Video Instance Segmentation 2019: A winning approach for combined Detection, Segmentation, Classification and Tracking.

Jonathon Luiten*
RWTH Aachen University
luiten@vision.rwth-aachen.de

Philip H.S. Torr
University of Oxford
phst@robots.ox.ac.uk

Bastian Leibe
RWTH Aachen University
leibe@vision.rwth-aachen.de

Abstract

Video Instance Segmentation (VIS) is the task of localizing all objects in a video, segmenting them, tracking them throughout the video and classifying them into a set of predefined classes. In this work, divide VIS into these four parts: detection, segmentation, tracking and classification. We then develop algorithms for performing each of these four sub tasks individually, and combine these into a complete solution for VIS. Our solution is an adaptation of UnOVOST, the current best performing algorithm for Unsupervised Video Object Segmentation, to this VIS task. We benchmark our algorithm on the 2019 YouTube-VIS Challenge, where we obtain first place with an mAP score of 46.7%.

1. Introduction

Video Instance Segmentation (VIS) is the task of localizing, segmenting, classifying and tracking all instances of a set of object classes within a video. This task can be seen as the extension of Instance Segmentation [10] to the video domain. This extension to video is a natural next step for the computer vision community in the search for algorithms that can understand real world scenes through the eyes of a video camera. This VIS task was recently introduced in [28], with the release of the YouTube-VIS dataset (YT-VIS) which is the YouTube-VOS [27] dataset adapted to the VIS task for 40 different object categories.

VIS differs from Semi-Supervised Video Object Segmentation (VOS) in that in VIS no first frames segmentations are given to guide which objects should be tracked, and from Unsupervised Video Object Segmentation (UVOS) in that the objects to be tracked are set by predefined classes rather than whichever objects are salient in the video. Given the similarities of VIS to UVOS, we propose to adapt the current best performing UVOS method, UnOVOST [29], to the VIS task. To do this we divide the VIS task into four components: Detection, Classification, Segmentation and Tracking. We focus on improving detection, classification and segmentation specifically for the VIS task and then using these as input to the UnOVOST algorithm for tracking.

Our final VIS solution is evaluated on the 2019 YT-VIS challenge, where we obtain on mAP score of 46.7% which obtains first place in the challenge, and outperforms the previous VIS state-of-the-art by more than 14.4 percentage points in mAP.

2. Related Work

Video Instance Segmentation. VIS is introduced in [28], where they provide the YT-VIS dataset, an adaption of [27] to this new task. This contains 40 object classes, including people, vehicles, animals and other common objects. As a baseline they adapt Mask R-CNN [5] by adding an association head used to track objects over time. Their unified architecture can be trained end-to-end for detection, classification, segmentation and tracking. This is in contrast to our work where we tackle each of these components separately to achieve maximum performance.

Multi-Object Tracking and Segmentation. Multi-Object Tracking and Segmentation (MOTS) [22] is very similar to VIS in that objects are to be tracked and segmented in video given a set of classes. MOTS differs in that it contains less classes, requires that masks do not overlap and has much longer videos with many more objects that frequently appear and disappear. Because of these differences the evaluation metrics used for these tasks are very different, but the current state-of-the-art approach is very similar, a Mask R-CNN adapted with a association head for tracking [22].

Unsupervised Video Object Segmentation. Another task related to VIS is Multi-Object Unsupervised Video Object Segmentation (UVOS) [1]. In UVOS objects are also required to be tracked and segmented throughout a video, but there is no given set of object classes which need to be segmented. Instead objects need to be tracked if they

*Work performed while at the University of Oxford

are "salient" throughout a whole video sequence. Unlike in VIS these objects don't need to be assigned a category label, but must not have any overlapping pixels between masks. Recently, the first DAVIS challenge on UVOS was held at CVPR'19. The winning method, and current state-of-the-art for the UVOS task is UnOVOST [29]. This method tracks objects in two stages, first building up short tracklets based on optical flow motion consistency, before merging these into long tracks using the visual similarity of tracklets given by a ReID embedding network. In this work we adapt this method to the VIS task by changing the way that detection, classification and segmentation are performed, but keeping the core tracking algorithm on UnOVOST unchanged.

Semi-Supervised Video Object Segmentation Semi-Supervised Video Object Segmentation (VOS) is also related to VIS. In VOS the objects to be tracked and segmented are given as segmentation masks in the first frame. VOS was first introduced in [18] for single objects and extended to multiple objects in [19]. A large scale dataset for VOS was introduces in [27]. Current best performing VOS methods either propagate labels from the first frame [23, 21], or detect and segment potential objects and then link these over time [13]. Our approach follows this second paradigm, but isn't able to use the first frame as guidance for which objects should be tracked, instead tracking all objects belonging to a set of classes. We also adapt the segmentation networks and ReID networks from [13], as these perform very well, winning the 2018 DAVIS Challenge [11] and the 2018 YouTube-VOS challenge [12], and obtaining 2nd in the 2019 DAVIS Challenge [14].

Instance Segmentation and Object Detection. The task of instance segmentation was introduced in [10] and was an extension of the popular task of object detection, from predicting bounding boxes to predicting segmentation masks. VIS can be seen as extending this task further to video. Because of this, the evaluation metrics for VIS are directly taken from Instance Segmentation [10] and only adapted to work across a whole video rather than a single image. VIS can then be approached as performing instance segmentation on each frame, and then linking these segmentations through time.

3. Method

Overview. Our approach is to adapt UnOVOST [29], which won the 2019 DAVIS Challenge on UVOS, to the VIS domain. To do this, we divide the VIS task into four subtasks and find solutions for each separately.

Detection. For detection we adapt a Mask R-CNN [5] detector (similarly to UnOVOST). However the detector needs to be adapted to the YT-VIS benchmark to detect the 40 object classes. We use a Mask R-CNN implementation from TensorPack [24], using a ResNet-101 [6] model with a Feature Pyramid Network [9], group normalisation [25] and cascade [2]. This model is pretrained on COCO [10] from scratch without ImageNet [4] pretraining.

To adapt this network to VIS, we created a training set by combining the YT-VIS [28], COCO [10] and OpenImages [8] datasets. We trained this detector on 39 classes, the 40 classes of YT-VIS with "monkey" and "ape" combined. This is because OpenImages only has a class which is a mix, and because in the YT-VIS training set it is unclear exactly what the difference between these two classes should be (e.g. baboons are labeled as both ape and monkey, some gorillas mislabeled as monkeys). Thus we detect these classes together and rely on our classifier later to distinguish between the two.

For COCO we use the 19 classes which overlap with the YT-VIS classes. The "bird" class was set to ignore regions (as multiple birds such as owl, eagle and duck are in YouTube-VOS). We map the OpenImages classes to YouTube-VOS classes, with all of our 39 classes being mapped to by at least one OpenImages class. We only use images that contain at least one annotation from our 39 classes that is not a person (because of too many people in OpenImages). We set all of the background of OpenImages images to be ignore regions and we don't sample negatives from this dataset (as OpenImages is not densely annotated). We reweigh how often we sample each image during training for class balancing. Classes are sampled such there in one epoch there are at least 5000 examples of each class. This results in sharks being sampled 18 times more often than horses. Also images form the YT-VIS dataset are sampled three times more often than those in COCO and OpenImages.

Classification. The classification branch our Mask R-CNN detector works reasonably well, but still often misclassifies examples. To improve this, we use a ResNeXt-101 32x48d classifier [26] pretrained on 940 million Instagram images [15], before being trained on ImageNet [4]. We then defined a mapping of ImageNet (INet) classes to YT-VIS classes.

This mapping results in 310 of the 1000 INet classes being mapped to our 40 YT-VIS classes, with 123 INet classes being mapped to dog and 20 to truck. Some classes are not represented (person, skateboard, giraffe, hand and surfboard). Some INet classes are mapped to multiple YT-VIS classes, e.g. "Amphibious vehicle" being mapped to both boat and truck. There are 11 INet classes mapped to just monkey, 2 to just ape and 7 to both due to the ambiguity in YT-VIS as to what is a ape and what is a monkey.

The final INet classification score for each YT-VIS class is then the sum of the classification scores for all of the contributing INet classes.

The final classification scores were then a weighted com-

	mAP	AP50	AP75	AR1	AR10
Ours	**46.7**	**69.7**	**50.9**	**46.2**	53.7
foolwood	45.7	67.4	49	43.5	50.7
bellejuillet	45	63.6	50.2	44.7	50.3
linhj	44.9	66.5	48.6	45.3	**53.8**
minmingdii	44.4	68.4	48.7	43.6	50.8
xiAaonice	40	57.8	44.9	39.6	45.2
guwop	40	60.8	43.9	41.2	49.1
exing	39.7	62.1	42.6	41.4	46.1
MaskTrack R-CNN[28]	32.3	53.6	34.2	33.6	37.3

Table 1. Results in the 2019 YouTube-VIS Challenge, compared to top 8 other participants, and the previous state-of-the-art.

bination of the scores from our Mask R-CNN detector and our INet trained classifier.

Segmentation. UnOVOST [29] used segmentations from Mask R-CNN maskrcnn. In [13], is was shown that by using a separate segmentation network on bounding box crops performs much better. We adopt this network from [13], a variant of DeepLabV3+ [3]. We take the pretrained weights from [13] and finetune this on the YT-VIS dataset [28] for the 40 classes.

Tracking. We use UnOVOST [29] to link our given segmentation masks in time to consistent object tracks. UnOVOST works in two stages. It first builds tracklets by linking together segmentations using optical flow. For a mask in frame t, we check the overlap between the mask generated by warping this mask into frame $t+1$ using optical flow, and the masks in frame $t+1$. If this overlap is greater than a threshold then these masks are merged into a tracklet. For optical flow estimation we use PWC-Net [20]. In a second stage, these tracklets are merged into long term object tracks using their visual similarity, as defined by a object reidentification vector extracted from a ReID network [17, 16]. This network is trained on YouTube-VOS [27] using a triplet loss variant [7] in order to generate 128 dimensional ReID vectors which are similar for crops of the same object (in different frames), and different for crops of different objects. For each tracklet, the ReID embedding is extracted for each proposal and averaged over the whole tracklet. The L2 distance between these embeddings is then the measure of the visual dissimilarity between two tracklets. Tracklets are then merged using a dynamic programming inspired algorithm which builds a tree of possible optimal tracks given tracklet's visual similarities. The best tracks are then selected from this tree based on their saliency and their temporal extent (longer tracks are preferred). We refer the reader to [29] for more details.

Putting it all together. In VIS segmentations are allowed to overlap, thus when we are not sure which class a track belongs to we propose the existence of the same track multiple times with different classes and scores.

To obtain a track's score for each class, we average the class scores for the mask in each timestep. Frames with

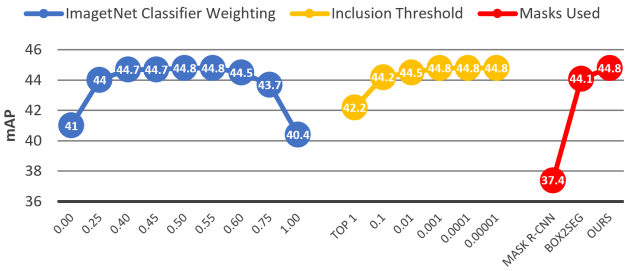

Figure 1. Results of three different ablation studies on the YT-VIS validation set. Blue: Varying the weighting between the detector scores and the ImageNet classifier scores. Yellow: Varying the minimum score threshold required for including a classified track into the results. Red: Varying the segmentations used; Mask R-CNN classification head [5] trained on COCO, Box2Seg [13] trained on COCO and Mapillary, and Ours which is Box2Seg finetuned on YT-VIS.

no masks are given 0 score thus short tracks are down weighted. We do this for both detection scores and INet scores. The final score is the weighted average of these two scores (with equal weighting). We output each track multiple times for every class with a score greater than 0.0001. Note that the detector doesn't discriminate between apes and monkey, so the one detection score is used for both. Also our INet classifier doesn't give scores for 5 of the 40 classes, so for these we only use detector scores.

4. Experiments

The main evaluation of our method is on the YT-VIS test set as part of the 2019 YT-VIS Challenge. We also ablate different design decision using the YT-VIS validation set.

The VIS task is evaluated using the mAP metric. This is similar to the mAP metric used for instance segmentation [10], however it has been extended over the whole video domain. AP is the area under the precision-recall curve, and mAP is the average of AP over multiple IoU (intersection over union) thresholds (50% to 95% in 5% steps) and averaged over all object classes. Other metrics are also presented such as AP50 and AP75 (AP at 50% and 75% IoU threshold respectively), and AR1 and AR10 (maximum recall given 1 or 10 proposals respectively, also averaged over IoU thresholds and classes). In order to adapt AP and AR from images (instance segmentation) to videos, the IoU is simply calculated over the whole video by summing up the intersections for every frame and dividing it by the sum of the unions for each frame. All frames, even frames with no ground truth object present are used for this evaluation.

Table 1 shows our results in the 2019 YT-VIS Challenge, and compares to the top 8 competitors and the previous state-of-the-art from [28]. Our approach wins the challenge with a 1 percentage point gap between our result and that of the second place team. We also improve 14.4 percentage points over MaskTrack R-CNN [28] the current state-

of-the-art baseline.

As well as our challenge winning results, we also present a number of ablation studies detailing different aspects of our architecture design. The blue curve in Figure 1 shows the effect of combining classification scores from both the detector and the INet trained classifier. Individually, both classifiers perform reasonably well, but in combination they perform much better. This is because the type of mistakes each make are very different. The detector struggles to correctly classify classes that were not so common in the training set, such as seals which are commonly classified as apes. The ImageNet classifier on the other hand, as it was not trained on crops but on whole images, will often misclassify a hat with a mouse in the middle as a mouse.

The yellow curve in Figure 1 shows the effect of including more and more proposals in the results which are uploaded. Interesting, due to the way mAP is calculated it seems that it is always better to include the classification result for every class for every track, no matter how low the classification score is. This is because, when classifications are completely incorrect, by including low scoring but correct tracks increases the recall, and including incorrect classifications but with a low score doesn't hurt the precision-recall curve because these are ranked lower than the correct classifications.

The red curve in Figure 1 shows the effect of using different segmentation results for the VIS task. The Box2Seg segmentation network [13] performs much better than the Mask R-CNN [5] segmentation head. Training Box2Seg on the YT-VIS instances improves performance over the COCO/Mapillary training.

5. Conclusion

In this paper we have adopted UnOVOST to the task on Video Instance Segmentation (VIS). In order to do this, we divided the VIS task into four separate components: detection, classification, segmentation and tracking. In order to successfully use UnOVOST for tracking, we needed to train a detector on a wide range of data specifically for the VIS task, adapt a segmentation network to the VIS classes and combine both detection scores and ImageNet trained classifier scores for classification. By adapting UnOVOST in this way, we are able to outperform previous VIS methods, and win the 2019 YouTube-VIS challenge. A future direction for research into VIS is in how to combine all four parts into one unified model that can be trained end-to-end over a whole video, while still maintaining competitive performance for each part.

Acknowledgments: This project has been funded, in parts, by ERC Consolidator Grant DeeViSe (ERC-2017-COG-773161), by a Google Faculty Research Award, by CCAV project Streetwise and by EPSRC/MURI grant EP/N019474/1. We would like to thank Paul Voigtlaender, Idil Esen Zulfikar and Joanna Materzynska for helpful discussions.

References

[1] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv preprint arXiv:1905.00737*, 2019.

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

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

[4] 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.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

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

[7] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[8] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[10] 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.

[11] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for the davis challenge on video object segmentation 2018. In *CVPRW*, 2018.

[12] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: proposal-generation, refinement and merging for the youtube-vos challenge on video object segmentation 2018. In *ECCVW*, 2018.

[13] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018.

[14] J. Luiten, P. Voigtlaender, and B. Leibe. Combining premvos with box-level tracking for the 2019 davis challenge. In *CVPRW*, 2019.

[15] D. K. Mahajan, R. B. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

[16] A. Ošep, P. Voigtlaender, J. Luiten, S. Breuers, and B. Leibe. Large-scale object discovery and detector adaptation from unlabeled video. *arXiv preprint arXiv:1712.08832*, 2017.

[17] A. Osep, P. Voigtlaender, J. Luiten, S. Breuers, and B. Leibe. Large-scale object mining for object discovery from unlabeled video. In *ICRA*, 2019.

[18] 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.

[19] 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.

[20] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[21] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019.

[22] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019.

[23] P. Voigtlaender, J. Luiten, and B. Leibe. Boltvos: Box-level tracking for video object segmentation. *arXiv preprint arXiv:1904.04552*, 2019.

[24] Y. Wu et al. Tensorpack. `https://github.com/tensorpack/`, 2016.

[25] Y. Wu and K. He. Group normalization. In *ECCV*, 2018.

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

[27] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018.

[28] L. Yang, Y. Fan, and N. Xu. Video instance segmentation. *arXiv preprint arXiv:1905.04804*, 2019.

[29] I. E. Zulfikar, J. Luiten, and B. Leibe. Unovost: Unsupervised offline video object segmentation and tracking for the 2019 unsupervised davis challenge. In *CVPRW*, 2019.