{R}^{L \times H \times W}$ as:

$$M^y(t) = \begin{cases} M^i_{t-T:t+T}(t) & \text{if } y^i_t = y \\ 0 & \text{otherwise .} \end{cases}$$

4.4. Implementation Details

Backbone Network. As our backbone we use a Spatiotemporal Sampling Network [5] based on a Deformable ResNeXt-101-64x4d [12, 41] with a feature pyramid network (FPN) [24] attached on top of it.

Detection Network. For detection, we use a Hybrid Task Cascade Network [7] with a 3 stage cascade.

Mask Propagation Branch. The residual block in the mask propagation branch consists of two 3×3 convolutions with 128 output channels each. The instance feature propagation is applied to the FPN feature map with the second largest spatial resolution. To propagate instance features from one frame to another, we predict 9 (x, y) offsets for every pixel, which are then used as input to a 3×3 deformable convolutional layer with 256 channels. To capture motion at different scales, we use three levels of dilated deformable convolutions with dilation rates $3, 6, 12$ as in [3].

High-Resolution Mask Refinement. Predicting masks from RoI features typically leads to low resolution predictions. We address this issue via a high-resolution mask refinement step. Given the center of a detected bounding box, we crop a 384×384 patch around the object, preserving the original aspect ratio. We then feed the RGB patch and the predicted low-resolution mask through 3 residual blocks each with 128 channels to obtain a high-resolution mask.

Method	Pre-training Data	mAP	AP@75	AR@1	AR@10
DeepSORT‡ [39]	Imagenet [34], COCO [25]	26.1	26.1	27.8	31.3
FEELVOS‡ [37]	Imagenet [34], COCO [25]	26.9	29.7	29.9	33.4
OSMN‡ [43]	Imagenet [34], COCO [25]	27.5	29.1	28.6	33.1
MaskTrack R-CNN‡ [42]	Imagenet [34], COCO [25]	30.3	32.6	31.0	35.5
MaskTrack R-CNN*	Imagenet [34], COCO [25]	36.9	40.2	34.3	42.9
EnsembleVIS [28]	Imagenet [34], COCO [25], Instagram [30], OpenImages [23]	44.8	48.9	42.7	51.7
MaskProp	Imagenet [34], COCO [25]	46.6	51.2	44.0	52.6
MaskProp	Imagenet [34], COCO [25], OpenImages [23]	**50.0**	**55.9**	**44.6**	**54.5**

Table 2: The results of video instance segmentation on the YouTube-VIS [42] validation dataset. We evaluate the performance of each method according to mean average precision (mAP), average precision at 75% IoU threshold (AP@75), and average recall given top 1 (AR@1) and top 10 (AR@10) detections. The baselines denoted with ‡ were implemented by the authors in [42], whereas the methods marked with * were implemented by us, and use the same backbone and detection networks as our approach. Despite its simplicity, MaskProp outperforms all prior video instance segmentation methods by a large margin.

Scoring Video-Level Sequences. Each video-level sequence contains a list of classification scores and predicted object labels. To assign a confidence score to each video-level sequence, we average classification scores associated with that sequence (separately for each object category).

Training. We use a similar training setup as in [7]. The loss weights for each of the three cascade stages are set to $1, 0.5$, and 0.25 respectively. The loss weight for the semantic segmentation branch is set to 0.1. We train our model on pairs of frames, where the second frame in a pair is randomly selected with a time gap $\delta \in [-25, 25]$ relative to first frame. We use a multi-scale training approach implemented by resizing the shorter side of the frame randomly between 400 and 800 pixels. Our model is trained in a distributed setting using 64 GPUs, each GPU holding a single clip. The training is done for 20 epochs with an initial learning rate of 0.008, which is decreased by 10 at 16 and 19 epochs. We initialize our model with a Mask R-CNN pretrained on COCO for the instance segmentation. The hyperparameters of RPN and FPN are the same as in [7].

Inference. During testing, we run the bounding box prediction branch on 1000 proposals, apply non-maximum suppression, and use boxes with a score higher than 0.1 as input to the mask prediction and mask propagation branches. During inference, our MaskProp is applied to video clips consisting of 13 frames.

5. Experimental Results

In this section, we evaluate MaskProp for video instance segmentation on YouTube-VIS [42], which contains $2,238$ training, 302 validation, and 343 test videos. Each video is annotated with per-pixel segmentation, category, and instance labels. The dataset contains 40 object categories. Since the evaluation on the test set is currently closed, we perform our evaluations on the validation set.

5.1. Quantitative Results

Video instance segmentation is a very recent task [42], and thus, there are only a few established baselines that we can compare our work to. We include in our comparison MaskTrack R-CNN [42], and the EnsembleVIS method, which won the ICCV 2019 video instance segmentation challenge [28]. Additionally, to make the comparison with MaskTrack R-CNN more fair, we reimplement it using the same backbone and detection networks as our MaskProp (see MaskTrack R-CNN* in Table 2).

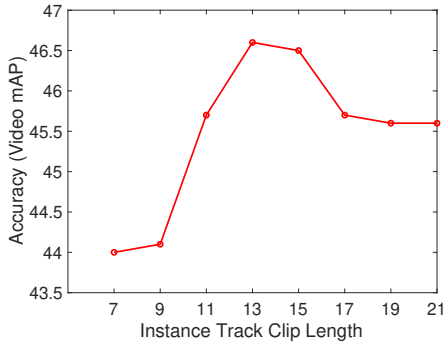

Figure 6: We plot video mAP as a function of an instance track clip length (denoted as $2T+1$ in the paper). Based on these results, we observe that optimal video instance segmentation performance is achieved when we propagate instance masks to $T = 6$ previous and subsequent frames.

We present our quantitative results in Table 2, where we assess each method according to 1) mean video average precision (mAP), 2) video average precision at IoU threshold of 75%, and 3) average recall given 1 and 10 highest scored instances per video. From these results, we observe that our MaskProp outperforms all the other baselines according to all four evaluation metrics, thus achieving state-of-the-art results in video instance segmentation on YouTube-VIS. It can be noted that we outperform EnsembleVIS [28] by 1.8 mAP even though our approach is much simpler and even if it uses orders of magnitude less labeled data for pre-training. Furthermore, compared to the MaskTrack R-CNN, our method achieves a 16.3 improvement in mAP. We also note that our implementation of MaskTrack [42] significantly improves upon the original work, but it is still 9.7

Method	mAP	AP@75
FlowNet2 Propagation	31.4	33.6
MaskTrack R-CNN*	36.9	40.2
MaskProp	**46.6**	**51.2**

Table 3: Here, we study the effectiveness of our mask propagation branch. If we replace it with the FlowNet2 propagation scheme, where masks are propagated using the optical flow predicted by a FlowNet2 network [20], the accuracy drops from 46.6 mAP to 31.4 mAP. Similarly, if we replace our mask propagation branch with the tracking branch from MaskTrack R-CNN, the accuracy drops to 36.9 mAP. Note that all of these baselines are implemented using the same backbone and detection networks.

worse in mAP compared to our model. Lastly, we note that additionally pre-training MaskProp on the OpenImages [23] dataset as was done in [28] further boosts our performance to 50.0 mAP.

5.2. Ablation Experiments

Mask Propagation Branch. To investigate the effectiveness of our mask propagation branch, we compare our method with a FlowNet2 [20] propagation baseline. For this baseline, we use exactly the same setup as for our MaskProp, except that instance masks are propagated using the optical flow predicted by a FlowNet2 network [20] rather than our proposed mask propagation scheme. For a more complete comparison, we also include the MaskTrack R-CNN* from Table 2, which uses the originally proposed tracking branch [42], but is implemented using the same backbone and detection networks as our MaskProp.

These baselines allow us to directly compare the effectiveness of our mask propagation scheme versus the propagation mechanisms employed by FlowNet2 and MaskTrack R-CNN [42] methods. The results in Table 3 show that MaskProp outperforms these baselines by a large margin.

Instance Track Clip Length. Due to occlusions, object instances in video may not be visible in some frames. If there are occlusions separated by $2T' + 2$ time-steps, we can use $T > T'$ to predict longer clip-level instance tracks. In Figure 6, we study video instance segmentation performance as a function of instance track clip length (denoted as $2T + 1$ in the paper). Our results indicate that the best accuracy is achieved when we use a clip length of 13, meaning that we propagate instances to 6 previous and 6 subsequent frames.

High-Resolution Mask Refinement. We also study the impact of our high-resolution mask refinement, described in Subsection 4.4. We report that removing this refinement causes a drop of 1.9% in video instance segmentation mAP.

Importance of Frame-Level Instance Masks. As described in our main draft, we use frame-level instance masks for instance-specific feature computation. To investigate the contribution of these masks to the performance of our system, we experiment with masks obtained from several different Mask R-CNN models. In Table 4, we present our results for this ablation. Our results indicate that frame-level instance masks obtained from stronger models allow us to achieve better video instance segmentation performance. Thus, we expect that future improvements in image instance segmentation will further benefit our method.

Backbone Network	mAP	AP@75
ResNet-50	40.0	42.9
ResNet-101	42.5	45.6
ResNeXt-101-64x4d	44.3	48.3
STSN [5]-ResNeXt-101-64x4d	**46.6**	**51.2**

Table 4: We study the effect of frame-level instance masks to our system's performance. We evaluate our method's accuracy when using instance masks obtained from Mask R-CNN with several different backbones. Our results indicate that frame-level instance masks obtained from stronger models lead to better video instance segmentation results.

5.3. Qualitative Results

In Figure 7, we compare our predicted clip-level instance tracks (last row of predictions for each clip) with the MaskTrack R-CNN predictions (first row of predictions). We use different colors to represent different object instances. Our qualitative results suggest that our MaskProp produces more robust and temporally coherent instance tracks than MaskTrack R-CNN. Such differences in performance are especially noticeable when a video contains large object motion, occlusions, or overlapping objects. More qualitative results are included in our supplementary video[1].

In Figure 5, we also visualize instance-specific features that are propagated from frame t to other frames in the given video clip for two different object instances detected in frame t. Here, we show activations from a randomly selected feature channel. Based on these results, we observe that our MaskProp reliably propagates features that are specific to each instance despite motion blur, object deformations and large variations in object appearance.

6. Conclusion

In this work, we introduced MaskProp, a novel architecture for video instance segmentation. Our method is conceptually simple, it does not require large amounts of labeled data for pre-training, and it produces state-of-the-art results on YouTube-VIS dataset. In future, we plan to extend MaskProp to scenarios where only bounding box annotations are available. We are also interested in applying our method to problems such as pose estimation and tracking.

[1] https://gberta.github.io/maskprop/

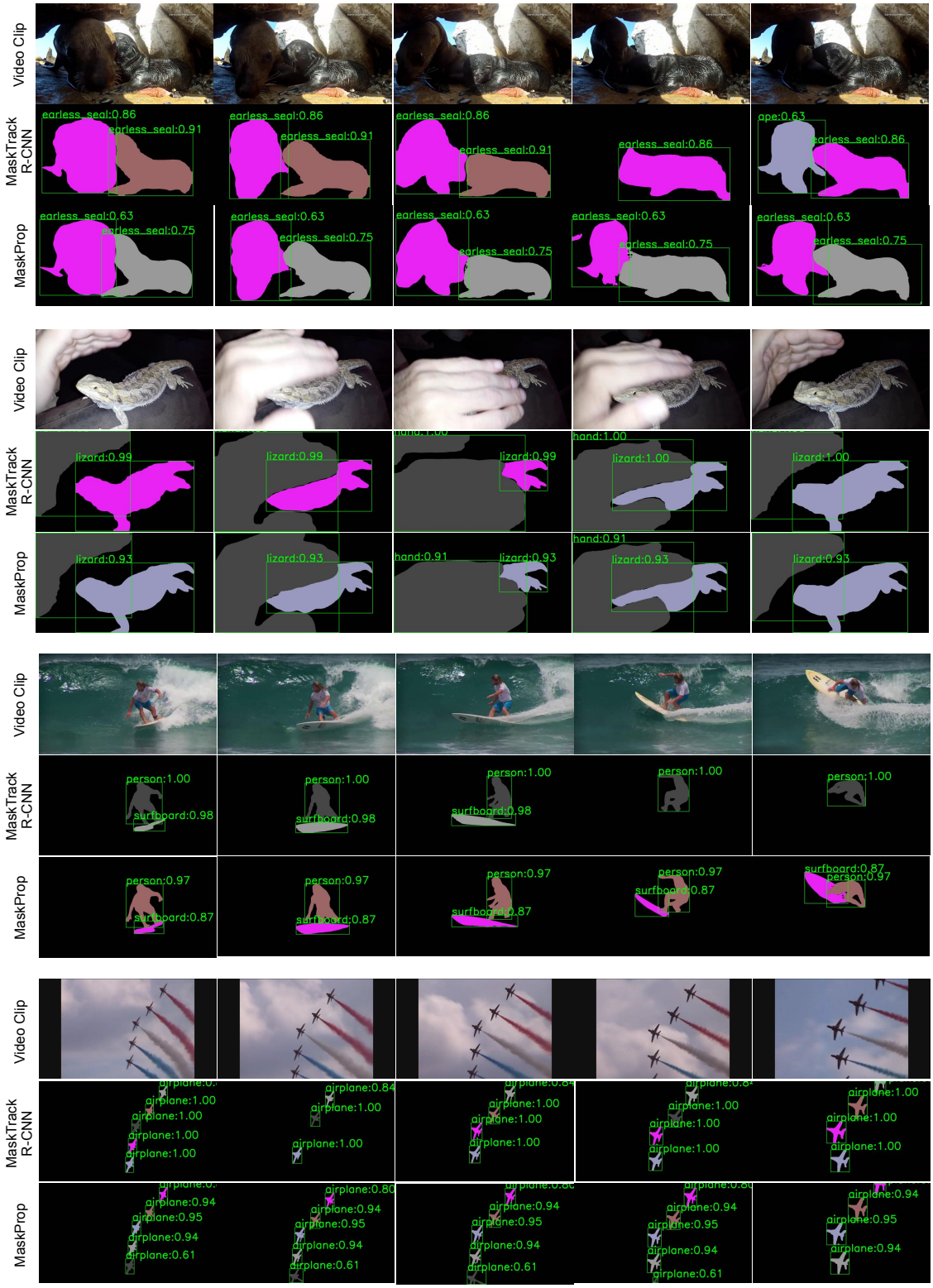

Figure 7: We compare our video instance segmentation results with MaskTrack R-CNN [42] predictions. Different object instances are encoded with different colors. The first row for each video shows the original frames. The second row illustrates the mask predictions of MaskTrack R-CNN and the third row those obtained with our MaskProp. Compared to MaskTrack R-CNN, our MaskProp tracks object instances more robustly even when they are occluded or overlap with each other.

References

[1] Anurag Arnab and Philip H. S. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 879–888, 2017. 2

[2] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2858–2866, July 2017. 2

[3] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely labeled videos. In *Advances in Neural Information Processing Systems 33*. 2019. 4, 6

[4] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Semantic segmentation with boundary neural fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[5] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks. In *ECCV (12)*, volume 11216 of *Lecture Notes in Computer Science*, pages 342–357. Springer, 2018. 2, 4, 6, 8

[6] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[7] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4974–4983. Computer Vision Foundation / IEEE, 2019. 6, 7

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. 1

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 833–851, 2018. 2

[10] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, pages 1189–1198. IEEE Computer Society, 2018. 2

[11] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 2

[12] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, volume 00, pages 764–773, Oct. 2017. 4, 6

[13] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[14] Ross Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015. 1

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 4

[18] Alexander Hermans*, Lucas Beyer*, and Bastian Leibe. In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*, 2017. 2

[19] Alba Herrera-Palacio, Carles Ventura, and Xavier Giro-i Nieto. Video object linguistic grounding. In *1st International Workshop on Multimodal Understanding and Learning for Embodied Applications*, MULEA '19, pages 49–51, New York, NY, USA, 2019. ACM. 2

[20] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *CoRR*, abs/1612.01925, 2016. 8

[21] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2117–2126, 2017. 2

[22] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. InstanceCut: from edges to instances with multicut. In *CVPR*, 2017. 2

[23] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale, 2018. 2, 7, 8

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, Zürich, September 2014. 2, 3, 7

[26] S. Liu, J. Jia, S. Fidler, and R. Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3516–3524, Oct 2017. 2

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[28] Jonathon Luiten, Philip H.S. Torr, and Bastian Leibe. Video instance segmentation 2019: A winning approach for combined detection, segmentation, classification and tracking. In *2019 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, South Korea, October 27-November 2, 2019*. IEEE, 2019. 1, 2, 7, 8

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

[30] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018. 2, 7

[31] Aljoša Ošep, Paul Voigtlaender, Jonathon Luiten, Stefan Breuers, and Bastian Leibe. Large-scale object mining for object discovery from unlabeled video. *ICRA*, 2019. 2

[32] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A.Sorkine-Hornung. Learning video object segmentation from static images. In *Computer Vision and Pattern Recognition*, 2017. 2

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015. 1

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575*, 2014. 2, 7

[35] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. 2018. 2

[36] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4491–4500, Oct 2017. 2

[37] 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 *CVPR*, 2019. 2, 7

[38] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[39] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *CoRR*, abs/1703.07402, 2017. 7

[40] Fanyi Xiao and Yong Jae Lee. Video object detection with an aligned spatial-temporal memory. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[41] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 6

[42] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1, 2, 3, 7, 8, 9

[43] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6499–6507, 2018. 2, 7

[44] Jifeng Dai Xiangyang Ji Yi Li, Haozhi Qi and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 2

[45] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239, 2017. 1

[46] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[47] Idil Esen Zulfikar, Jonathon Luiten, and Bastian Leibe. Unovost: Unsupervised offline video object segmentation and tracking for the 2019 unsupervised davis challenge. 2