CTVIS: Consistent Training for Online Video Instance Segmentation (Supplemental Material)

Kaining Ying[1,2*] Qing Zhong[4*] Weian Mao[4] Zhenhua Wang[3†] Hao Chen[1†]
Lin Yuanbo Wu[5] Yifan Liu[4] Chengxiang Fan[1] Yunzhi Zhuge[4] Chunhua Shen[1]

[1] Zhejiang University [2] College of Computer Science and Technology, Zhejiang University of Technology
[3] College of Information Engineering, Northwest A&F University
[4] The University of Adelaide, Australia [5] Swansea University, UK

https://github.com/KainingYing/CTVIS

In this appendix, we first briefly describe the source code files. (Appendix A). Then we provide the procedure for generating the pseudo-video dataset and the information about the dataset (Appendix B). In addition, we give more results on VIS (Appendix C). Next, we extend CTVIS to video panoptic segmentation (Appendix D). Finally, we discuss the limitation and future work.

A. Code

We include the source code in the zip file. Further implementation details of consistent training can be found in `ct_plugin.py`, and the training and inference procedures are included in `ctvis_model.py`. `get_pseudo_data.py` is used to generate the pseudo-video datasets. Furthermore, all codes, environmental guidelines, model weights and training logs will be made publicly available.

B. Pseudo-Video Datasets

Attributions	YTVIS21*		OVIS*	
	Train	Test	Train	Test
Images / Videos	88,462	421	86,080	140
Instances	434,159	986	405,866	881
Masks	434,159	29,297	405,866	73,027

Table 1. The attributions of YTVIS21* and OVIS*.

As mentioned in Section 4.3, we adopt the overlapping categories of COCO and VIS datasets for training. Specifically, we use COCO to generate pseudo-videos as the training set and select a portion of the target VIS dataset as the validation set. Since the validation set of the VIS dataset used for evaluation is not publicly available, we split the original training set into two non-overlapping parts, respectively used as training and validation sets. In order to facilitate reproducibility, we provide a detailed description of the dataset in this appendix. For YTVIS22*, we select *person*, *motorbike*, *car*, *airplane*, *train*, *truck*, *boat*, *bird*, *cat*, *dog*, *horse*, *cow*, *elephant*, *bear*, *zebra*, *giraffe*, *flying disc*, *snowboard*, *skateboard*, *surfboard* and *tennis racket* as the categories. For OVIS*, we select *person*, *bicycle*, *vehicle*, *motorcycle*, *airplane*, *boat*, *cat*, *dog*, *horse*, *sheep*, *cow*, *elephant*, *bear*, *zebra* and *giraffe* as the categories. Specifically, we select only those images and videos from the COCO and target VIS datasets that contain instances of the categories of interest. We select 421 and 140 videos from the training set of YTVIS21 and OVIS as the validation of YTVIS* and OVIS*, respectively. Table 1 shows the attributes of the YTVIS21* and OVIS*. The generation procedure is included in `get_pseudo_data.py`.

C. Additional Results on VIS

C.1. Datasets information

YTVIS19 [15] covers 40 object classes and contains 2,238/302 videos for training/validation. YTVIS21 [15] retains the number of categories of YTVIS19 while expanding the datasets to 2,985/421 videos for training/validation and improving the instance annotation quality. OVIS [10] has 25 object classes and contains 607/140 videos for training/validation. While the number of videos is less, each OVIS video includes more frames (69.4 frames on average) than YTVIS19 (27.6 frames) and YTVIS21 (30.2 frames). Moreover, OVIS samples typically involve more instances and severe occlusion and thus are more challenging.

*KY (email: kaining.ying.cv@gmail.com) and QZ contributed equally to this work. This work was done when KY, QZ, WM, YZ were visiting Zhejiang University.
†Correponding authors.

C.2. Without COCO Joint Training

We provide results trained solely on video data (for a fair comparison with previous methods [2, 16]). As shown in Table 3, CTVIS (ResNet-50 as the backbone) trained without COCO data achieves 49.3 on AP on YTVIS21, close to 50.1 with COCO joint training. Meanwhile, CTVIS significantly outperforms previous methods under the same w/o CJT setting.

Methods	Throughput	FPS	$AP^{YTVIS19}$	$AP^{YTVIS21}$	AP^{OVIS}
IDOL	3.3	14.3	49.5	43.9	30.2
CTVIS	1.3	13.9	55.1	50.1	35.5

Table 2. Comparison with IDOL in terms of FPS and throughput. We use the ResNet-50 as the backbone and test on RTX-3090.

Methods	CJT	AP
CrossVIS [16]	×	34.2
Mask2Former-VIS [2]	×	40.6
CTVIS	×	49.3
CTVIS	✓	50.1

Table 3. Performance on YTVIS21 without using COCO joint training (CJT). We use the ResNet-50 as the backbone.

C.3. FPS and Throughtout

As shown in Table 2, vanilla IDOL is faster than CTVIS. For training (batch size is 1), the throughput of IDOL is 3.3 samples per second, while the throughput of CTVIS is 1.3. In terms of inference (using ResNet-50 as the backbone, tested on 1 piece of GeForce RTX 3090, and the batch size is 1), CTVIS runs at 13.9 frames per second, which is negligibly slower than IDOL's 14.3. However, CTVIS notably outperforms IDOL by 5% in terms of AP.

C.4. Results on YTVIS22

As shown in Table 4, we evaluate our CTVIS on YTVIS22 [15], which shares the same training set with YTVIS21 and extends with 71 long videos for validation. Because most methods do not report the results on YTVIS22, we use the official source codes and model weights trained on YTVIS21 provided by the authors to measure the average precision in this experiment. We observe that the performance improvement for long videos is significant as the CTVIS with ResNet-50 surpasses the previous SOTA offline method (VITA [5]) and online method (IDOL [14]) by 7.1 and 3.1 on AP^L, respectively. With the stronger backbone Swin-L, CTVIS outperforms the IDOL by 2.4 on AP^L, which suggests that CTVIS generalizes well to long complicated videos.

	Methods	AP	AP^S	AP^L
ResNet-50 [4]	Mask2Former-VIS [2]	35.4	40.7	30.2
	MinVIS [6]	33	44.3	21.6
	VITA [5]	38.9	45.7	32
	IDOL [14]	<u>41.8</u>	<u>47.3</u>	<u>36.3</u>
	CTVIS (Ours)	**44.9**	**50.3**	**39.4**
Swin-L [8]	Mask2Former-VIS [2]	43.4	52.6	34.2
	MinVIS [6]	44.3	55.5	33
	VITA [5]	49.3	57.7	41
	IDOL [14]	<u>52.3</u>	<u>60.7</u>	<u>44</u>
	CTVIS (Ours)	**53.8**	**61.2**	**46.4**

Table 4. Compare CTVIS with SOTA methods [2, 5, 6, 14] on YTVIS22 [15]. AP^S and AP^L denote the performance evaluated on short videos and long videos, respectively. AP is obtained by averaging AP^S and AP^L over classes. The best and second best are highlighted by **bold** and <u>underlined</u> numbers, respectively.

C.5. Instance Embedding Visualization

We visualize the learned instance embeddings by t-SNE [12] in Figure 1. Each plot is for a video and the plot of the same color represents the identical video. The embeddings of the same instance (across different frames) is denoted by one particular color. As there are no annotations in the validation set, we select videos from the training set. For the VIS model [14] without consistent training (first row of Figure 1), instance embeddings are scattered. With consistent training, CTVIS learns more discriminative instance embeddings, which allows a robust tracking of instances in videos.

C.6. More Qualititative VIS Results

Figure 2 gives additional results of CTVIS with Swin-L on the validation and testing sets of OVIS. CTVIS is capable of handling complex scenes that involve object occlusion, deformation, small-scale objects, and is able to obtain accurate segmentation results. We also provide a visualization video, please refer to `demo_dance.mp4` for details.

D. Extend to Video Panoptic Segmentation

Video panoptic segmentation (VPS) [7, 9] requires segmenting and tracking *things* across video and segmenting stuff (*i.e. sky*, *grass*) in each frame. In this appendix, we simply apply our proposed CTVIS to VPS.

D.1. Results on VIPSeg

We select the VIPSeg as the dataset, which is a challenging dataset with 2,806/343 in-the-wild videos for training/validation. It contains 124 semantic classes (58 *thing* and 66 *stuff* classes). Following prior works [1, 9], we use the VPQ, VPQ^{Th}, VPQ^{St} and STQ as the evaluated metrics. We

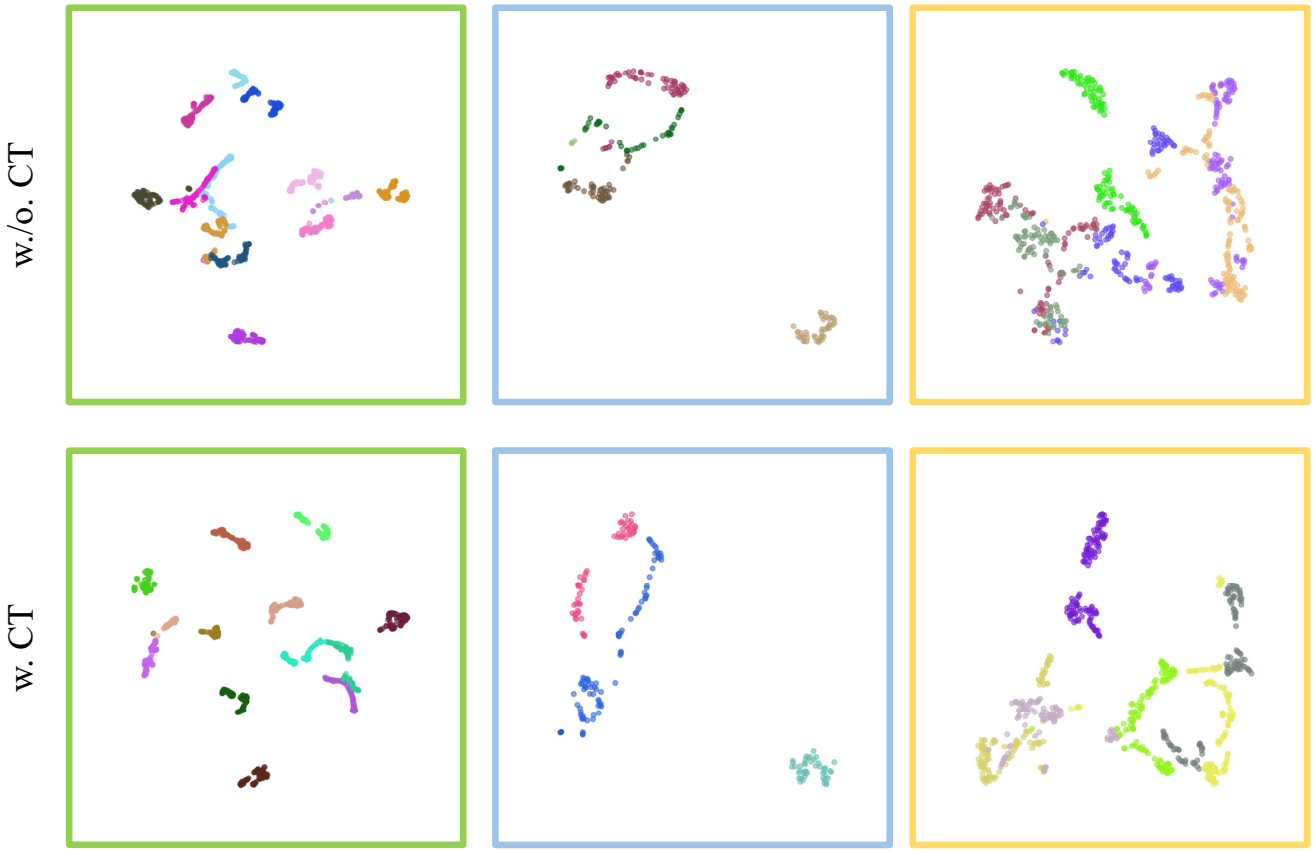

Figure 1. Visualization of instance embeddings without (w./o.) or with (w.) consistent training. Each plot is for a video, and plots of the same color represents an identical video. Within each plot, embeddings of the same instance (in different frames) take one particular color. Clearly, CTVIS enables the learning of more discriminative instance embeddings.

Methods	Backbone	VPQ	VPQ^{Th}	VPQ^{St}	STQ
VPSNet [7]	ResNet-50	14	14	14.2	20.8
VPSNet-SiamTrack [13]	ResNet-50	17.2	17.3	17.3	21.1
VIP-Deeplab [11]	ResNet-50	16	12.3	18.2	22
Clip-PanoFCN [9]	ResNet-50	22.9	25	20.8	31.5
TarVIS * [1]	ResNet-50	33.5	**39.2**	28.5	43.1
CTVIS (Ours)	ResNet-50	**37.5**	36.8	**38.2**	**44.7**
TarVIS * [1]	Swin-L	48	**58.2**	39	52.9
CTVIS (Ours)	Swin-L	**49.5**	48.9	**49.9**	**56.4**

Table 5. Compare CTVIS with SOTA VPS methods on VIPseg. $*$ indicates that TarVIS introduces other video segmentation datasets to train a unified model.

evaluate CTVIS with ResNet-50 and Swin-L as backbones, respectively.

As shown in Table 5, CTVIS is very competitive in comparison with SOTA models [1, 7, 9, 11, 13] proposed for VIPSeg. Please note that TarVIS [1] also takes Mask2Former [3] as the base segmentation model and trains on multiple segmentation datasets [9, 10, 15]. In comparison, CTVIS is simply trained on the VIPSeg dataset, which outperforms TarVIS in all metrics except for VPQ^{Th}. Specifically, with the Swin-L backbone, CTVIS outperforms TarVIS by 3.5 in STQ, which is the most important metric.

D.2. Visualization Results

We visualize the qualitative results of CTVIS with ResNet-50 on the testing set of VIPSeg [9] in Figure 3, where CTVIS offers superior segmentation results even for complex scenes.

E. Limitation and Future Work

CTVIS has its own limitations. For example, it introduces extra computation in order to maintain the memory bank during training. Moreover, as CTVIS is a training strategy, its performance heavily depends on the performance of the base frame-wise segmentor (e.g. Mask2Former). Hence, CTVIS is liable to go wrong if the base segmentor cannot provide decent segmentation results. We suggest introducing some temporal modules in the future to enhance the consistency of the segmentation, as well as incorporating motion constraints to improve tracking.

References

[1] Ali Athar, Alexander Hermans, Jonathon Luiten, Deva Ramanan, and Bastian Leibe. Tarvis: A unified approach for target-based video segmentation. *arXiv preprint arXiv:2301.02657*, 2023. 2, 3

[2] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2Former for Video Instance Segmentation. *arXiv preprint arXiv:2112.10764*, 2021. 2

[3] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-Attention Mask Transformer for Universal Image Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1290–1299, 2022. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 2

[5] Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. VITA: Video Instance Segmentation via Object Token Association. In *Adv. Neural Inform. Process. Syst.*, 2022. 2

[6] Huang, De-An and Yu, Zhiding and Anandkumar, Anima. MinVIS: A Minimal Video Instance Segmentation Framework without Video-based Training. In *Adv. Neural Inform. Process. Syst.*, 2022. 2

[7] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video Panoptic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9859–9868, 2020. 2, 3

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *IEEE Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. 2, 5, 6

[9] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-Scale Video Panoptic Segmentation in the Wild: A Benchmark. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21033–21043, 2022. 2, 3, 4, 6

[10] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded Video Instance Segmentation: A Benchmark. *Int. J. Comput. Vis.*, 130(8):2022–2039, 2022. 1, 3, 5

[11] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3997–4008, 2021. 3

[12] Laurens Van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. 9(11), 2008. 2

[13] Sanghyun Woo, Dahun Kim, Joon-Young Lee, and In So Kweon. Learning To Associate Every Segment for Video Panoptic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2705–2714, 2021. 3

[14] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In Defense of Online Models for Video Instance Segmentation. In *Eur. Conf. Comput. Vis.*, pages 588–605. Springer, 2022. 2

[15] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *IEEE Int. Conf. Comput. Vis.*, pages 5188–5197, 2019. 1, 2, 3

[16] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover Learning for Fast Online Video Instance Segmentation. In *IEEE Int. Conf. Comput. Vis.*, pages 8043–8052, 2021. 2

Figure 2. Additional qualitative results of CTVIS with Swin-L [8] on the validation and testing set of OVIS [10]. Each row represents a video. The same color denotes the same instance. Best viewed in color.

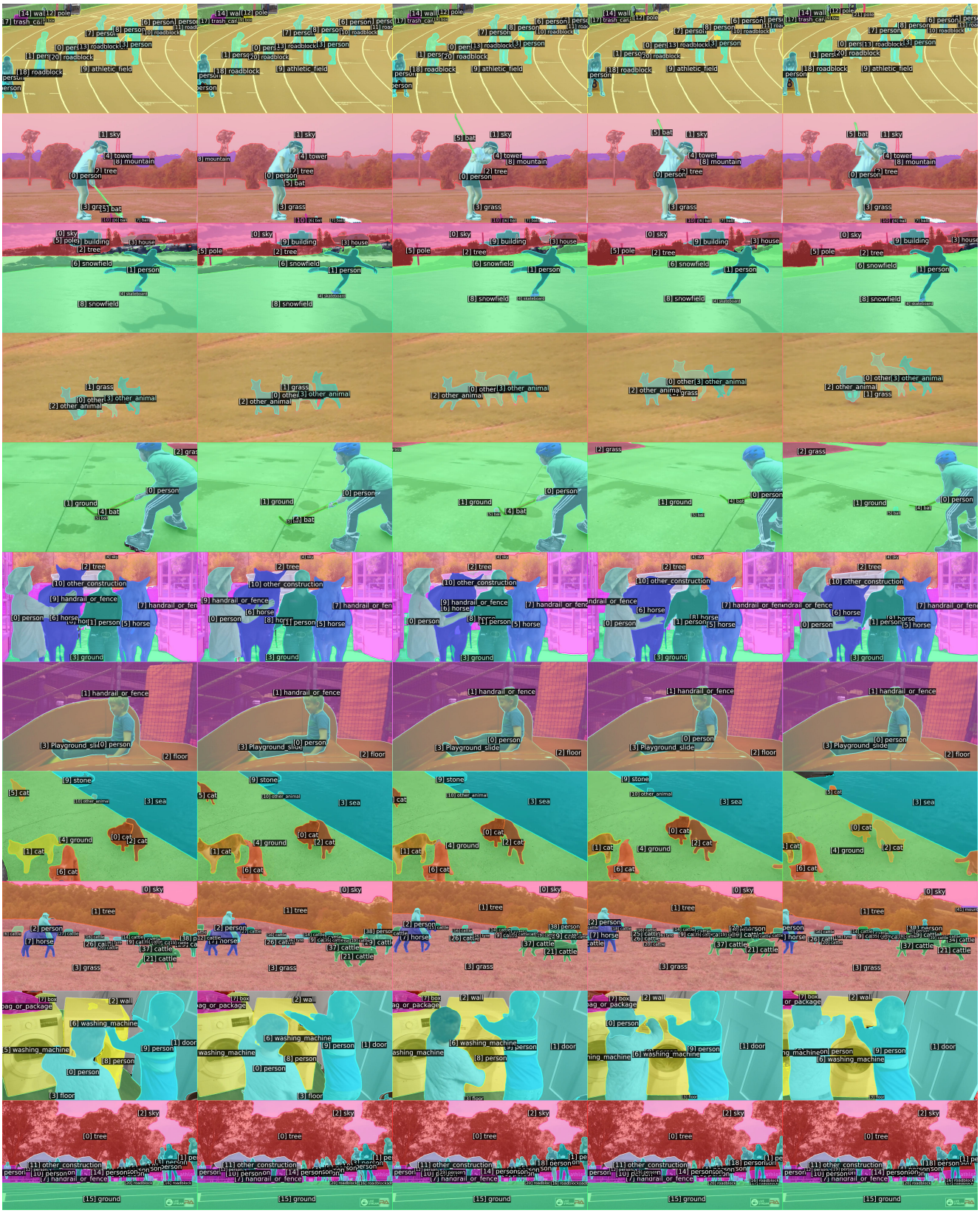

Figure 3. Qualitative results of CTVIS with Swin-L [8] on the testing set of VIPSeg [9]. Each row represents a video. The same color represents the same instance. Best viewed in color.