Online Adaptation of Convolutional Neural Networks for the 2017 DAVIS Challenge on Video Object Segmentation

Paul Voigtlaender Bastian Leibe
Visual Computing Institute
RWTH Aachen University
{voigtlaender,leibe}@vision.rwth-aachen.de

Abstract

This paper describes our method used for the 2017 DAVIS Challenge on Video Object Segmentation [26]. The challenge's task is to segment the pixels belonging to multiple objects in a video using the ground truth pixel masks, which are given for the first frame. We build on our recently proposed Online Adaptive Video Object Segmentation (OnAVOS) [28] method which pretrains a convolutional neural network for objectness, fine-tunes it on the first frame, and further updates the network online while processing the video. OnAVOS selects confidently predicted foreground pixels as positive training examples and pixels, which are far away from the last assumed object position as negative examples. While OnAVOS was designed to work with a single object, we extend it to handle multiple objects by combining the predictions of multiple single-object runs. We introduce further extensions including upsampling layers which increase the output resolution. We achieved the fifth place out of 22 submissions to the competition.

1. Introduction

Video object segmentation (VOS) is a fundamental computer vision task with important applications in video editing, robotics, and autonomous driving. The goal of VOS is to segment the pixels of one or more objects in a video using the ground truth pixel masks of the first frame. While single-object and multi-object tracking on bounding box level has received much attention in the computer vision community, their variant on pixel level, *i.e.* VOS, has been less well explored, mainly due to the lack of datasets of sufficient size and quality. However, the recent introduction of the DAVIS 2016 dataset [25] for single-object VOS and the DAVIS 2017 dataset and competition [26] for multi-object VOS together with the adoption of deep learning techniques, led to a significant advancement in the state of the art of VOS. The most successful methods are based on pretrained fully-convolutional neural networks, which are fine-tuned on the first frame of the target video [24, 5, 17, 28]. Most of these methods leave the network parameters fixed after fine-tuning, which means that they cannot deal well with large changes in appearance, *e.g.* arising from an altered viewpoint. This is in contrast to our re-

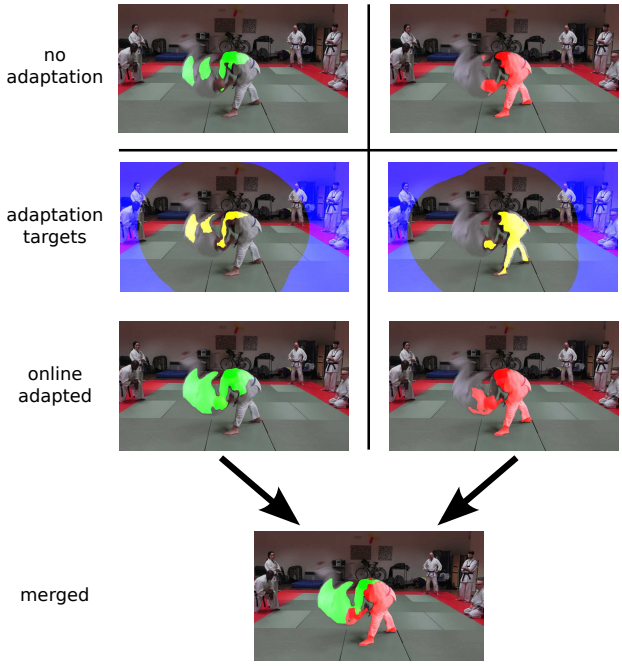


Figure 1: Overview of the proposed multi-object version of *OnAVOS* [28]. For each object in the video, *OnAVOS* is run once using the corresponding pixel mask for the first frame (not shown). *OnAVOS* updates the network online using the adaptation targets to improve the results. Positive adaptation targets are shown in yellow and negative targets are shown in blue. Finally, the single-object predictions are merged to yield the multi-object segmentation. It can be seen that the online adaptation significantly improves the segmentation of the left person.

cently introduced Online Adaptive Video Object Segmentation (*OnAVOS*) [28] method, which adapts to these changes by updating the network online while processing the video, leading to significant improvements on the DAVIS 2016 [25] and the YouTube-Objects [27, 13] datasets for single-object VOS. See Fig. 1 for an overview of the online adaptation approach. In this work, we adopt *OnAVOS* and generalize it in a simple way to work with multiple objects. We demonstrate its effectiveness for this task by achieving the fifth place on the 2017 DAVIS challenge [26].

2. Related Work

Fully Convolutional Neural Networks for Semantic Segmentation. Long *et al*. [20] introduced fully convolutional neural networks (FCNs) for semantic segmentation which replace the fully-connected layers of a pretrained convolutional classification network by 1×1 convolutions, enabling the network to output dense predictions for semantic segmentation instead of just one global class prediction. Recently, Wu *et al*. [29] introduced a very wide fully-convolutional ResNet [12] variant, which achieved outstanding results for classification and semantic segmentation. We adopt their network architecture and pretrained weights for our experiments.

Video Object Segmentation with Convolutional Neural Networks. Caelles *et al*. introduced the one-shot video object segmentation (*OSVOS*) [5] approach, which pretrains a convolutional neural network on ImageNet [7], then fine-tunes it on the 30 training videos of DAVIS 2016, and finally fine-tunes on the first frame of the target video. The resulting fine-tuned network is then applied on each frame of the video individually. The *MaskTrack* method [24] pretrains a convolutional neural network to propagate pixel masks from one frame to the next while exploiting optical flow information. They also fine-tune on the first frame. *LucidTracker* [17] uses a similar approach to *MaskTrack* and introduces an elaborate data augmentation method, which generates a large number of training examples from the first frame. Caelles *et al*. [4] incorporate the semantic information of an instance segmentation method into their VOS pipeline. The current best result on DAVIS 2016 is obtained by *OnAVOS*, which extends the basic pipeline of *OSVOS* by an online adaptation mechanism. *OnAVOS* is described in more detail in Section 3.

Online Adaptation. Online adaptation methods are a common element of many multi-object tracking methods both for classical methods like online boosting [10] or the Tracking-Learning-Detection framework [16], and in the context of deep learning [21]. However, its use on pixel level, *i.e*. for VOS, is less well explored and prior work mainly focuses on classical methods like online updated color or shape models [2, 1, 23] or random forests [8].

3. Online Adaptive Video Object Segmentation

The Online Adaptive Video Object Segmentation (*OnAVOS*) [28] method extends *OSVOS* by an additional objectness pretraining step and online updates. Fig. 2 illustrates its pipeline, which will be described in the following.

Base Network. The first step of *OnAVOS* is to pretrain a convolutional neural network on large datasets like ImageNet [7], Microsoft COCO [19], and PASCAL [9] for classification or semantic segmentation to learn a powerful representation of objects. The resulting pretrained network is called the base network.

Objectness Network. In the next step, the base network is fine-tuned for pixel objectness [15, 14] on the PASCAL dataset [9] with extended augmentations [11]. By treating each of the 20 classes as foreground and all other pixels as background, the network is trained for binary classification and learns a general notion of which pixels belong to objects.

Domain Specific Objectness Network. In order to better match the target domain, *i.e*. videos from DAVIS, the network is then further fine-tuned for objectness on the DAVIS 2017 training sequences using all annotated objects as foreground target and all other pixels as background. This yields the domain specific objectness network.

Test Network. At test time, the domain specific objectness network is fine-tuned on the first frame of the target video in order to adapt to the specific appearance of the object of interest. The resulting network is called the test network and can either directly be applied to the rest of the video as in *OSVOS* [5], or be further updated online as described below.

Online Adapted Test Network. Algorithm 1 shows the online update mechanism of *OnAVOS* for single-object VOS. When processing a new video frame, positive and negative pixels are selected as training examples. For positive examples, we select pixels, for which the network is very confident that they belong to the foreground, *i.e*. pixels, for which the foreground probability provided by the network is above a threshold $\alpha = 0.99$. For negative examples, a different strategy is employed, since using strong background predictions as negative examples destroys all chances to adapt to changes in appearance. Instead, all pixels, which have a distance of more than $d = 190$ pixels from the predicted foreground mask of the last frame are selected as negative examples, and all remaining pixels are assigned a "don't care" label. To deal with noise, an erosion operation can optionally be applied to the mask of the last frame, before calculating the distance. See Fig. 1 for an example of the selected adaptation targets. The obtained pixel labels can then be used to fine-tune the network on the current frame. However, naively doing so quickly leads

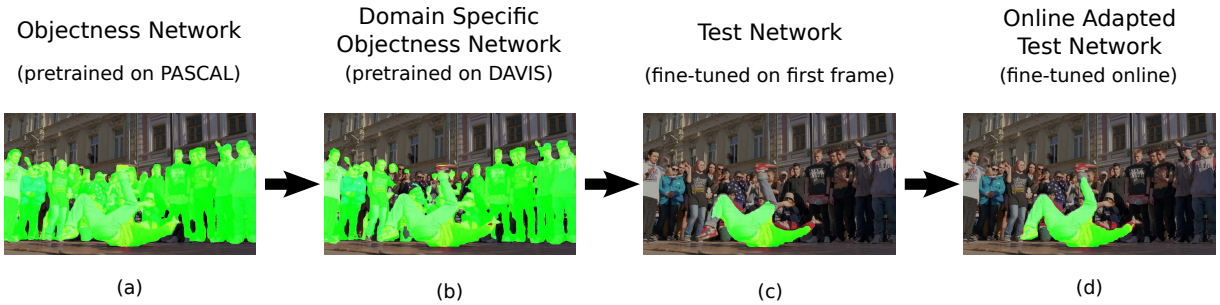


Figure 2: The pipeline of *OnAVOS* [28] for a single object. Starting with pretrained weights, the network learns a general notion of objectness on PASCAL (a). The network is then further pretrained for objectness on the DAVIS training set to better match the target domain (b). At test time, the network is fine-tuned on the first frame of the target video to adapt to the appearance of the object of interest (c). *OnAVOS* adapts the network online while processing the video, which makes it more robust against appearance changes (d).

Algorithm 1 Online Adaptive Video Object Segmentation (*OnAVOS*) for single object VOS

Input: Objectness network $\mathcal{N}$, positive threshold α, distance threshold d, total online steps n_{online}, current frame steps n_{curr}

Fine-tune $\mathcal{N}$ for 500 steps on $frame(1)$
$lastmask \leftarrow ground\_truth(1)$
for $t = 2 \dots T$ **do**
  $lastmask \leftarrow erosion(lastmask)$
  $dtransform \leftarrow distance\_transform(lastmask)$
  $negatives \leftarrow dtransform > d$
  $posteriors \leftarrow forward(\mathcal{N}, frame(t))$
  $positives \leftarrow (posteriors > \alpha) \setminus negatives$
  if $lastmask \neq \emptyset$ **then**
    interleaved:
      Fine-tune $\mathcal{N}$ for n_{curr} steps on $frame(t)$ using $positives$ and $negatives$
      Fine-tune $\mathcal{N}$ for $n_{online} - n_{curr}$ steps on $frame(1)$ using $ground\_truth(1)$
  end if
  $posteriors \leftarrow forward(\mathcal{N}, frame(t))$
  $lastmask \leftarrow (posteriors > 0.5) \setminus negatives$
  Output $lastmask$ for frame t
end for

to drift. In order to prevent this drift, the first frame with known ground-truth is mixed in as additional training example during online updates, the weight of the loss of the training examples from the current frame is reduced by a factor $\beta = 0.05$, and a different learning rate is used. We perform $n_{online} = 15$ update steps on each frame, from which only $n_{curr} = 3$ use the current frame and all other use the first frame.

In order to achieve robustness against occlusions and bad segmentation results, we assume that the tracked object is lost, when the predicted foreground mask is empty after the erosion operation. In this case we do not apply online updates on the current frame, until the foreground mask becomes non-empty again.

Network Architecture. The above described online adaptation method can be applied to any neural network, which provides pixel-level predictions. Here we adopt the architecture used by *OnAVOS*, *i.e*. network A from Wu *et al*. [29] which is a very wide convolutional neural network with 38 hidden layers with residual connections [12] and roughly 124 million parameters. It does not include transposed convolutions or skip connections, but uses dilated convolutions [31] in the upper layers instead of strides or pooling in order to retain a sufficiently high resolution. In all our experiments, we use the bootstrapped cross-entropy loss [30] which only considers a fraction (25% in our experiments) of the hardest pixels, *i.e*. pixels with the highest cross-entropy value. This loss function works well for unbalanced class distributions, which are common for VOS. For optimization we use Adam [18] and a batch size of only one image.

4. Extensions

Multiple Objects. The original formulation of *OnAVOS* is designed for the single-object case. We extend it to the multi-object case by running the single-object *OnAVOS* version one time for each object in the target video and then combining the predictions in a simple way. The fine-tuned network used in the run for the kth object provides for each pixel a probability p_k, that the pixel belongs to the kth object. If at least one of the objects is predicted with a probability of more than 0.5, then we select the object with the highest probability for this pixel. Otherwise, we label the pixel as background. We do not use a CRF for the multi-object case. Our merging strategy performed better than the simple version provided by the DAVIS 2017 development kit, which merges single-object masks according to the arbitrary order, in which they are given to the script. However, we expect that a more elaborate strategy for handling multiple objects will provide further improvements.

Upsampling Layers. Due to strided convolutions, our network provides only one output for each 8×8 pixel block.

We noticed that especially for small objects occurring in DAVIS 2017, this can be a significant limitation. Hence, we added three upsampling layers between the last convolutional layer and the output layer with 200, 128, and 64 feature maps, respectively. The upsampling layers consist of three components and are designed to avoid the checkerboard artifacts, which often arise when using transposed convolutions [22]. The first component consists in enlarging the input feature map by a factor of two in both dimensions using nearest neighbor interpolation. Afterwards, the output of the last layer which has the same resolution is concatenated to the upsampled feature maps. Finally, a 3×3 convolution is performed. The combination of all three upsampling layers restores the original resolution resulting in a prediction of each individual pixel.

Lucid Data Dreaming. We use a simplified variant of the data augmentation mechanism from *LucidTracker* [17] to generate additional training examples based on the first video frame. To this end, we remove the object of interest from the scene and fill the missing pixel using an inpainting technique [6]. Both the background and the object are then independently and randomly transformed using rotation, scaling and thin-plate spline deformations. Finally, the transformed object is inserted at a random position in the transformed background. For simplicity, the insertion was done naively without a blending technique, and we did not exploit that the pixel masks of multiple objects in the scene are known. However, we expect that this could lead to further improvements. We used the images generated this way only for the fine-tuning on the first frame. We found it helpful to only use them for every second update step and use the original first frame otherwise. Additionally, we reduced the weight of the loss for the generated examples by a factor of 10.

Ensembling. Since each run of *OnAVOS* involves a considerable amount of randomness, which mainly arises from data augmentations, we averaged the pixel-wise posterior probabilities over four runs.

5. Experiments

Our experiments are based on the same implementation as *OnAVOS* [28]. We will make our implementation available together with config files and pretrained models at `https://www.vision.rwth-aachen.de/page/OnAVOS`. Most parts of our experimental setup are identical to the setup used by *OnAVOS* [28]. We adopt network A from Wu *et al*. [29], which was pretrained on ImageNet [7], COCO [19] and PASCAL [9, 11], as the base network. We conducted all experiments on the DAVIS 2017 dataset [26] which consists of a training set with 60 sequences, and a validation, a test-dev, and a test-challenge set with 30 sequences each. The performance is measured using the mean intersection-over-union (mIoU) and the $\mathcal{F}$-measure which quantifies the quality of the contours [25]. To minimize the influence of random variations, we report the average over at least two runs as results when no ensembling is used. We tuned the online adaptation hyperparameters on the validation set to better fit to the multi-object scenario. During pretraining and fine-tuning, we used random scaling, flipping and gamma data augmentations. At test time, we averaged the predictions for each frame over 10 randomly sampled augmented versions of the input image.

Subset	validation		test-dev		test-challenge	
Measure	mIoU	$\mathcal{F}$	mIoU	$\mathcal{F}$	mIoU	$\mathcal{F}$
No adaptation	57.0	61.8	46.9	51.7	-	-
OnAVOS	61.0	66.1	50.1	55.4	-	-
+Upsampling	63.0	68.8	50.2	56.0	-	-
+Lucid D. D.	63.1	69.4	52.5	58.8	-	-
+Ensemble	64.5	71.2	53.4	59.6	54.8	60.5

Table 1: Results on the DAVIS 2017 dataset for multi-object VOS. The entry *OnAVOS* refers to our multi-object generalization of *OnAVOS* without further extensions. As can be seen, the online adaptation of *OnAVOS* significantly improves the results and each additional extension is useful. The validation set is easier than the other sets as it contains less objects per sequence.

Table 1 presents the final results with all extensions on three subsets of DAVIS 2017 and shows the effect of each extension on the validation and test-dev sets. The winner of the DAVIS 2017 challenge is selected based on the average of the mIoU and the $\mathcal{F}$-measure on the test-challenge set. We achieve an mIoU of 54.8% and an $\mathcal{F}$-measure of 60.5%, which leads to an average of 57.7% with which we achieved the fifth place in the challenge. The table further shows that the online adaptation scheme of *OnAVOS* is highly effective also in the multi-object case. On the validation set, the mIoU is improved from 57.0% to 61.0% and on the test-dev set from 46.9% to 50.1%. Additionally, it can be seen that each of the proposed extensions (see Section 4) leads to improvements on its own and their combination leads to an increase of over 3% on both subsets.

Difficulties and Failure Cases. The numbers in Table 1 (*e.g*. 54.8% mIoU on the test-challenge set) show that the DAVIS 2017 dataset with multiple objects is significantly more difficult than DAVIS 2016, on which *OnAVOS* achieves an mIoU score of 85.7%. Hence, it is worthwhile to study, what exactly causes the difficulties in DAVIS 2017. Fig. 3 shows qualitative results on four selected sequences. Subfigures (a) and (b) show the results for two sequences, in which the objects of interest are of a different type, *i.e*. a rider and a horse, or a person and a bike. In both sequences, our method is able to produce high-quality results although it has to handle multiple objects. This is in contrast to the se-

quence of subfigure (c), where the objects of interest are all of the same type, *i.e*. three persons, and hence have a similar appearance. While the contours are still relatively accurate, after some time our method confuses the persons; in the last frame the right person is incorrectly labeled as another person. In the context of tracking, this kind of error is well-known and called identity-switch [3]. In the dogs-jump (d) sequence, an identity switch occurs even for different typed objects, *i.e*. a dog is labeled with the identity of a person. Another difficulty in DAVIS 2017 is the presence of very small or thin objects, which are hard to model by a fine-tuned convolutional neural network. An example of this is the left person in the lab-coat sequence (c), who is holding two phones, which are hardly visible due to their small size but still labeled as individual objects in the ground truth.

6. Conclusion

In this work, we extended *OnAVOS* to the multi-object scenario and applied it to the DAVIS 2017 dataset. We showed that also for multiple objects, the online adaptation scheme is highly effective. Furthermore, we introduced upsampling layers, lucid data dreaming augmentations and ensembling as extensions to *OnAVOS* and showed their effectiveness. While our results are promising and we achieved the fifth place in the 2017 DAVIS challenge, we expect that explicit incorporation of temporal context information and handling of object identities will lead to further improvements on top of *OnAVOS* in the future.

Acknowledgements. The work in this paper is funded by the EU project STRANDS (ICT-2011-600623) and the ERC Starting Grant project CV-SUPER (ERC-2012-StG-307432).

References

[1] X. Bai, J. Wang, and G. Sapiro. Dynamic color flow: A motion-adaptive color model for object segmentation in video. In *ECCV*, 2010.

[2] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: Robust video object cutout using localized classifiers. *ACM Trans. Graphics*, 28(3):70:1–70:11, 2009.

[3] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *JVIP*, 2008.

[4] S. Caelles, Y. Chen, J. Pont-Tuset, and L. Van Gool. Semantically-guided video object segmentation. *arXiv preprint arXiv:1704.01926*, 2017.

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

[6] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *Trans. Image Proc.*, 2004.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[8] L. F. Ellis and V. Zografos. Online learning for fast segmentation of moving objects. In *ACCV*, 2012.

[9] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[10] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006.

[11] B. Hariharan, P. Arbelaez, L. D. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014.

[14] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017.

[15] S. D. Jain, B. Xiong, and K. Grauman. Pixel objectness. *arXiv preprint arXiv:1701.05349*, 2017.

[16] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, 34(7):1409–1422, 2012.

[17] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. In *arXiv preprint arXiv: 1703.09554*, 2017.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[21] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[22] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

[23] K. E. Papoutsakis and A. A. Argyros. Integrating tracking with fine object segmentation. *Image and Vision Computing*, 31(10):771–785, 2013.

[24] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017.

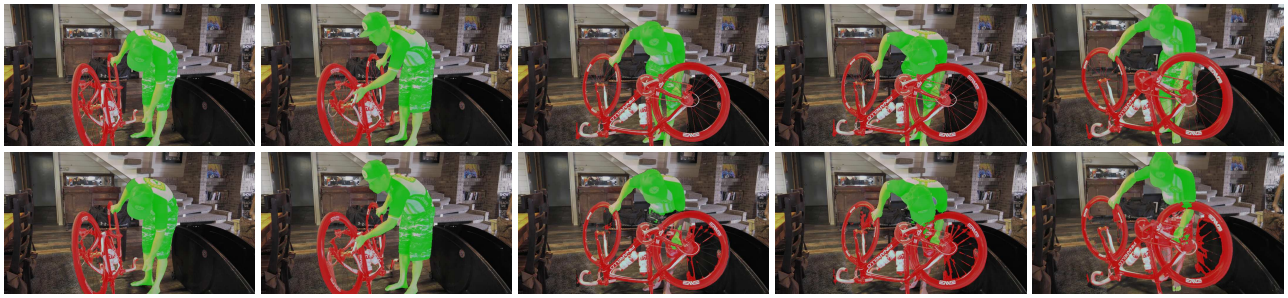
[25] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[26] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[27] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[28] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017.

(a) horsejump-high



(b) bike-packing

(c) lab-coat

(d) dogs-jump

Figure 3: Qualitative results on the DAVIS 2017 validation set. The first row of each subfigure shows the ground truth annotations and the second row shows the result of our complete system with all extensions. The most left images show the first frame, for which the ground truth annotation is given. Our method yields good results for the first two sequences, but on the other two shown sequences some identity switches occur.

[29] Z. Wu, C. Shen, and A. v. d. Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[30] Z. Wu, C. Shen, and A. v. d. Hengel. Bridging category-level and instance-level semantic image segmentation. *arXiv preprint arXiv:1605.06885*, 2016.

[31] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.