Supplement to VSPW: A Large-scale Dataset for Video Scene Parsing in the Wild

Jiaxu Miao Yunchao Wei Yu Wu Chen Liang Guangrui Li Yi Yang

A Appendix

A.1 More Details about Label Propagation

Given human-annotated results at 1 f/s, we utilize a label propagation algorithm to help densely annotate videos at 15 f/s. Since the frames within a second usually are very similar to each other, we propose to adopt semi-supervised VOS models to propagate the labels from the annotated key frames to their adjacent unlabelled ones ("S3" of Fig. 2 (a)). Currently, there are two kinds of semi-supervised VOS methods, *i.e.*, fine-tuning based methods [1] and propagation based methods [14]. In this work, we tried both solutions. We follow [1] to fine-tune model on the labeled frames for each video and then inference on the rest unlabeled frames to get machine-labeled masks, as shown in Fig. 2 (a). As the propagation based method, we adopt the state-of-the-art method CFBI [14] and modified it to adapt our setting, as shown in Fig. 2 (b).

Concretely, for the finetuning-based method, we use HRNetV2 [11] pre-trained by ADE20k [17] as the segmentation model. For each video, we firstly finetune the model given key frame annotations and then predict the masks of other frames. During training, the epoch number of finetuning is set to 100, and the batch size is 2. The learning rate is 0.02 with the ploy learning rate policy, where the decay power is 0.9, and the weight decay is 0.0001. We employ the adaptive bootstrapped cross-entropy loss, which takes into account 100% to 15% hardest pixels from the first step to the last step for computing the loss. The multi-scale strategy is adopted by both training and testing stages. For the propagation-based model, we adopt the latest state-of-the-art model, CFBI [14]. Originally, CFBI propagates information of the first frame and the previous frame to the current processing frame. Since there are multiple annotated key frames available at 1 f/s in our setting, therefore, we modified CFBI to bidirectionally propagate masks. We train CFBI using YoutubeVOS [13] and DAVIS [9] jointly. For detail of the model architecture and training setting, please refer to [14].

We compare the two models quantitatively and qualitatively. Qualitatively, compared with the finetuning-based model, we found that the "motion" of the masks generated by the propagation-based model looks smoother. Besides, "spots" are easier to appear in the masks generated by the finetuning-based model. Fig. 1 shows the qualitative comparisons.

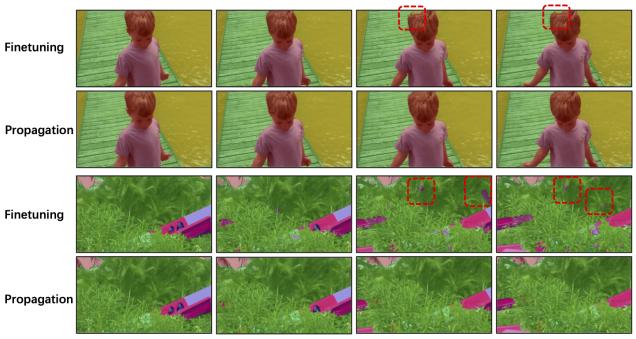

Figure 1. Qualitative comparison between the finetuning-based model and the propagation-based model.

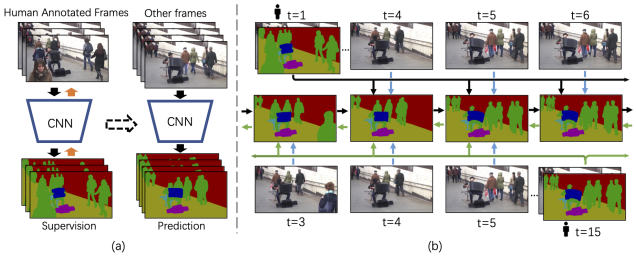

Figure 2. (a) Fine-tuning based model. (b) Propagation based model.

It is not easy to quantitatively compare the two methods since there is no ground-truth. To tackle this, we use the masks generated by the VOS models as input to predict the masks of key frames, reversely. Since key frames are annotated by human labour, these masks can serve as ground truths for the evaluation. We sample 58 videos to quantitatively test the finetuning-based model and the propagation-based model. Table. 1 in the paper shows the comparison and the propagation model [14] significantly outperforms another one [1]. Finally, we choose the bidirectional propagation model to generate masks of the unlabeled video frames.

A.2 More Dataset Statistics

We provide more dataset statistics here considering the space limitation of the paper. Fig. 3 shows the ranked object category frequencies in the frame/video/pixel level, re-

Table 1. Comparison of the finetuning-based model and the propagation based model.

Method	Mean IOU	Weighted IOU	Pixel Acc.	Pixel Acc. per Class
Finetuning [1]	82.76%	93.01%	96.19%	87.97%
Propagation [14]	89.82%	95.87%	97.86%	95.16%

spectively. The object frequencies shows a long-tail distribution. The category appearing with the most frequency is "person". "Tree", "sky", "wall", "grass" and "ground" are backgrounds appearing with high frequencies. Fig. 4 shows the distribution of videos per scene, and top 50% of scenes are shown here. All the videos are selected from 231 scenes. The distribution is relatively uniform, proving that our VSPW covers diverse scenes.

Fig. 5 shows the histogram of pixels for parent classes and their subclasses. There are totally 25 parent classes and 124 subclasses. In each parent class, the distribution of the subclass frequencies is also long-tailed.

A.3 More Details about Evaluation Metrics

There are another two commonly used metrics for semantic segmentation. **Pixel Accuracy** indicates the proportion of correctly classified pixels; **Mean Accuracy** indicates the proportion of correctly classified pixels averaged over all the classes. Results are shown in Table. 3.

Following [7], we calculate the **Temporal Consistency** (TC) by using the mIoU of the predicted mask at the frame t and the warped mask from previous frame $t-1$ by the optical flow,

$$TC(\mathbf{Q}_{t-1}, \mathbf{Q}_t) = \frac{\mathbf{Q}_t \cap \hat{\mathbf{Q}}_{t-1}}{\mathbf{Q}_t \cup \hat{\mathbf{Q}}_{t-1}}, \tag{1}$$

where $\mathbf{Q}_t$ represents the predicted segmentation map of frame I_t and $\hat{\mathbf{Q}}_{t-1}$ represents the warped segmentation map from frame I_{t-1} to frame I_t. We compute the warp mIoU for each video and average the warp mIoU on the videos in the validation/test set. Thus the final Temporal Consistency (TC) score is:

$$TC = \frac{1}{N}\sum_{n=1}^{N}\frac{\mathcal{Q}_n \cap \hat{\mathcal{Q}}_n}{\mathcal{Q}_n \cup \hat{\mathcal{Q}}_n}, \tag{2}$$

where $\mathcal{Q} = \{\mathbf{Q}_2, ..., \mathbf{Q}_T\}$ and $\hat{\mathcal{Q}} = \{\hat{\mathbf{Q}}_1, ..., \hat{\mathbf{Q}}_{T-1}\}$. N denotes the video number. Considering the evaluation time, we test 100 videos in validation and test set for the TC score. The TC score measures the temporal stability by considering two adjacent frames, but ignores the long-range video consistency. Long-range video consistency means the predictions of one object does not change across adjacent C frames, where $C \geq 2$.

Temporal stability has also been studied in VOS tasks [8]. In [8], the temporal stability (TS) is calculated by transforming masks into polygons and matching the SCD (Shape Context Descriptor) distances, which is extremely time-consuming. Thus we randomly select 20 videos from the validation set to calculate TS, and results are shown in Table. 2. TCB achieves better TS than image-based methods while similar to Netwarp and ETC.

Score	TCB_{ocr}	OCR	TCB_{psp}	PSP	Netwarp	ETC
TS ↓	0.299	0.351	0.303	0.344	0.296	0.301

Table 2. Comparison on temporal stability (VOS-metrics).

A.4 Implementation Details

We use ResNet-101 [5] as our backbone and initialize the backbone by the ImageNet [3] pre-trained model. Other modules are initialized from scratch. During training, the input image is augmented by random flipping, random scaling in the range of [0.8,2.0] and random cropping to 479×479. We employ SGD with momentum 0.9 to optimize our model. The clip number of the support frames is 3, and the batch size is set to 8, which means in each step the input contains 8 video clips. The dilation numbers $d1, d2, d3$ of the support frames are $3, 6, 9$, respectively. We set the initial learning rate as 0.002, weight decay as 0.0001 and the total epoch number as 120. We perform the polynomial learning rate policy with factor $(1-(\frac{iter}{iter_{max}})^{0.9})$. The weight on the final loss is set as 1, and the weight on the auxiliary loss is set as 0.4. The standard BatchNorm [6] layer is replaced by the Synchronize BatchNorm [10] to collect the mean and standard-deviation of BatchNorm across multiple GPUs during training. For fair comparisons, all the comparable methods (PSPNet [16], UperNet [12], Deeplabv3+ [2] , OCRNet [15],NetWarp [4],ETC [7]) use the same training settings. For ETC [7], we use the ResNet-101 as the backbone without distillation, because we do not compare the efficiency in this paper.

References

[1] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *IEEE CVPR*, pages 221–230, 2017. 1, 2

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 2, 5

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[4] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video cnns through representation warping. In *ICCV*, pages 4453–4462, 2017. 2, 5

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 2

[7] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *ECCV*, 2020. 2, 5

[8] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE CVPR*, pages 724–732, 2016. 2

[9] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 1

[10] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, pages 5639–5647, 2018. 2

[11] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE CVPR*, pages 5693–5703, 2019. 1

[12] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 2, 5

[13] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *ECCV*, 2018. 1

[14] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, pages 332–348. Springer, 2020. 1, 2

[15] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, August 2020. 2, 5

[16] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE CVPR*, 2017. 2, 5

[17] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *IEEE CVPR*, pages 633–641, 2017. 1

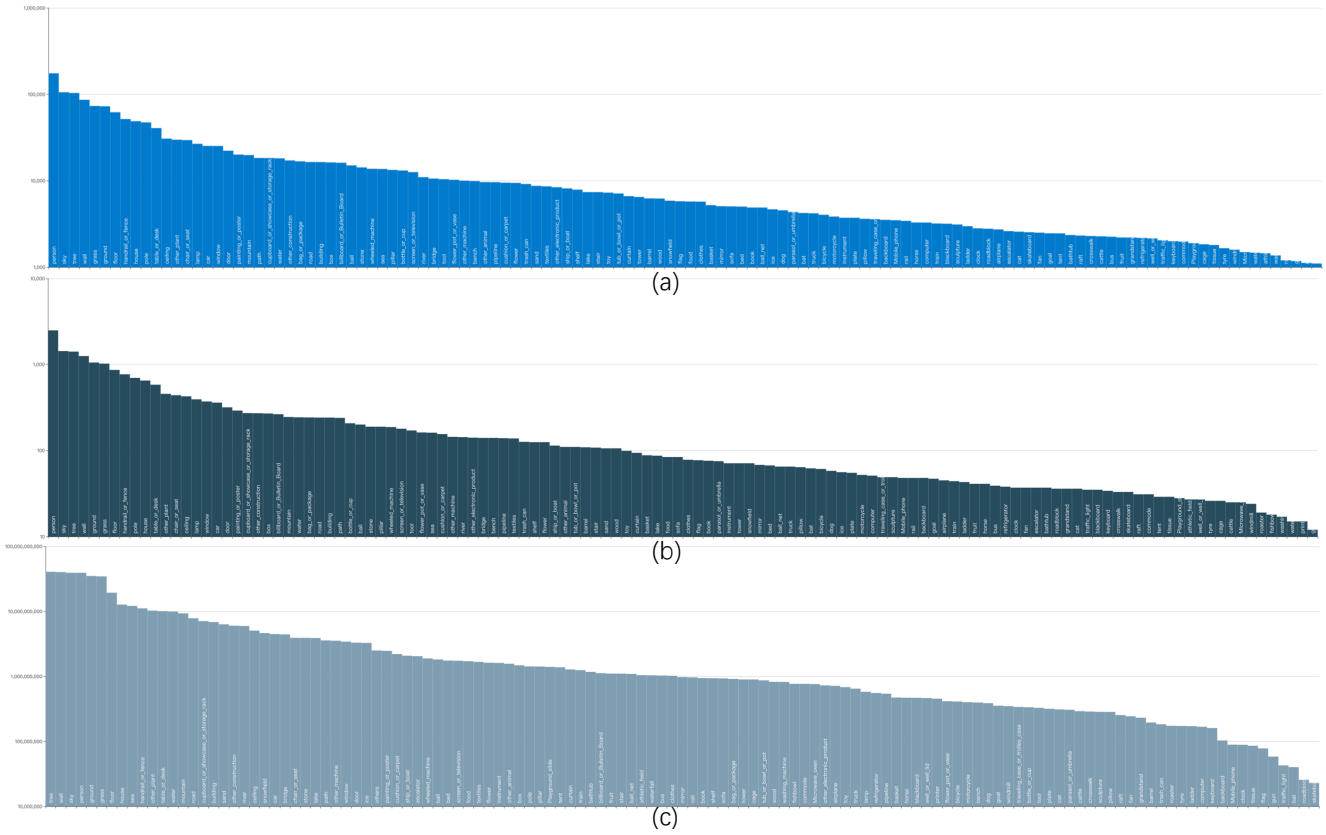

Figure 3. The ranked category frequencies in the (a) frame (b) video (c) pixel level.

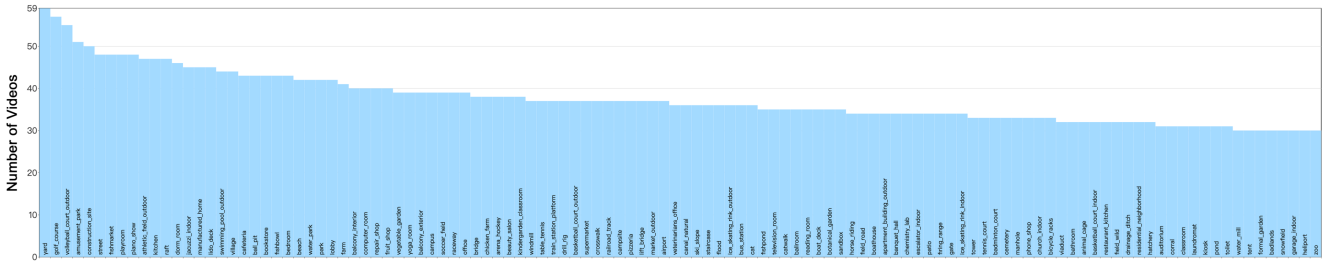

Figure 4. The distribution of videos per scene.

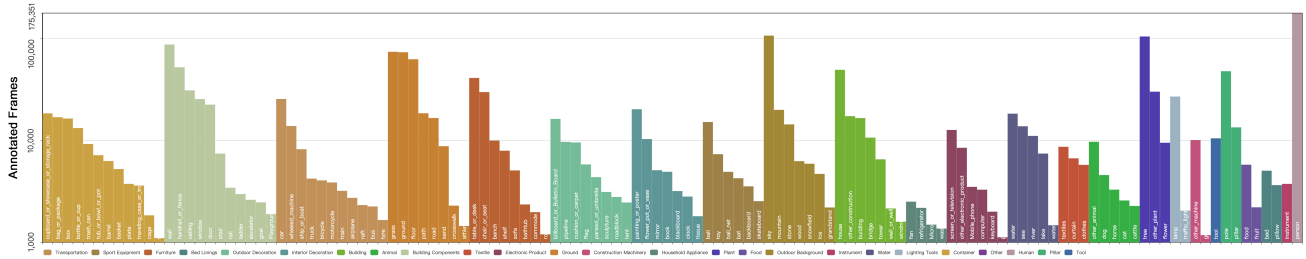

Figure 5. The histogram of pixels for parent classes and their subclasses.

(a) Results on the validation set.

Method	Backbone	mIOU	Weighted IOU	Pixel Acc.	Pixel Acc. per Class	TC	VC_8	VC_{16}
DeepLabv3+ [2]	ResNet-101	34.67%	58.81%	72.82%	45.48%	65.45%	83.24%	78.24%
UperNet [12]	ResNet-101	36.46%	58.60%	72.64%	47.35%	63.10%	82.55%	76.08%
PSPNet [16]	ResNet-101	36.47%	58.08%	72.34%	46.33%	65.89%	84.16%	79.63%
OCRNet [15]	ResNet-101	36.68%	59.24%	73.14%	47.12%	66.21%	83.97%	79.04%
ETC [7]	PSPNet [16]	36.55%	58.29%	72.41%	46.58%	67.94%	84.10%	79.22%
NetWarp [4]	PSPNet [16]	36.95%	57.93%	72.14%	47.09%	67.85%	84.36%	79.42%
ETC [7]	OCRNet [15]	37.46%	59.13%	72.99%	47.94%	68.99%	84.10%	79.10%
NetWarp [4]	OCRNet [15]	37.52%	58.94%	72.93%	47.72%	68.89%	84.00%	78.97%
TCB st-ppm	ResNet-101	37.46%	58.57%	72.50%	47.59%	70.30%	86.95%	82.12%
TCB st-ocr	ResNet-101	37.40%	59.26%	73.22%	48.55%	72.20%	86.88%	82.04%
TCB st-ocr memory	ResNet-101	37.82%	59.49%	73.01%	48.62%	73.63%	87.86%	83.99%

(b) Results on the test set.

Method	Backbone	mIOU	Weighted IOU	Pixel Acc.	Pixel Acc. per Class	TC	VC_8	VC_{16}
DeepLabv3+ [2]	ResNet-101	32.15%	57.08%	70.86%	42.76%	70.01%	80.98%	75.02%
UperNet [12]	ResNet-101	33.46%	54.84%	70.26%	44.77%	66.32%	79.33%	73.29%
PSPNet [16]	ResNet-101	33.78%	56.38%	72.34%	46.23%	70.29%	83.35%	78.29%
OCRNet [15]	ResNet-101	34.02%	56.78%	70.91%	44.97%	69.55%	82.94%	77.42%
ETC [7]	PSPNet [16]	33.84%	56.51%	70.80%	44.28%	69.43%	82.81%	77.06%
NetWarp [4]	PSPNet [16]	33.68%	56.61%	70.82%	44.41%	69.10%	82.55%	77.09%
ETC [7]	OCRNet [15]	34.55%	57.27%	71.25%	45.67%	69.25%	83.12%	78.00%
NetWarp [4]	OCRNet [15]	35.00%	57.67%	71.63%	45.94%	70.23%	83.15%	77.21%
TCB st-ppm	ResNet-101	34.61%	57.25%	71.31%	45.85%	72.02%	85.19%	80.23%
TCB st-ocr	ResNet-101	35.12%	58.11%	72.17%	46.53%	73.86%	85.11%	80.12%
TCB st-ocr memory	ResNet-101	35.62%	58.19%	72.21%	46.88%	74.33%	86.21%	81.90%

Table 3. Comparison on the validation set and the test set. VC_C means we use a clip number C.