

# Learning Video Object Segmentation from Static Images

<sup>\*</sup>Anna Khoreva<sup>3</sup>    <sup>\*</sup>Federico Perazzi<sup>1,2</sup>    Rodrigo Benenson<sup>3</sup>  
 Bernt Schiele<sup>3</sup>    Alexander Sorkine-Hornung<sup>1</sup>

<sup>1</sup>Disney Research    <sup>2</sup>ETH Zurich  
<sup>3</sup>Max Planck Institute for Informatics, Saarbrücken, Germany

## Abstract

*Inspired by recent advances of deep learning in instance segmentation and object tracking, we introduce video object segmentation problem as a concept of guided instance segmentation. Our model proceeds on a per-frame basis, guided by the output of the previous frame towards the object of interest in the next frame. We demonstrate that highly accurate object segmentation in videos can be enabled by using a convnet trained with static images only. The key ingredient of our approach is a combination of offline and online learning strategies, where the former serves to produce a refined mask from the previous' frame estimate and the latter allows to capture the appearance of the specific object instance. Our method can handle different types of input annotations: bounding boxes and segments, as well as incorporate multiple annotated frames, making the system suitable for diverse applications. We obtain competitive results on three different datasets, independently from the type of input annotation.*

## 1. Introduction

Convolutional neural networks (convnets) have shown outstanding performance in many fundamental areas in computer vision, enabled by the availability of large-scale annotated datasets (e.g., ImageNet classification [22, 39]). However, some important challenges in video processing can be difficult to approach using convnets, since creating a sufficiently large body of densely, pixel-wise annotated video data for training is usually prohibitive.

One example domain is video object segmentation. Given only one or a few frames with segmentation mask annotations of a particular object instance, the task is to accurately segment the same instance in all other frames of the video. Current top performing approaches either interleave box tracking and segmentation [48], or propagate the

first frame segment annotation in space-time via CRF-like and GrabCut-like techniques [26, 44].

One of the key insights and contributions of this paper is that fully annotated video data is not necessary. We demonstrate that highly accurate video object segmentation can be enabled using a convnet trained with *static images* only.

We approach video object segmentation from a new angle. We show that a convnet designed for semantic image segmentation [8] can be utilized to perform per-frame instance segmentation, i.e., segmentation of generic objects while distinguishing different instances of the same class. For each new video frame the network is guided towards the object of interest by feeding in the previous' frame mask estimate. We therefore refer to our approach as *guided instance segmentation*. To the best of our knowledge, it represents the first fully trained approach to video object segmentation.

Our system is efficient due to its feed-forward architecture and can generate high quality results in a single pass over the video, without the need for considering more than one frame at a time. This is in stark contrast to many other video segmentation approaches, which usually require global connections over multiple frames or even the whole video sequence in order to achieve coherent results. The method can handle different types of annotations and in the extreme case, even simple bounding boxes as input are sufficient, achieving competitive results, rendering our method flexible with respect to various practical applications.

Key to the video segmentation quality of our approach is a combined offline / online learning strategy. In the offline phase, we use deformation and coarsening on the image masks in order to train the network to produce accurate output masks from their rough estimates. An online training phase extends ideas from previous works on object tracking [12, 29] to video segmentation and enables the method to be easily optimized with respect to an object of interest in a novel input video.

The result is a single, generic system that compares favourably to most classical approaches on three extremely

---

\*The first two authors contributed equally

heterogeneous video segmentation benchmarks, despite using the same model and parameters across all videos. We provide a detailed ablation study and explore the impact of varying number and types of annotations, and moreover discuss extensions of the proposed model, allowing to improve the quality even further.

## 2. Related work

The idea of performing video object segmentation via tracking at the pixel level is at least a decade old [36]. Recent approaches interweave box tracking with box-driven segmentation (e.g. TRS [48]), or propagate the first frame segmentation via graph labeling approaches.

**Local propagation** JOTS [47] builds a graph over neighbour frames connecting superpixels and (generic) object parts to solve the video labeling task. ObjFlow [44] builds a graph over pixels and superpixels, uses convnet based appearance terms, and interleaves labeling with optical flow estimation. Instead of using superpixels or proposals, BVS [26] formulates a fully-connected pixel-level graph between frames and efficiently infer the labeling over the vertices of a spatio-temporal bilateral grid [7]. Because these methods propagate information only across neighbor frames they have difficulties capturing long range relationships and ensuring globally consistent segmentations.

**Global propagation** In order to overcome these limitations, some methods have proposed to use long-range connections between video frames [14, 23, 50]. In particular, we compare to FCP [31], Z15 [51] and W16 [45] which build a global graph structure over object proposal segments, and then infer a consistent segmentation. A limitation of methods utilizing long-range connections is that they have to operate on larger image regions such as superpixels or object proposals for acceptable speed and memory usage, compromising on their ability to handle fine image detail.

**Unsupervised segmentation** Another family of work does general moving object segmentation (over all parts of the image), and selects post-hoc the space-time tube that best match the annotation, e.g. NLC [14] and [16, 24, 48].

In contrast, our approach sides-steps the use of any intermediate tracked box, superpixels, or object proposals and proceeds on a per-frame basis, therefore efficiently handling even long sequences at full detail. We focus on propagating the first frame segmentation forward onto future frames, using an online fine-tuned convnet as appearance model for segmenting the object of interest in the next frames.

**Box tracking** Some previous works have investigated approaches that improve segmentation quality by leveraging object tracking and vice versa [10, 13, 36, 48]. More recent, state-of-the-art tracking methods are based on discriminative correlation filters over handcrafted features (e.g. HOG)

and over frozen deep learned features [11, 12], or are convnet based trackers on their own right [18, 29].

Our approach is most closely related to the latter group. GOTURN [18] proposes to train offline a convnet so as to directly regress the bounding box in the current frame based on the object position and appearance in the previous frame. MDNet [29] proposes to use online fine-tuning of a convnet to model the object appearance.

Our training strategy is inspired by GOTURN for the offline part, and MDNet for the online stage. Compared to the aforementioned methods our approach operates at pixel level masks instead of boxes. Differently from MDNet, we do not replace the domain-specific layers, instead finetuning all the layers on the available annotations for each individual video sequence.

**Instance segmentation** At each frame, video object segmentation outputs a single instance segmentation. Given an estimate of the object location and size, bottom-up segment proposals [34] or GrabCut [38] variants can be used as shape guesses. Also specific convnet architectures have been proposed for instance segmentation [17, 32, 33, 49]. Our approach outputs per-frame instance segmentation using a convnet architecture, inspired by works from other domains like [6, 40, 49]. A concurrent work [5] also exploits convnets for video object segmentation. Differently from our approach their segmentation is not guided, which might result in performance decay over time. Furthermore, the offline training exploits full video sequence annotations that are notoriously difficult to obtain.

One-shot

**Interactive video segmentation** Applications such as video editing for movie production often require a level of accuracy beyond the current state-of-the-art. Thus several works have also considered video segmentation with variable annotation effort, enabling human interaction using clicks [20, 43, 46], or strokes [1, 15, 52]. In this work we consider instead box or (full) segment annotations on multiple frames. In §5 we report results when varying the amount of annotation effort (from one frame per video to all frames).

## 3. MaskTrack method

We approach the video object segmentation problem from a new angle we refer as *guided instance segmentation*. For each new frame we wish to label pixels as object/non-object of interest, for this we build upon the architecture of the existing pixel labelling convnet and train it to generate per-frame instance segments. We pick DeepLabv2 [8], but our approach is agnostic of the specific architecture selected.

The challenge is then: how to inform the network which instance to segment? We solve this by using two complementary strategies. One is guiding the network towards the

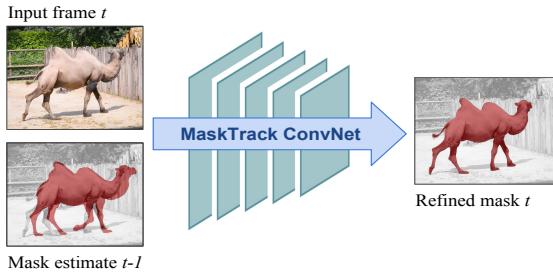


Figure 1: Given a rough mask estimate from the previous frame  $t - 1$ , we train a convnet to provide a refined mask output for the current frame  $t$ .

instance of interest by feeding in the previous' frame mask estimate during offline training (§3.1). And a second is employing online training (§3.2) to fine-tune the model to become more specialized for the specific instance.

### 3.1. Learning to segment instances offline

In order to guide the pixel labeling network to segment the object of interest, we begin by expanding the convnet input from RGB to RGB+mask channel (4 channels). The extra mask channel is meant to provide an estimate of the visible area of the object in the current frame, its approximate location and shape. We can then train the labelling convnet to provide as output an accurate segmentation of the object, given as input the current image and a rough estimate of the object mask. Our tracking network is de-facto a "mask refinement" network.

There are two key observations that make this approach practical. First, very rough input masks are enough for our trained network to provide sensible output segments. Even a large bounding box as input will result in a reasonable output (see §5.2). The input mask's main role is to point the convnet towards the correct object instance to segment.

Second, this particular approach does not require us to use video as training data, such as done in [3, 18, 29]. Because we only use a mask as additional input, instead of an image crop as in [3, 18], we can easily synthesize training samples from single frame instance segmentation annotations. This allows to train from a large set of diverse images and avoids having to use existing (scarce and small) video segmentation benchmarks for training.

Figure 1 shows our overall architecture. To simulate the noise in the previous frame output masks, during offline training we generate input masks by deforming the annotated masks via affine transformation as well as non-rigid deformations via thin-plate splines [4], followed by a coarsening step (dilation morphological operation) to remove details of the object contour. We apply this data generation procedure over a dataset of  $\sim 10^4$  images containing diverse object instances, see examples in Figure 2. At test



Figure 2: Examples of training mask generation. From one annotated image, multiple training masks are generated. The generated masks mimic plausible object shapes on the preceding frame.

time, given the mask estimate at time  $t - 1$ , we apply the dilation operation and use the resulting rough mask as input for object segmentation in frame  $t$ .

The affine transformations and non-rigid deformations aim at modelling the expected motion of an object between two frames. The coarsening permits us to generate training samples that resembles the test time data, simulating the blobby shape of the output mask given from the previous frame by the convnet. These two ingredients make the estimation more robust and help to avoid accumulation of errors from the preceding frames.

After training the resulting convnet has learnt to do guided instance segmentation, similar to networks like DeepMask [32] and Hypercolumns [17], but instead of taking a bounding box as guidance, we can use an arbitrary input mask. The training details are described in §4.

When using offline training only, the segmentation procedure consists of two steps: the previous frame mask is coarsened and then fed into the trained network to estimate the current frame mask. Since objects have a tendency to move smoothly through space, the object mask in the preceding frame will provide a good guess in the current frame and simply copying the coarse mask from the previous frame is enough. This approach is fast and already provides good results. We also experimented using optical flow to propagate the mask from one frame to the next, but found the optical flow errors to offset the gains.

With only the offline trained network, the proposed approach allows to achieve competitive performance compared to previously reported results (see §5.2). However the performance can be further improved by integrating online training strategy.

### 3.2. Learning to segment instances online

For further boosting the video segmentation quality, we borrow and extend ideas that were originally proposed for tracking. Current top performing tracking techniques [12, 29] all use some form of online training. We thus consider improving results by adding this as a second strategy.

The idea is to use, at test time, the segment annotation of the first video frame as additional training data. Using aug-

mented versions of this single frame annotation, we proceed to fine-tune the model to become more specialized for the specific object instance at hand.

We use a similar data augmentation as for offline training. On top of affine and non-rigid deformations for the input mask, we also add image flipping and rotations to generate multiple training samples from one frame. We generate  $\sim 10^3$  training samples from this single annotation, and proceed to fine-tune the model previously trained offline.

With online fine-tuning, the network weights partially capture the appearance of the specific object being tracked. The model aims to strike a balance between general instance segmentation (so as to generalize to the object changes), and specific instance segmentation (so as to leverage the common appearance across video frames). The details of the online fine-tuning are provided in §4. In our experiments we only do fine-tuning using the annotated frame(s).

To the best of our knowledge our approach is the first to use a pixel labelling network (like DeepLabv2 [8]) for the task of video object segmentation. We name our full approach (using both offline and online training) **MaskTrack**.

### 3.3. Variants

Additionally we consider variations of the proposed model. First, we want to show that our approach is flexible and could handle different types of input annotations, using less supervision in the first frame annotation. Second, motion information could be easily integrated in the system, improving the quality of the object segments.

**Box annotation** Here we discuss a variant named MaskTrack<sub>Box</sub>, that takes a bounding box annotation in the first frame as an input supervision instead of a segmentation mask. To handle this variant we use on the first frame a second convnet model trained with bounding box rectangles as input masks. From the next frame onwards we use the standard MaskTrack model.

**Optical flow** On top of MaskTrack, we consider to employ optical flow as a source of additional information to guide the segmentation. Given a video sequence, we compute the optical flow using EpicFlow [37] with Flow Fields matches [2] and convolutional boundaries [27]. In parallel to the vanilla MaskTrack, we proceed to compute a second output mask using the magnitude of the optical flow field as input image (replicated into a three channel image). The model is used as-is, without retraining. Although it has been trained on RGB images, this strategy works because object flow magnitude roughly looks like a gray-scale object, and still captures useful object shape information, see examples in Figure 3. Using the RGB model allows to avoid training the convnet on a video dataset with segmentation annotations.

We then fuse by averaging the output scores given by the two parallel networks (using RGB image and optical flow magnitude as inputs). We name this variant

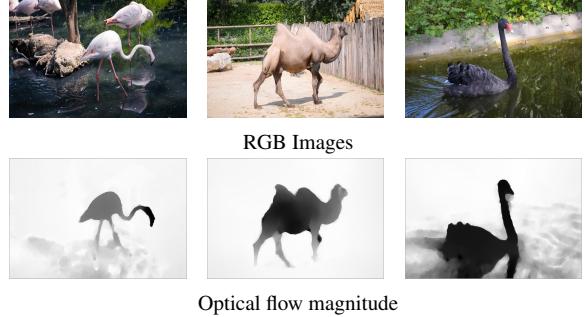


Figure 3: Examples of optical flow magnitude images.

**MaskTrack+Flow.** Optical flow provides complementary information to the MaskTrack with RGB images, improving the overall performance.

## 4. Network implementation and training

Following, we describe the implementation details of our approach, namely the offline and online training strategy and the data augmentation.

**Network** For all our experiments we use the training and test parameters of DeepLabv2-VGG network [8]. The model is initialized from a VGG16 network pre-trained on ImageNet [42]. For the extra mask channel of filters in the first convolutional layer we use gaussian initialization. We also tried zero initialization, but observed no difference.

**Offline training** The advantage of our method is that it does not require expensive pixel-label annotations on videos for training the convnet. Thus we can employ images and annotations from existing saliency segmentation datasets. We consider images and segmentation masks from ECSSD [41], MSRA10K [9], SOD [28], and PASCAL-S [25]. This results in an aggregate set of 11 282 training images.

The input masks for an extra channel are generated by deforming the binary segmentation masks via affine transformation and non-rigid deformations, as discussed in §3.1. For affine transformation we consider random scaling ( $\pm 5\%$  of object size) and translation ( $\pm 10\%$  shift). Non-rigid deformations are done via thin-plate splines [4] using 5 control points and randomly shifting the points in x and y directions within  $\pm 10\%$  margin of the original segmentation mask width and height. Then the mask is coarsened using dilation operation with 5 pixel radius. This mask deformation procedure is applied over all object instances in the training set. For each image two different masks are generated, see examples in Figure 2.

For training we follow [8] and use SGD with mini-batches of 10 images and a polynomial learning policy with initial learning rate of 0.001. The momentum and weight decay are set to 0.9 and 0.0005, respectively. The network is trained for 20k iterations.

**Online training** For online adaptation we finetune the model previously trained offline on the first frame for 200 iterations with training samples generated from the first frame annotation. We augment the first frame by image flipping and rotations as well as by deforming the annotated masks for an extra channel via affine and non-rigid deformations with the same parameters as for the offline training. This results in an augmented set of  $\sim 10^3$  training images.

The network is trained with the same learning parameters as for offline training, finetuning all convolutional and fully connected layers.

At test time our base MaskTrack system runs at about 12 seconds per frame (average over DAVIS dataset, amortizing the online fine-tuning time over all video frames), which is a magnitude faster compared to ObjFlow [44] (takes 2 minutes per frame, averaged over DAVIS dataset).

## 5. Results

In this section we describe our evaluation protocol (§5.1), study the quantitative importance of the different components of our system (§5.2), and report results comparing to state of art techniques over three datasets (190 videos total, §5.3), as well as comparing the effects of different amounts of annotation on final quality (§5.4). Additional quantitative and qualitative results are provided in the supplementary material.

### 5.1. Experimental setup

**Datasets** We evaluate the proposed approach on three different video object segmentation datasets: DAVIS [30], YoutubeObjects [35], and SegTrack-v2 [24]. These datasets include assorted challenges such as appearance change, occlusion, motion blur and shape deformation.

DAVIS [30] consists of 50 high quality videos, totalling 3 455 frames. Pixel-level segmentation annotations are provided for each frame, where one single object or two connected objects are separated from the background.

YoutubeObjects [35] includes videos with 10 object categories. We consider the subset of 126 videos with more than 20 000 frames, for which the pixel-level ground truth segmentation masks are provided by [19].

SegTrack-v2 [24] contains 14 video sequences with 24 objects and 947 frames. Every frame is annotated with a pixel-level object mask. As instance-level annotations are provided for sequences with multiple objects, each specific instance segmentation is treated as separate problem.

**Evaluation** We evaluate using the standard mIoU metric: intersection-over-union of the estimated segmentation and the ground truth binary mask, also known as *Jaccard Index*, averaged across videos. For DAVIS we use the provided benchmark code [30], which excludes the first and the last frames from the evaluation. For YoutubeObjects and SegTrack-v2 only the first frame is excluded.

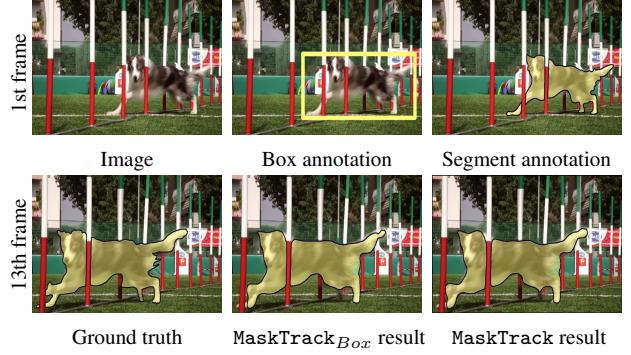


Figure 4: By propagating annotation from the 1st frame, either from segment or just bounding box annotations, our system generates results comparable to ground truth.

Previous work used diverse evaluations procedure. To ensure a consistent comparison between methods, when needed, we re-computed scores for using shared results maps, or by generating them using available open source code. In particular, we collected new results for ObjFlow [44] and BVS [26] in order to present other methods with results across the three datasets.

### 5.2. Ablation study

We first study different ingredients of our method. We experiment on the DAVIS dataset and measure the performance using the mean intersection-over-union metric (mIoU). Table 1 shows the importance of each of the ingredients described in §3 and reports the improvement of adding extra components to the MaskTrack model.

**Add-ons** We first study the effect of adding a couple of ingredients on top of our base MaskTrack system, which are specifically fine-tuned for DAVIS. We see that optical flow provides complementary information to the appearance, boosting further the results ( $74.8 \rightarrow 78.4$ ). Adding on top a well-tuned post-processing CRF [21] can gain a couple of mIoU points, reaching 80% mIoU on DAVIS, the best known result on this dataset.

Albeit optical flow can provide interesting gains, we found it to be brittle when going across different datasets. Different strategies to handle optical flow provide 1~4% on each dataset, but none provide consistent gains across all datasets; mainly due to failure modes of the optical flow algorithms. For the sake of presenting a single model with fix parameters across all datasets, we refrain from using a per-dataset tuned optical flow in the results of §5.3.

**Training** We next study the effect of offline/online training of the network. By disabling online fine-tuning, and only relying on offline training we see a  $\sim 5$  IoU percent points drop, showing that online fine-tuning indeed expand the tracking capabilities. If instead we skip offline training

and only rely on online fine-tuning performance drop drastically, albeit the absolute quality (57.6 mIoU) is surprisingly high for a system trained on ImageNet+single frame.

By reducing the amount of training data from 11k to 5k we only see a minor decrease in mIoU; this indicates that the amount of training data is not critical to get reasonable performance. That being said, further increase of the training data volume would lead to improved results.

Additionally, we explore the effect of the offline training on video data instead of using static images. We train the model on the annotated frames of two combined datasets, SegTrack-v2 and YoutubeObjects. By switching to train on video data we observe a minor decrease in mIoU; this could be explained by lack of diversity in the video training data due to the small scale of the existing datasets, as well as the effect of the domain shift between different benchmarks. This shows that employing static images in our approach does not result in any performance drop.

**Mask deformation** We also study the influence of mask deformations. We see that coarsening the mask via dilation provides a small gain, as well as adding non-rigid deformations. All-and-all, Table 1 shows that the main factor affecting the quality is using any form of mask deformations when creating the training samples (both for offline and online training). This ingredient is critical for our overall approach, making the segmentation estimation more robust at test time to the noise in the input mask.

**Input channel** Next we experiment with different variants of the extra channel input. Even by changing the input from segments to boxes, a model trained for this modality still provides reasonable results.

Most interestingly we also evaluated a model that does not use any mask input. Without the additional input channel, this pixel labelling convnet was trained for saliency offline and fine-tuned online to capture the appearance of the object of interest. This model obtains competitive results (72.5 mIoU), showing the power of modelling the video segmentation task as an instance segmentation problem.

### 5.3. Single frame annotations

Table 2 presents results when the first frame is annotated with an object segmentation mask. This is the protocol commonly used on DAVIS, SegTrack-v2, andYoutubeObjects.

We see that MaskTrack obtains competitive performance across all three datasets. This is achieved using our purely frame-by-frame feed-forward system, using the exact same model and parameters across all datasets. Our MaskTrack results are obtained in a single pass, do not use any global optimization, not even optical flow. We believe this shows the promise of formulating video object segmentation from the instance segmentation perspective.

On SegTrack-v2, JOTS [47] reported higher numbers (71.3 mIoU), however, they report tuning their method pa-

Aspect	System variant	mIoU	$\Delta$ mIoU
Add-ons	MaskTrack+Flow+CRF	<b>80.3</b>	+1.9
	MaskTrack+Flow	78.4	+3.6
	MaskTrack	74.8	-
Training	No online fine-tuning	69.9	-4.9
	No offline training	57.6	-17.2
	Reduced offline training	73.2	-1.6
Mask	Training on video	72.0	-2.8
	No dilation	72.4	-2.4
	No deformation	17.1	-57.7
deformation	No non-rigid deformation	73.3	-1.5
	Boxes	69.6	-5.2
	No input	72.5	-2.3

Table 1: Ablation study of our MaskTrack method on DAVIS. Given our full system, we change one ingredient at a time, to see each individual contribution. See §5.2 for discussion.

Method	Dataset, mIoU		
	DAVIS	YoutbObjs	SegTrack-v2
Box oracle	45.1	55.3	56.1
Grabcut oracle	67.3	67.6	74.2
ObjFlow [44]	71.4	70.1	67.5
BVS [26]	66.5	59.7	58.4
NLC [14]	64.1	-	-
FCP [31]	63.1	-	-
W16 [45]	-	59.2	-
Z15 [51]	-	52.6	-
TRS [48]	-	-	<b>69.1</b>
MaskTrack	<b>74.8</b>	<b>71.7</b>	67.4
MaskTrack <sub>Box</sub>	73.7	69.3	62.4

Table 2: Video object segmentation results on three datasets. Compared to related state of the art, our approach provides consistently good results. On DAVIS the extended version of our system MaskTrack+Flow+CRF reaches 80.3 mIoU. See §5.3 for details.

rameters’ per video, and thus it is not comparable to our setup with fix-parameters.

Table 2 also reports results for the MaskTrack<sub>Box</sub> variant described in §3.3. Starting only from box annotations on the first frame, our system still generates comparably good results (see Figure 4), remaining on the top three best results in all the datasets covered.

By adding additional ingredients specifically tuned for different datasets, such as optical flow (see §3.3) and CRF post-processing, we can push the results even further, reaching 80.3 mIoU on DAVIS, 72.6 onYoutubeObjects and 70.3 on SegTrack-v2. The dataset specific tuning is described in the supplementary material.

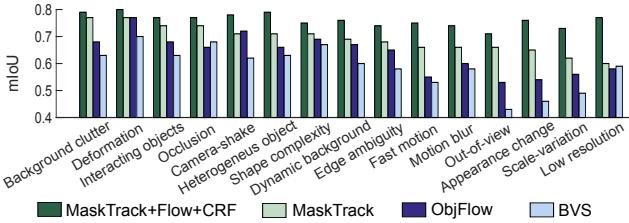


Figure 5: Attribute based evaluation on DAVIS.

Figure 6 presents qualitative results of the proposed MaskTrack model across three different datasets.

**Attribute-based analysis** Figure 5 presents a more detailed evaluation on DAVIS [30] using video attributes. The attribute based analysis shows that our generic model, MaskTrack, is robust to various video challenges present in DAVIS. It compares favourably on any subset of videos sharing the same attribute, except camera-shake, where ObjFlow [44] marginally outperforms our approach. We observe that MaskTrack handles fast-motion and motion-blur well, which are typical failure cases for methods relying on spatio-temporal connections [26, 44].

Due to the online fine-tuning on the first frame annotation of a new video, our system is able to capture the appearance of the specific object of interest. This allows it to better recover from occlusions, out-of-view scenarios and appearance changes, which usually affect methods that strongly rely on propagating segmentations on a per-frame basis.

Incorporating optical flow information into MaskTrack substantially increases robustness on all categories. As one could expect, MaskTrack+Flow+CRF better discriminates cases involving color ambiguity and salient motion. However, we also observed less-obvious improvements in cases with scale-variation and low-resolution objects.

**Conclusion** With our simple, generic system for video object segmentation we are able to achieve competitive results with existing techniques, on three different datasets. These results are obtained with fixed parameters, from a forward-pass only, using only static images during offline training. We also reach good results even when using only a box annotation as starting point.

#### 5.4. Multiple frames annotations

In some applications, e.g. video editing for movie production, one may want to consider more than a single frame annotation on videos. Figure 7 shows the video object segmentation quality result when considering different number of annotated frames, on the DAVIS dataset. We show results for both pixel accurate segmentation per frame, or bounding box annotation per frame.

For these experiments, we run our method twice, forward and backwards; and for each frame pick the result closest in

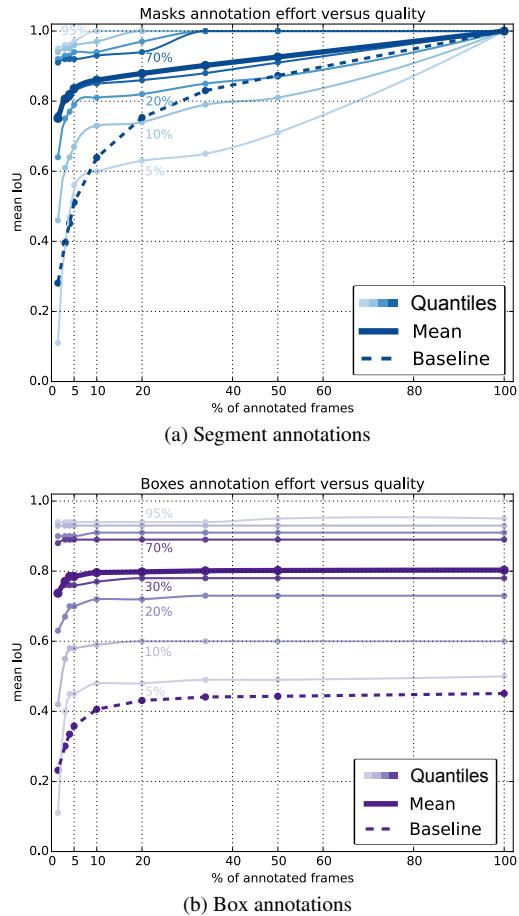


Figure 7: Percent of annotated frames versus video object segmentation quality. We report mean IoU, and quantiles at 5, 10, 20, 30, 70, 80, 90, and 95%. Result on DAVIS dataset, using segment or box annotations. The baseline simply copies the annotations to adjacent frames. Discussion in §5.4.

time to the annotated frame (either from forward or backwards propagation). Here, the online fine-tuning uses all annotated frames instead of only the first one. For the experiments with box annotations (Figure 7b), we use a similar procedure to  $\text{MaskTrack}_{\text{Box}}$ . Box annotations are first converted to segments, and then apply MaskTrack as-is, treating these as the original segment annotations.

The evaluation reports the mean IoU results when annotating one frame only (same as table 2), and every 40th, 30th, 20th, 10th, 5th, 3rd, and 2nd frame. Since DAVIS videos have length  $\sim 100$  frames, 1 annotated frame corresponds to  $\sim 1\%$ , otherwise annotations every 20th is 5% of annotated frames, 10th 10%, 5th 20%, etc. We follow the same DAVIS evaluation protocol as §5.3, ignoring first and last frame, and choosing to include the annotated frames in the evaluation (this is particularly relevant for the box anno-

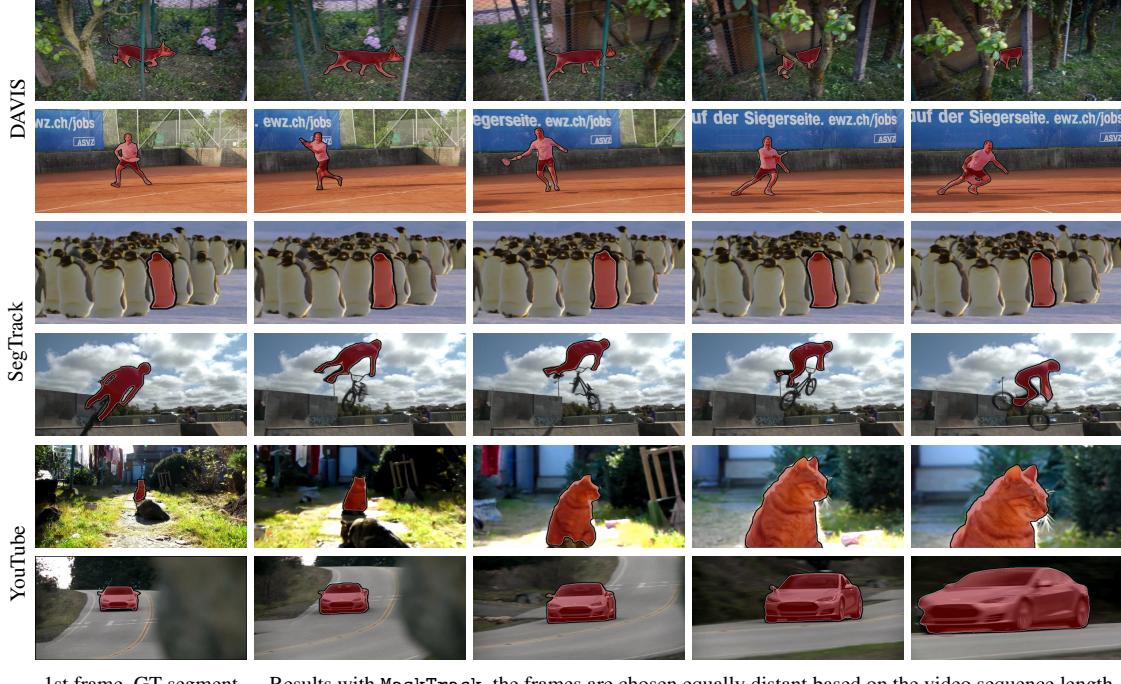


Figure 6: Qualitative results of three different datasets. Our algorithm is robust to challenging situations such as occlusions, fast motion, multiple instances of the same semantic class, object shape deformation, camera view change and motion blur.

tation results).

Other than mean IoU we also show the quantile curves indicating the cutting line for the 5%, 10%, 20%, etc. lowest quality video frame results. This gives a hint of how much targeted additional annotations might be needed. The higher mean IoU of these quantiles are, the better.

The baseline for these experiments consists in directly copying the ground truth annotations from the nearest annotated neighbour. This baseline indicates "level zero" for our results. For visual clarity, we only include the mean value for the baseline.

**Analysis** We can see that Figures 7a and 7b show slightly different trends. When using segment annotations (Figure 7a), the baseline quality increases steadily until reaching IoU 1 when all frames are annotated. Our MaskTrack approach provides large gains with 30% of annotated frames or less. For instance when annotating 10% of the frames we reach mIoU 0.86, notice that also the 20% quantile is at 0.81 mIoU. This means with only 10% of annotated frames, 80% of all video frames will have a mean IoU above 0.8, which is good enough to be used for many applications, or can serve as initialization for a refinement process. With 10% of annotated frames the baseline only reaches 0.64 mIoU. When using box annotations (Figure 7b) the quality of the baseline and our method saturates. There is only so much information our instance segmenter can esti-

mate from boxes. After 10% of annotated frames, not much additional gain is obtained. Interestingly, the mean IoU and 30% quantile here both reach  $\sim 0.8$  mIoU range. Additionally, 70% of the frames have IoU above 0.89.

**Conclusion** Results indicate that with only 10% of annotated frames we can reach satisfactory quality, even when using only bounding box annotations. We see that with moving from one annotation per video to two or three frames ( $1\% \rightarrow 3\% \rightarrow 4\%$ ) quality increases sharply, showing that our system can adequately leverage a few extra annotations per video.

## 6. Conclusion

We have presented a novel approach to video object segmentation. By treating video object segmentation as a guided instance segmentation problem, we have proposed to use a pixel labelling convnet for frame-by-frame segmentation. By exploiting both offline and online training with image annotations only our approach is able to produce highly accurate video object segmentation. The proposed system is generic and reaches competitive performance on three extremely heterogeneous video segmentation benchmarks, using the same model and parameters across all videos. The method can handle different types of input annotations and our results are competitive even when using only bounding box annotations (instead of segmentation masks).

We provided a detailed ablation study, and explored the effect of varying the amount of annotations per video. Our results show that with only one annotation every 10th frame we can reach 85% mIoU quality. Considering we only do per-frame instance segmentation without any form of global optimization, we deem these results encouraging to achieve high quality via additional post-processing.

We believe the use of labelling convnets for video object segmentation is a promising strategy. Future work should consider exploring more sophisticated network architectures, incorporating temporal dimension and adding global optimization strategies.

## References

- [1] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *TOG*, 2009. [2](#)
- [2] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015. [4, 12](#)
- [3] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. *arXiv preprint arXiv:1606.09549*, 2016. [3](#)
- [4] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI*, 1989. [3, 4](#)
- [5] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool. One-shot video object segmentation. *arXiv:1611.05198*, 2016. [2](#)
- [6] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *CVPR*, 2016. [2](#)
- [7] J. Chen, S. Paris, and F. Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graph.*, 26(3), July 2007. [2](#)
- [8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. [1, 2, 4, 12](#)
- [9] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *TPAMI*, 2015. [4](#)
- [10] P. Chockalingam, S. N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *ICCV*, 2009. [2](#)
- [11] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCV workshop*, pages 58–66, 2015. [2](#)
- [12] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*. Springer, 2016. [1, 2, 3](#)
- [13] S. Duffner and C. Garcia. Pixeltrack: A fast adaptive algorithm for tracking non-rigid objects. In *ICCV*, 2013. [2](#)
- [14] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. [2, 6](#)
- [15] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: Non-successive mask transfer and interpolation for video cutout. *SIGGRAPH Asia*, 2015. [2](#)
- [16] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. [2](#)
- [17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. [2, 3](#)
- [18] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016. [2, 3](#)
- [19] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014. [5](#)
- [20] S. D. Jain and K. Grauman. Click carving: Segmenting objects in video with point clicks. In *HCOMP*, 2016. [2](#)
- [21] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. [5, 12](#)
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. [1](#)
- [23] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Proc. ICCV*, 2011. [2](#)
- [24] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. [2, 5, 11](#)
- [25] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. [4](#)
- [26] N. Maerki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. [1, 2, 5, 6, 7, 11, 12, 13](#)
- [27] K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool. Convolutional oriented boundaries. In *ECCV*, 2016. [4, 12](#)
- [28] V. Movahedi and J. H. Elder. Design and perceptual validation of performance measures for salient object segmentation. In *CVPR Workshops*, 2010. [4](#)
- [29] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. [1, 2, 3](#)
- [30] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. [5, 7, 11](#)
- [31] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, 2015. [2, 6](#)
- [32] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *NIPS*, 2015. [2, 3](#)
- [33] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016. [2](#)
- [34] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *PAMI*, 2016. [2](#)
- [35] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. [5, 11](#)
- [36] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, 2007. [2](#)

- [37] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. [4](#), [12](#)
- [38] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Trans. Graphics*, 2004. [2](#)
- [39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. [1](#)
- [40] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs. Automatic portrait segmentation for image stylization. In *Computer Graphics Forum*, 2016. [2](#)
- [41] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical image saliency detection on extended CSSD. *TPAMI*, 2016. [4](#)
- [42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. [4](#)
- [43] T. V. Spina and A. X. Falcão. Fomtrace: Interactive video segmentation by image graphs and fuzzy object models. *arXiv preprint arXiv:1606.03369*, 2016. [2](#)
- [44] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. [1](#), [2](#), [5](#), [6](#), [7](#), [11](#), [12](#), [13](#)
- [45] H. Wang, T. Raiko, L. Lensu, T. Wang, and J. Karhunen. Semi-supervised domain adaptation for weakly labeled semantic video object segmentation. In *ACCV*, 2016. [2](#), [6](#)
- [46] T. Wang, B. Han, and J. Collomosse. Touchcut: Fast image and video segmentation using single-touch interaction. *CVIU*, 2014. [2](#)
- [47] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. Jots: Joint online tracking and segmentation. In *CVPR*, 2015. [2](#), [6](#)
- [48] F. Xiao and Y. J. Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016. [1](#), [2](#), [6](#), [12](#)
- [49] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *CVPR*, 2016. [2](#)
- [50] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. [2](#)
- [51] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia. Semantic object segmentation via detection in weakly labeled video. In *CVPR*, 2015. [2](#), [6](#)
- [52] F. Zhong, X. Qin, Q. Peng, and X. Meng. Discontinuity-aware video object cutout. *TOG*, 2012. [2](#)

# Supplementary material

## A. Content

This supplementary material provides both additional quantitative and qualitative results, as well as an attached video.

- Section B provides additional quantitative results for DAVIS, YoutubeObjects, and SegTrackv-2 (see Tables S5 - S4).
- Detailed attribute-based evaluation is reported in Section C and Table S6.
- The dataset specific tuning for additional ingredients is described in Section D.
- Additional qualitative results with first frame box and segment supervision are presented in Section E and Figure 6.
- Examples of mask generation for the extra input channel are shown in Section F and Figure 2.
- Examples of optical flow magnitude images are presented in Section G and Figure 3.

## B. Additional quantitative results

In this section we present additional quantitative results for three different datasets: DAVIS [30], YoutubeObjects [35], and SegTrackv-2 [24]. This section complements §5.3 in the main paper.

**DAVIS** We present the per-sequence comparison with other state-of-the-art methods on DAVIS in Table S3.

**SegTrack-v2** Table S4 reports the per-sequence comparison with other state-of-the-art methods on SegTrack-v2.

**YoutubeObjects** The per-category comparison with other state-of-the-art methods on YoutubeObjects is shown in Table S5.

## C. Attribute-based evaluation

Table S6 presents a more detailed evaluation on DAVIS using video attributes and complements Figure 5 in the main paper.

The attribute based evaluation shows that our generic model, MaskTrack, is robust to various video challenges present in DAVIS. It compares favourably on any subset of videos sharing the same attribute, except camera-shake, where ObjFlow [44] marginally outperforms our approach.

We observe that MaskTrack handles well fast-motion, appearance change and out-of-view, where competitive methods are failing [26, 44].

Sequence	Method, mIoU			
	BVS [26]	ObjFlow [44]	MaskTrack	+Flow+CRF
bear	<b>95.5</b>	94.6	92.8	93.1
blackswan	94.3	<b>94.7</b>	91.9	90.3
bmx-bumps	43.4	48.0	39.6	<b>57.1</b>
bmx-trees	38.2	14.9	32.1	<b>57.5</b>
boat	64.4	<b>80.8</b>	78.2	54.7
breakdance	50.0	49.6	59.4	<b>76.1</b>
breakdance-flare	72.7	76.5	<b>89.2</b>	77.6
bus	86.3	68.2	79.1	<b>89.0</b>
camel	66.9	<b>86.7</b>	80.4	80.1
car-roundabout	85.1	90.0	82.8	<b>96.0</b>
car-shadow	57.8	84.6	90.3	<b>93.5</b>
car-turn	84.4	87.6	<b>92.3</b>	88.6
cows	89.5	91.0	<b>91.9</b>	88.2
dance-jump	74.5	<b>80.4</b>	66.2	78.8
dance-twirl	49.2	56.7	67.8	<b>84.4</b>
dog	72.3	89.7	86.8	<b>90.8</b>
dog-agility	34.5	<b>86.0</b>	83.7	78.9
drift-chicane	3.3	17.5	0.5	<b>86.2</b>
drift-straight	40.2	31.4	46.0	<b>56.0</b>
drift-turn	29.9	3.5	<b>86.5</b>	86.0
elephant	84.9	87.9	<b>91.4</b>	87.2
flamingo	<b>88.1</b>	87.3	70.9	79.0
goat	66.1	<b>86.5</b>	85.8	84.5
hike	75.5	<b>93.4</b>	74.5	93.1
hockey	82.9	<b>85.0</b>	84.0	83.4
horsejump-high	80.1	<b>86.2</b>	78.4	81.8
horsejump-low	60.1	<b>82.2</b>	79.6	80.6
kite-surf	42.5	<b>70.2</b>	58.7	60.0
kite-walk	<b>87.0</b>	85.0	77.4	64.5
libby	77.6	59.4	<b>78.8</b>	77.5
lucia	90.1	89.7	88.4	<b>91.1</b>
mallard-fly	<b>60.6</b>	55.0	56.7	57.2
mallard-water	90.7	89.9	<b>91.0</b>	90.4
motocross-bumps	40.1	48.5	53.9	<b>59.9</b>
motocross-jump	34.1	59.4	<b>69.0</b>	68.3
motorbike	56.3	47.8	46.5	<b>56.7</b>
paragliding	87.5	94.7	93.2	<b>95.9</b>
paragliding-launch	<b>64.0</b>	63.7	58.9	62.1
parkour	75.6	<b>86.1</b>	85.3	88.2
rhino	78.2	89.5	<b>93.2</b>	91.1
rollerblade	58.8	<b>88.6</b>	33.0	78.7
scooter-black	33.7	76.5	64.9	<b>82.4</b>
scooter-gray	50.8	29.6	81.7	<b>82.9</b>
soapbox	78.9	68.9	86.1	<b>89.9</b>
soccerball	84.4	8.0	85.8	<b>89.0</b>
stroller	76.7	<b>87.7</b>	86.2	85.4
surf	49.2	<b>95.6</b>	92.6	92.8
swing	78.4	60.4	80.7	<b>81.9</b>
tennis	73.7	81.8	<b>87.3</b>	86.2
train	87.2	<b>91.7</b>	90.8	90.4
Mean	66.5	71.1	74.8	<b>80.3</b>

Table S3: Per-sequence results on the DAVIS dataset.

Sequence	Method, mIoU				Category	Method, mIoU			
	BVS [26]	ObjFlow [44]	TRS [48]	MaskTrack		BVS [26]	ObjFlow [44]	TRS [48]	MaskTrack
bird of paradise	89.7	87.1	<b>90.0</b>	84.0	aeroplane	80.8	<b>85.3</b>	81.6	
birdfall	65.3	52.9	<b>72.5</b>	56.6	bird	76.4	<b>83.1</b>	82.9	
bmx#1	67.1	<b>87.9</b>	86.1	81.9	boat	60.1	70.6	<b>74.7</b>	
bmx#2	3.2	4.0	<b>40.3</b>	0.1	car	56.7	<b>68.8</b>	66.9	
cheetah#1	5.4	25.9	61.2	<b>69.3</b>	cat	52.7	60.6	<b>69.6</b>	
cheetah#2	9.2	37.2	<b>39.4</b>	17.4	cow	64.8	71.5	<b>75.0</b>	
drift#1	68.5	<b>77.9</b>	70.7	47.4	dog	61.6	71.6	<b>75.2</b>	
drift#2	32.7	27.4	70.7	<b>70.9</b>	horse	53.1	62.3	<b>64.9</b>	
frog	76.1	78.4	80.2	<b>85.3</b>	motorbike	41.6	<b>59.9</b>	49.8	
girl	86.5	84.2	86.4	<b>86.8</b>	train	62.1	74.7	<b>77.7</b>	
hummingbird#1	53.2	<b>67.2</b>	53.0	39.0	Mean per object	59.7	70.1	<b>71.7</b>	
hummingbird#2	28.7	68.5	<b>70.5</b>	49.6	Mean per class	61.0	70.9	<b>71.9</b>	
monkey	85.7	87.8	83.1	<b>89.3</b>					
monkeydog#1	40.5	47.1	<b>74.0</b>	25.3					
monkeydog#2	17.1	21.0	<b>39.6</b>	31.7					
parachute	93.7	93.3	<b>95.9</b>	93.7					
penguin#1	81.6	80.4	53.2	<b>93.7</b>					
penguin#2	82.0	83.5	72.9	<b>85.2</b>					
penguin#3	78.5	83.9	74.4	<b>90.1</b>					
penguin#4	76.4	86.2	57.2	<b>90.5</b>					
penguin#5	47.8	<b>82.3</b>	63.5	78.4					
penguin#6	84.3	87.3	65.7	<b>89.3</b>					
soldier	55.3	<b>86.8</b>	76.3	82.0					
worm	65.4	83.2	<b>82.4</b>	80.4					
Mean	58.4	67.5	<b>69.1</b>	67.4					

Table S4: Per-sequence results on the SegTrack-v2 dataset.

Furthermore, incorporating optical flow information and CRF post-processing into MaskTrack substantially increases robustness on all categories, reaching over 70% mIoU on each subcategory. In particular, MaskTrack+Flow+CRF better discriminates cases of low resolution, scale-variation and appearance change.

## D. Dataset specific tuning

As mentioned in §5.3 in the main paper by adding additional ingredients specifically tuned for different datasets, such as optical flow and CRF post-processing, we can push the results even further, reaching 80.3 mIoU on DAVIS, 72.6 on YoutubeObjects and 70.3 on SegTrackv-2. In this section we discuss the dataset specific tuning.

**Optical flow** Although optical flow can provide interesting gains, we found it to be brittle when going across different datasets. Therefore we explored different strategies to handle optical flow.

As discussed in §3.3 of the main paper given a video sequence, we compute the optical flow using EpicFlow [37] with Flow Fields matches [2] and convolutional boundaries [27]. In parallel to the MaskTrack with RGB images, we

Table S5: Per-category results on the YoutubeObjects dataset.

proceed to compute a second output mask using the magnitude of the optical flow field as input image (replicated into a three channel image). We then fuse by averaging the output scores given by the two parallel networks (using RGB image and optical flow magnitude as inputs).

For DAVIS we use the original MaskTrack model (trained with RGB images) as-is, without retraining. However, this strategy fails on YoutubeObjects and SegTrackv-2, mainly due to the failure modes of the optical flow algorithm and its sensitivity to the video data quality. To overcome this limitation we additionally trained the MaskTrack model using optical flow magnitude images on video data instead of RGB images. Training on optical flow magnitude images helps the network to be robust to the optical flow errors during the test time and provides a marginal improvement onYoutubeObjects and SegTrackv-2.

Overall integrating optical flow on top of MaskTrack provides 1~4% on each dataset.

**CRF post-processing** As have been shown in [8] adding on top a well-tuned post-processing CRF [21] can gain a couple of mIoU points. Therefore following [8] we cross-validate the parameters of the fully connected CRF per each dataset based on the available first frame segment annotations of all video sequences. We employ coarse-to-fine search scheme for tuning CRF parameters and fix the number of mean field iterations to 10. We apply the CRF on a temporal window of 3 frames to improve the temporal stability of the results. The color (RGB) and the spatio-temporal (XYT) standard deviation of the *appearance kernel* are set, respectively, to 10 and 5. The pairwise term weight is set to 5. We employ an additional *smoothness kernel* to remove small isolated regions. Both its weight and the spatial (XY) standard deviation are set to 1.

Attribute	Method, mIoU				
	BVS [26]	ObjFlow [44]	MaskTrack	MaskTrack+Flow	MaskTrack+Flow+CRF
Appearance change	0.46	0.54	<b>0.65</b>	0.75	<b>0.76</b>
Background clutter	0.63	0.68	<b>0.77</b>	0.78	<b>0.79</b>
Camera-shake	0.62	<b>0.72</b>	0.71	0.77	<b>0.78</b>
Deformation	0.7	0.77	<b>0.77</b>	0.78	<b>0.8</b>
Dynamic background	0.6	0.67	<b>0.69</b>	0.75	<b>0.76</b>
Edge ambiguity	0.58	0.65	<b>0.68</b>	<b>0.74</b>	<b>0.74</b>
Fast-motion	0.53	0.55	<b>0.66</b>	0.74	<b>0.75</b>
Heterogeneous object	0.63	0.66	<b>0.71</b>	0.77	<b>0.79</b>
Interacting objects	0.63	0.68	<b>0.74</b>	0.75	<b>0.77</b>
Low resolution	0.59	0.58	<b>0.6</b>	0.75	<b>0.77</b>
Motion blur	0.58	0.6	<b>0.66</b>	0.72	<b>0.74</b>
Occlusion	0.68	0.66	<b>0.74</b>	0.75	<b>0.77</b>
Out-of-view	0.43	0.53	<b>0.66</b>	<b>0.71</b>	<b>0.71</b>
Scale variation	0.49	0.56	<b>0.62</b>	0.72	<b>0.73</b>
Shape complexity	0.67	0.69	<b>0.71</b>	0.72	<b>0.75</b>

Table S6: Attribute based evaluation on DAVIS.

## E. Additional qualitative results

In this section we provide additional qualitative results for the MaskTrack<sub>Box</sub> and MaskTrack systems, described in §3 in the main paper. Figure S8 shows the video object segmentation results when considering different types of annotations on DAVIS. Starting from segment annotations or even only from box annotations on the first frame, our model generates high quality segmentations, making the system suitable for diverse applications.

## F. Examples of training mask generation

In §5.2 of the main paper we show that the main factor affecting the quality is using any form of mask deformations when creating the training samples (both for offline and online training). The mask deformation ingredient is crucial for our MaskTrack approach, making the segmentation estimation more robust at test time to the noise in the input mask. Figure S9 complements Figure 2 in the main paper and shows examples of generated masks using affine transformation as well as non-rigid deformations via thin-plate splines (see §4 in the main paper for more details).

## G. Examples of optical flow magnitude images

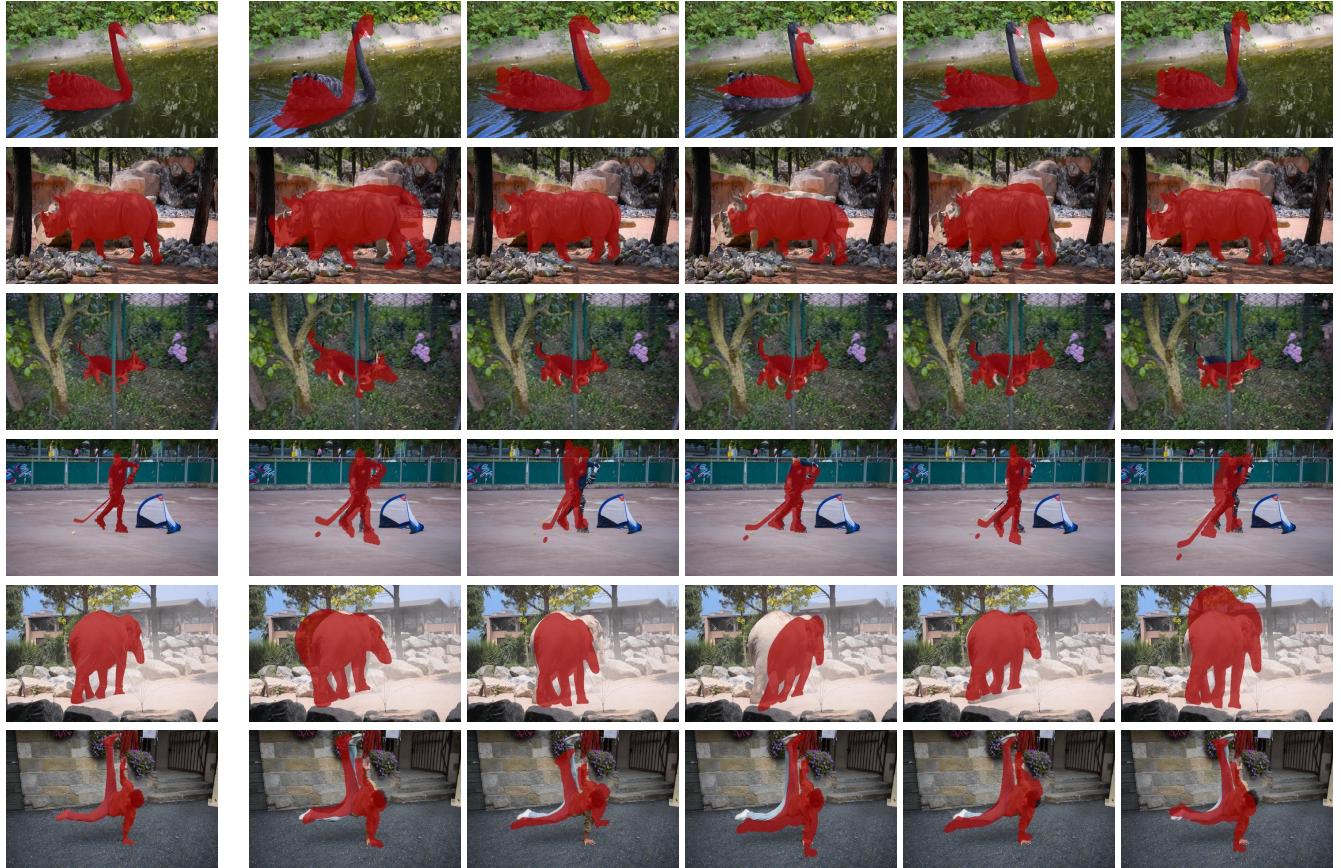
In §3.3 of the paper we propose to employ optical flow magnitude as a source of additional information to guide the segmentation. The flow magnitude roughly looks like a gray-scale object and captures useful object shape information, therefore complementing the MaskTrack model with RGB images as inputs. Examples of optical flow magnitude images are shown in Figure S10. Figure S10 complements Figure 3 in the main paper.



1st frame annotation

Results with  $\text{MaskTrack}_{\text{Box}}$  and  $\text{MaskTrack}$ , the frames are chosen equally distant based on the video sequence length

Figure S8: Qualitative results of  $\text{MaskTrack}_{\text{Box}}$  and  $\text{MaskTrack}$  on Davis using 1st frame annotation supervision (box or segment). By propagating annotation from the 1st frame, either from segment or just bounding box annotations, our system generates results comparable to ground truth.

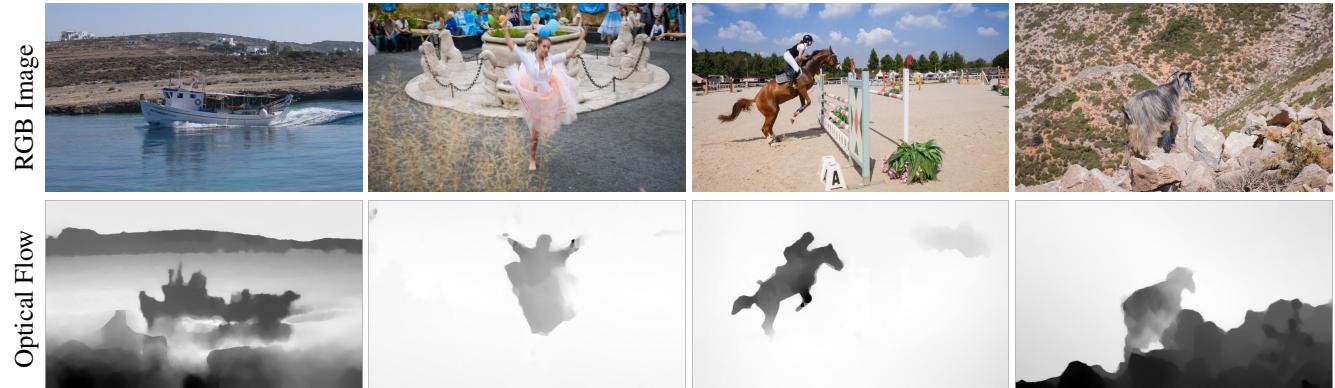


Annotated image

Example training masks

Figure S9: Examples of training mask generation. From one annotated image, multiple training masks are generated. The generated masks mimic plausible object shapes on the preceding frame.

**DAVIS**



**SegTrack-v2**



Figure S10: Examples of optical flow magnitude images for different datasets.