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

Jiaxu Miao[1,2] Yunchao Wei[3] Yu Wu[2,3] Chen Liang[1] Guangrui Li[3] Yi Yang[1†]
[1]Zhejiang University [2] Baidu Research [3] ReLER, University of Technology Sydney
jiaxumiao.m@gmail.com, yunchao.wei@uts.edu.au, yi.yang@uts.edu.au

Abstract

In this paper, we present a new dataset with the target of advancing the scene parsing task from images to videos. Our dataset aims to perform Video Scene Parsing in the Wild (VSPW), which covers a wide range of real-world scenarios and categories. To be specific, our VSPW is featured from the following aspects: 1) Well-trimmed long-temporal clips. *Each video contains a complete shot, lasting around 5 seconds on average. 2)* Dense annotation. *The pixel-level annotations are provided at a high frame rate of 15 f/s. 3)* High resolution. *Over 96% of the captured videos are with high spatial resolutions from 720P to 4K. We totally annotate 3,536 videos, including 251,633 frames from 124 categories. To the best of our knowledge, our VSPW is the first attempt to tackle the challenging video scene parsing task in the wild by considering diverse scenarios. Based on VSPW, we design a generic Temporal Context Blending (TCB) network, which can effectively harness long-range contextual information from the past frames to help segment the current one. Extensive experiments show that our TCB network improves both the segmentation performance and temporal stability comparing with image-/video-based state-of-the-art methods. We hope that the scale, diversity, long-temporal, and high frame rate of our VSPW can significantly advance the research of video scene parsing and beyond. The dataset is available at* https://www.vspwdataset.com/.

1. Introduction

Scene parsing aims to assign a unique semantic label to every pixel in a given image, which is a fundamental research topic in the computer vision community and has many potential applications such as image editing, autonomous driving and robotics. With the development of the Convolutional Neural Networks (CNNs), many kinds of fully convolutional neural networks [46, 75, 10, 26] have been proposed to advance this research area. In addition, several image-based datasets, *e.g.*, Pascal-Context [18], ADE20K [78], COCO-Stuff [5], Cityscapes [16], have also been collected to evaluate the effectiveness of these scene parsing approaches.

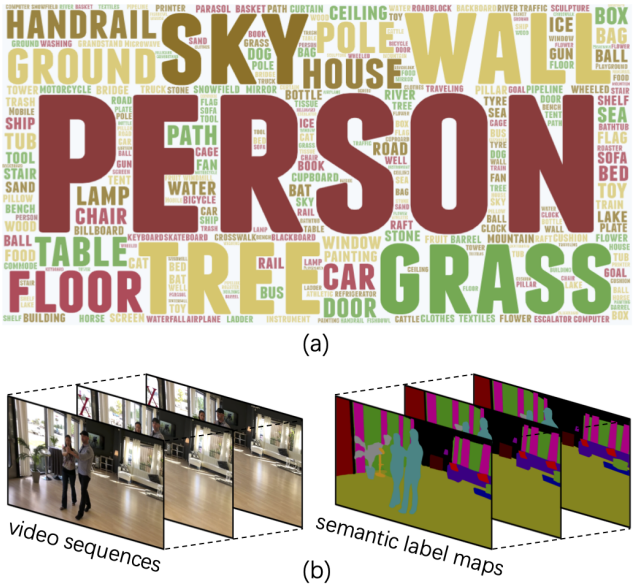

Figure 1. (a) The category cloud of our VSPW. (b) One example for the video scene annotation.

However, the real-world is actually video-based rather than a static state, and learning to perform video scene parsing is more reasonable and practical for realistic applications. Although remarkable progress has been made in image-based scene parsing, few works have been proposed to consider the video scene parsing, which is mainly limited by the lack of suitable benchmarks. Although CamVID [2] has been proposed to tackle the video scene parsing, this dataset is heavily limited by its small scale (701 frames from 6 videos), low frame rate (1 f/s), and single scenario (only the street view is considered). Cityscapes [16] and NYUv2 [58] are often used for the video scene parsing task. However, they are actually image-based datasets because only one frame or several nonadjacent frames in a video clip are annotated.

To advance the scene parsing task from images to videos,

†Corresponding author.
‡Part of this work was done when Jiaxu Miao was an intern at Baidu Research.

we present a new dataset in this work, aiming at performing the challenging yet practical Video Scene Parsing in the Wild (VSPW). The dataset covers a wide range of real-world scenarios (*e.g.*, art galleries, lecture rooms, beach, and street views) and categories from both things (*e.g.*, person, car, desk) and stuff (*e.g.*, road, wall, sky). To the best of our knowledge, our VSPW is the first attempt to tackle the challenging video scene parsing task by considering diverse scenarios. Concretely, our VSPW has the following characteristics:

- *Well-trimmed long-temporal clips.* Based on the predefined real-world scenarios, we collected the related videos from the Internet. Each video is carefully examined and trimmed into a complete shot, lasting around 5 seconds on average.

- *Dense annotation.* Different from previous similar works [2], we provide the pixel-level annotations at a competitive frame rate of 15 f/s, making the temporal information be well considered to learn better video scene parsing models.

- *High resolution.* We abandon those poor videos with low resolution or heavy shake, and only keep high-quality ones. Within our VSPW, over 96% of videos are with high spatial resolutions from 720P to 4K.

Overall, our VSPW totally provides 3,536 annotated videos, including 251,633 frames from 124 categories. Labeling such a large-scale dataset for video scene parsing is very challenging, *e.g.*, time-consuming, expensive, and hard to keep category consistency across the whole video, which may be the main reason that prevents the video scene parsing from being well studied till now. To efficiently and accurately facilitate the annotation process, we develop a human-computer collaboration scheme, which can not only significantly reduce the human effort but also guarantee high-quality annotation masks.

Based on our VSPW, we further propose a simple end-to-end Temporal Context Blending (TCB) network. Our TCB enables the network to harness long-range contextual information from previous frames to help segment the current one, which effectively alleviates those false predictions caused by motion blur, view and scale variations, etc. Extensive experiments on cutting-edge image-based [75, 9, 72, 65] and video-based [20, 45] segmentation methods are evaluated as strong baselines. Compared with these baselines, our TCB network shows its advantages in terms of both segmentation performance and temporal stability. We hope our VSPW can significantly motivate more researchers to develop efficient & accurate algorithms and help ease the future research of video scene parsing.

2. Related Work

2.1. Image Segmentation Datasets

Till now, existing image segmentation datasets can be roughly divided into two subsets, *i.e.*, semantic object segmentation and semantic scene parsing. The former one aims to segment objects of interest and popular benchmarks mainly include Pascal VOC [18], MS COCO [42], and OpenImages [33]. The later one targets on recognizing the semantics of all pixels in the given image and popular benchmarks mainly include COCO Stuff [5], ADE20K [78], LVIS [22], Pascal-Context [49], SUN database [64], Mapillary Vistas [50], Cityscapes [16], and CamVid [2]. To the best of our knowledge, Cityscapes [16], NYUv2 [58] and CamVid [2] can also be considered as video-based scene parsing datasets. However, 1) Cityscapes [16] is actually image-based, which provides the annotation of only one frame in one video sequence; 2) NYUv2 [58] is also image-based. It provides annotations of 1,449 nonadjacent frames from 435,103 frames. 3) CamVid [2] only annotates 6 videos with a low frame rate, *i.e.*, 1 f/s. Besides, Cityscapes and CamVid only focus on one scenario, *i.e.*, the street view, while NYUv2 only focuses on indoor scenarios. Differently, our VSPW contains 3,536 videos from 231 scenarios, which is the first truly meaningful dataset for video scene parsing in the wild.

2.2. Video Object Segmentation Datasets

Video object segmentation (VOS) aims to segment a particular object instance in a video sequence given only the object mask on the first frame, which is class-agnostic. Previous datasets [29, 19, 35, 3, 52, 21] are with some limitations from either small scales or simple contents. Recently, two large-scale ones are proposed, *i.e.*, DAVIS [54] and Youtube-VOS [67], which significantly boost the development of VOS [4, 61, 62, 47, 13, 48]. Most recently, Youtube-VOS is further extended to perform video instance segmentation task [68], where the semantics of each instance is given yet the stuff categories from the background are not considered. Both our VSPW and VOS datasets aim to learn robust spatial-temporal features. However, our VSPW is more generic since all the semantic and instance information is not available in advance during inference.

2.3. Image Segmentation Models

Starting from the fully convolutional networks [46], many subsequent FCN-based models have greatly advanced the image segmentation. Based on the specific tasks, these models are mainly employed to conduct scene parsing [65, 75, 60, 41, 74, 76, 53, 31, 69, 26, 14, 73, 37, 63, 9, 10, 11], instance segmentation [23, 39, 8, 12, 7, 1] and panoptic segmentation [32, 36, 59, 66, 38, 70, 15]. Our VSPW is most related to the scene parsing task, and the current advanced

approaches mainly include Deeplab series [9, 10, 11], non-local series [63, 80, 73, 26], PSPNet [75], HRNet [60], OCRNet [72], etc.

2.4. Video Segmentation Models

Video semantic segmentation requires dense labeling for all pixels in each frame of a video sequence. Recently, several video semantic segmentation approaches have been proposed and previous work can be summarized into two streams. One stream aims to improve the accuracy by exploiting the temporal relations [20, 51, 43, 44, 30]. For instance, NetWarp [20] employs optical flow to warp the feature of the previous frame to the current frame. Another stream focuses on reducing the latency and improving the efficiency [45, 25, 57, 28, 6, 79, 40]. For instance, ETC [45] uses a temporal loss to improve the temporal consistency and the knowledge distillation to reduce the computing cost. Limited to the previous datasets [16, 2], previous methods utilize the semi-supervised setting, which employ the adjacent frames without annotations for the video segmentation. These methods may fail when meeting the large-scale video dataset with large diverse categories.

3. VSPW Dataset

In this section, we first introduce the video collection process in § 3.1. Second, the details of the annotation process are given in § 3.2. Finally, we provide the statistics of our VSPW in § 3.3.

3.1. Video Collection

We aim to cover diverse scenarios in our VSPW dataset. We selected 231 popular scenes, including both indoor and outdoor scenes, of which most are from Places 365 [77]. Based on these scenes, we totally collected 3,536 videos from Youtube. When collecting videos, we prefer videos with moderate object motions or camera motions.

Each video is further cut into a complete shot lasting from 3 to 10 seconds. We clearly reviewed these videos and defined a hierarchic category set for all the things/stuff shown in the video guided by [78], resulting in 25 parent categories and 124 detailed object categories. Please refer to the Appendix for more details.

3.2. Video Annotation

Labeling a video scene parsing dataset is much more difficult than an image dataset due to the following factors. First, it is hard to make the categories consistent across different videos; for instance, one object labeled as "road" in one video is easily annotated as "ground" in another video if the camera view is changed. Second, it is hard to make the categories consistent across adjacent frames in one video, and the "motion" of segmentation masks in a video should look smooth. Third, it is time-consuming to carefully annotate dense video frames. For instance, annotating one 10s video with 15 fps will result in 150 independent frames, and annotating one frame for every pixel costs about 10-15 minutes. Therefore, densely annotating each frame in a video is unaffordable in terms of both finance and human efforts.

Dataset	#Videos	#Images	#Scenes	#Object classes	#FPS	#c/f
Cityscapes [16]	-	2,500	1	30	-	12.2
NYUv2 [58]	-	1,449	464	26	-	-
CamVid [2]	6	701	1	32	1	10.8
VSPW	3,536	251,633	231	124	15	8.55

Table 1. Comparison of video scene parsing datasets.

To efficiently and accurately collect a large-scale video scene parsing dataset, we adopt a human-computer collaboration strategy to tackle the above-mentioned challenges. First, several experts are required to carefully review all videos and identify all categories within each video (§ 3.2.1). Then, for each video, key frames with 1 f/s speed are selected for artificial segmentation annotation. After that, we employ a modified semi-supervised video object segmentation (VOS) approach to propagate semantic labels from the annotated key frames to those unlabelled intermediate frames (§ 3.2.2). Finally, we refine the propagated frames and run the VOS model repeatedly until all pixel-wise annotations are satisfied (§ 3.2.3). More details are provided below.

3.2.1 Video Category Annotation

To make the categories consistent across different videos, the video-level categories are annotated by three expert workers ("S1" of Fig. 2 (a)). For each video, one expert worker first looks through the entire video and record all categories appearing in this video. Then, the annotated video-level categories are sent to the other two expert workers to conduct further examinations. When annotating one frame of a given video, only its video-level categories are prepared for selection. In this way, common workers can not only save a large amount of time in choosing one specific category from hundreds of candidates, but also heavily relieve the problem of category inconsistent across videos. As shown in Fig. 2 (b), only those video-level categories identified by expert workers are available for common workers to choose.

3.2.2 Label Propagation

Densely annotating a video (*e.g.*, 15f/s) is often time-consuming and wastes human labor. To tackle this problem as well as reduce the finance cost, we adopt a human-computer collaboration labeling mechanism. As shown in Fig. 2 (a), we first require the common workers to annotate the frames with a rate of 1 f/s (S2). Then, we adopt the state-of-the-art video object segmentation method, *i.e.*,

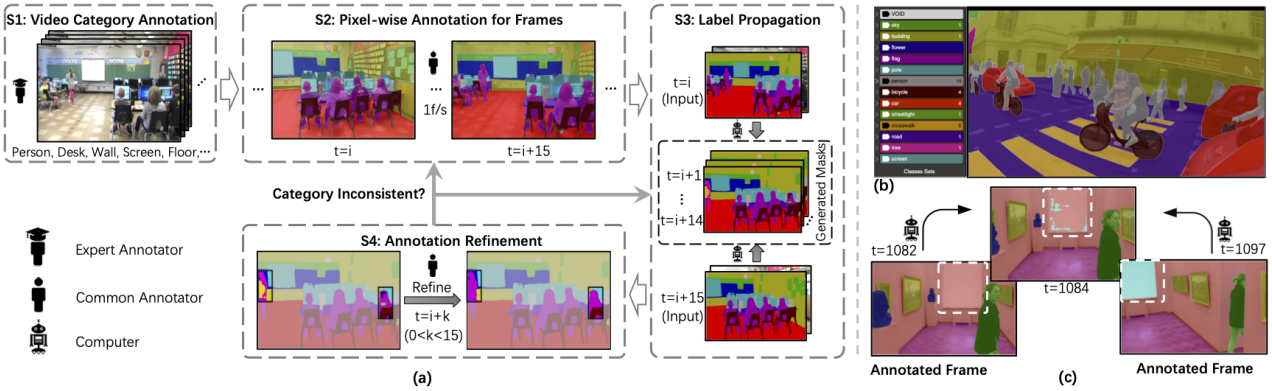

Figure 2. **(a)** The video annotation pipeline. **(b)** Interface of the segmentation editor. **(c)** Semi-supervised VOS model can help to check the consistency of categories. If a generated mask has spots, the adjacent human annotated frames may contain inconsistent categories.

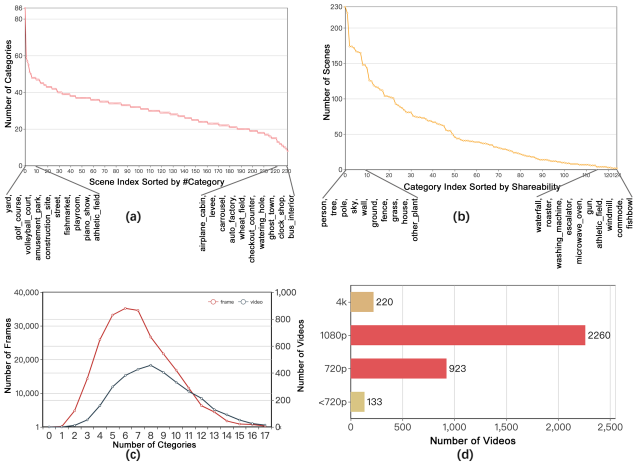

Figure 3. **(a)** The distribution of categories per scene. **(b)** The distribution of scenes per category. **(c)** The number of categories per frame/video. **(d)** The distribution of the resolution for VSPW.

CFBI [71], to propagate the labels from the annotated key frames to their adjacent unlabelled ones and generate masks at 15 f/s. Originally, CFBI is designed to propagate object mask from the first frame to subsequent unlabeled frames. Since there are multiple annotated key frames available at 1 f/s in our setting, we modified the CFBI to bidirectionally propagate masks from two adjacent labeled frames to those unlabeled ones between them (S3). Please refer to the Appendix for more details.

3.2.3 Dense Annotation Refinement

After generating the masks of the unlabelled video frames by CFBI, common workers are further asked to check the quality and refine the annotations artificially ("S4" of Fig. 2 (a)). Particularly, annotation refinement includes two parts: (1) if "spots" appears, the two labeled masks of key frames should be checked to find out if there is inconsistent category annotation, (2) if there are mistakes or defects in the generated masks, common workers are asked to correct them. One example is given in Fig. 2 (c). When the same object is annotated by inconsistent categories (*i.e.*, "windows" and "wall"), the generated masks for unlabeled frames will often be confused due to the bidirectional propagation, resulting in unexpected spots around this object (t=1,084). After the annotation refinement, the refined masks are used as inputs again to generate better masks by CFBI. The model propagation and artificial refinement are operated repeatedly until the results are satisfied.

3.3. Dataset Statistics

Our VSPW contains $3,536$ videos, including $251,633$ frames from 124 categories. Each video contains a well-trimmed long-temporal shot, lasting around 5 seconds on average. The pixel-level masks are provided at a high frame rate of 15 f/s. Fig. 3 (a) and Fig. 3 (b) show how categories are shared across different scenes and how scenes are shared across different categories. Some common stuff categories (*e.g.*, "tree","sky","ground") and one thing category "person" share most of the scenes. Fig. 3 (c) shows the distribution of categories in each frame/video. Most of the frames/videos contain 6/8 categories. Fig. 3 (d) shows the distribution on video resolutions. Over 96% of the captured videos are with high resolution from 720P to 4K. 26% of videos in VSPW are with 720P, and 64% of videos are with 1080P. More statistics can be found in the Appendix.

3.4. Comparison with Other Datasets

The comparisons between our dataset and existing related datasets are shown in Table 1. We mainly compare our dataset with three video-based scene parsing datasets, Cityscapes [16], NYUv2 [58] and CamVid [2]. Cityscapes and CamVid only focus on a single scene (street views). NYUv2 only focuses on indoor scenes. We may notice that the number of object classes per frame (c/f) in VSPW is less than Cityscapes [16] and CamVid [2]. This is because the scene of Cityscapes [16] and CamVid [2] is only the street view which inclines to contain more categories. Compared with these two street view datasets, our VSPW has much

more diverse scenes (231). In addition, our VSPW is also featured with more videos (3,536), more annotated frames (251,633), and a higher frame rate (15 f/s). All these characteristics make our VSPW be the first dataset for tackling the video scene parsing task in the wild.

4. Method

In this section, we introduce a generic Temporal Context Blending (TCB) network, which can effectively leverage long-range contextual information from previous frames to enhance the segmentation of the current one.

4.1. Temporal Context Blending Network

Video scene parsing aims to assign a semantic label to every pixel in given sequential frames. Recent progress on image-based scene parsing tasks [75, 11, 72, 63, 26] has proved that context aggregation modules can significantly help improve the segmentation performance. However, for the video scene parsing task, only considering the contexts within one frame will lose valuable contextual information in the temporal dimension. Therefore, in this work, our mission is to design an effective video-based context aggregation module, which can help acquire valuable contextual information from both spatial-level and temporal-level.

Motivated by image-based methods (OCRNet [72], PSPNet [75]) that harness the contextual information in object regions or multi-scale spatial regions to augment the extracted features, we propose a generic framework to aggregate the contexts in the temporal dimension, namely *Temporal Context Blending Network* (TCB). Fig. 4 (a) shows the pipeline of our TCB framework. For the frame I_t at time t in one video, we use a clip of support frames with a length of N to boost the segmentation performance and temporal stability. The dilation numbers (distance away from the current frame I_t) of support frames are set as $t-d1, ..., t-dN$ (which means the clip of support frames are $\{I_{t-d1}, ..., I_{t-dN}\}$). It is flexible to set the dilation numbers; thus, we can use long-range temporal frames.

As shown in Fig. 4 (a), given the current frame I_t and a clip of support frames $\{I_{t-d1}, ..., I_{t-dN}\}$, we first use a ResNet [24] model pre-trained on ImageNet [17] as the backbone to extract their features, $\mathbf{F}_t$ and $\{\mathbf{F}_{t-d1}, ..., \mathbf{F}_{t-dN}\}$, respectively. Then a temporal context blending module is employed to learn the spatial-temporal contextual representations, $\mathbf{F}_c$, which contain the contextual information from both spatial and temporal dimensions. We concatenate the contextual representations $\mathbf{F}_c$ and the features of the current frame $\mathbf{F}_t$ to generate the augmented feature $\mathbf{F}_a$. It is followed by a convolutional segmentation head to generate the final predictions. Besides, following previous practices [75, 72], we also apply an auxiliary loss to intermediate feature representations from the backbone (as indicated by the green arrows in Fig. 4) to assist the learning process. In this paper, we propose two types of temporal context blending modules, utilizing the contextual information in object regions (Spatial-Temporal OCR) and multi-scale regions (Spatial-Temporal PPM), respectively.

4.1.1 Spatial-Temporal OCR

OCR [72] uses the weighted feature in the object region as the object representations, and computes the relation between each pixel and each object region for the augmentation of the pixel representations. Motivated by OCR, we propose to aggregate the object representations in both spatial and temporal dimensions to acquire richer object information. Concretely, as shown in Fig. 4 (b), give a training clip $\mathcal{S} = \{I_t, I_{t-d1}, ..., I_{t-dN}\}$, we first partition the frame I into K soft object regions $\{\mathbf{M}^1, ..., \mathbf{M}^K\}$, where K is the total number of categories. Each soft object region $\mathbf{M}^k (1 \leq k \leq K)$ refers to the probability of pixels belonging to the class k. During training, we learn the soft object regions using the intermediate representation outputs from the backbone under the supervision from the ground-truth, as the green arrows shown in Fig. 4 (b).

The spatial-temporal object region representation $\mathbf{f}_k$ for the class k is the aggregation of the pixel representations in $\mathbf{F}$ weighted by their confidence score belonging to the kth object region, which is averaged across the temporal frames,

$$\mathbf{f}_k = \frac{1}{N+1} \sum_{\hat{t}=1}^{N+1} \sum_{i \in I_{\hat{t}}} \mathbf{M}_{\hat{t},i}^k \mathbf{x}_i^{\hat{t}}, \tag{1}$$

where $\hat{t} \in \{t, t-d1, ..., t-dN\}$, $\mathbf{x}_i^{\hat{t}}$ is the representation of pixel $p_i^{\hat{t}}$ in $\mathbf{F}_{\hat{t}}$ for the frame $I_{\hat{t}}$.

After obtaining the spatial-temporal object region representations $\{\mathbf{f}_1, ..., \mathbf{f}_K\}$, we compute the relation between each pixel in the target frame I_t and each object region,

$$w_{ik} = \frac{\exp \kappa(\mathbf{x}_i, \mathbf{f}_k)}{\sum_{j=k}^{K} \exp \kappa(\mathbf{x}_i, \mathbf{f}_j)}, \tag{2}$$

where $\kappa(\mathbf{x}, \mathbf{f}) := \phi(\mathbf{x})^T \psi(\mathbf{f})$, $\phi(\mathbf{x})$ and $\psi(\mathbf{f})$ are non-linear functions implemented by 1×1 $conv \rightarrow BN \rightarrow ReLU$.

Finally we use the weights w_{ik} to compute the object contextual representation for pixel p_i in the target frame I_t,

$$y_i = \rho(\sum_{k=1}^{K} w_{ik} \delta(\mathbf{f}_k)), \tag{3}$$

where $\rho(\cdot)$ and $\delta(\cdot)$ denote transform functions implemented by 1×1 $conv \rightarrow BN \rightarrow ReLU$. Then we have the spatial-temporal contextual representations $\mathbf{F}_c$ where each pixel representation in $\mathbf{F}_c$ is y_i. The spatial-temporal contextual representations $\mathbf{F}_c$ is concatenated with the original feature $\mathbf{F}_t$ of I_t to augment the pixel representations.

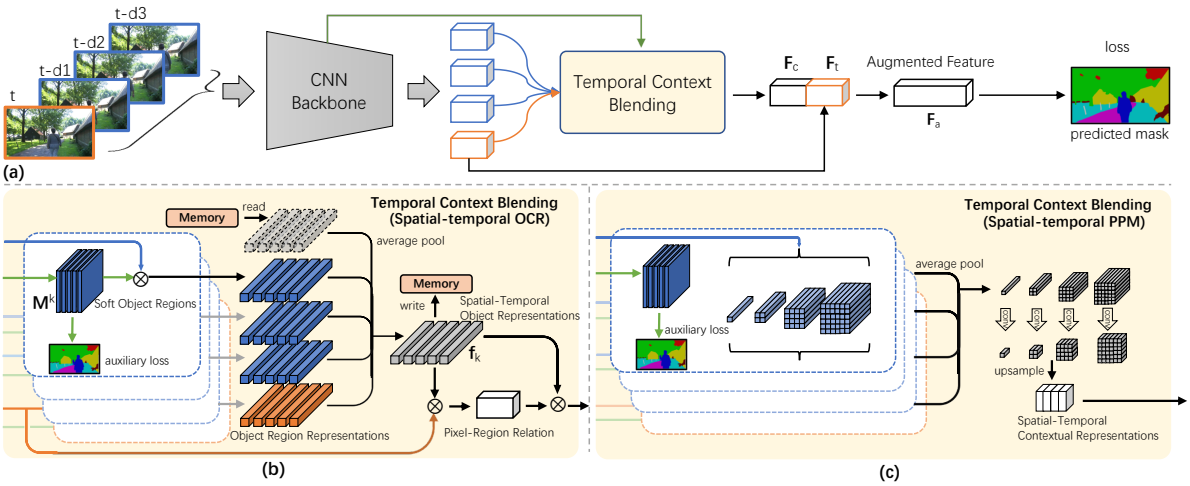

Figure 4. **(a)** An overview of our TCB network. The blue lines denote the support frames and the orange lines denote the target frame. **(b)** Temporal context blending with spatial-temporal object-contextual representations. **(c)** Temporal context blending with the spatial-temporal pyramid pooling.

Limited to the GPU memory, the number of support frames can not be large during the inference stage. To tackle this issue, we propose to employ a Memory module to record the spatial-temporal object representations $\{\mathbf{f}_1, ..., \mathbf{f}_K\}$ of historical frames. Specifically, for each inference step, the spatial-temporal object representations are written into memory by weighted averaging the previous spatial-temporal object representations and the object representations for each frame $I_{\hat{t}}$, as shown in Fig. 4 (b). In this way, the memory module can gradually aggregate the object representations for long-range temporal frames or even the entire video, which leverage abundant contextual information in the temporal dimension.

4.1.2 Spatial-Temporal PPM

Motivated by Pyramid Pooling Module (PPM) [75], which computes multi-scale spatial contextual information, we propose to aggregate the spatial contextual information in the temporal dimension, as shown in Fig. 4 (c). For each frame I, we extract the feature $\mathbf{F}$. Using the pyramid pooling module with different pooling kernels, the feature $\mathbf{F}$ is average-pooled into features with four different scales $(1 \times 1, 2 \times 2, 3 \times 3, 6 \times 6)$. To aggregate the temporal contexts, we average-pool the four features in the temporal dimension, respectively. Then we use transform functions $(1 \times 1\ conv \rightarrow BN \rightarrow ReLU)$ to reduce the channel dimension and upsample the four low-dimension feature maps to get the same size feature as the original feature map $\mathbf{F}$ via bilinear interpolation. Finally, we can obtain the spatial-temporal contextual representations $\mathbf{F}_c$ by concatenating the different levels of features. The spatial-temporal contextual representations $\mathbf{F}_c$ is concatenated with the original feature $\mathbf{F}_t$ of I_t to augment the pixel representations.

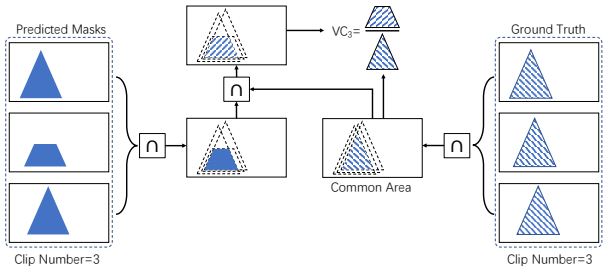

Figure 5. The pipeline for computing the video consistency. GT common area is the pixels whose GT labels are not changed in this clip. This metric is to urge predictions on these unchanged pixels to be consistent in a clip.

5. Experiment

5.1. Dataset Splits

The train set, validation set and test set of VSPW contain $2,806/343/387$ videos with $198,244/24,502/28,887$ frames, respectively. Considering the limitation of the computation source, we resize all the frames in VSPW into 480P (the size of the short side is resized to 480) for training and testing.

5.2. Evaluation Metrics

There are two commonly used metrics for semantic segmentation [46]: **Mean IoU** indicates the intersection-over-union between the predicted and ground truth pixels, averaged over all the classes. **Weighted IoU** indicates the IoU weighted by the total pixel ratio of each class. Different from image segmentation, video scene parsing needs category consistency and temporal stability across frames in one video, to make the motion of masks look "smooth". Previous video segmentation methods [45, 34] measure the temporal stability of a video based on the flow warping error between two adjacent frames. Following [45], 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. Please refer to Appendix for more details about the TC score.

Even though the TC score can measure the temporal stability by considering two adjacent frames, it ignores the video consistency from the long-range aspect. Here, the long-range video consistency means the predictions of one object does not change across adjacent C frames, where $C \geq 2$. To this end, we propose a new metric called **Video Consistency** (VC) to evaluate the category consistency among long-range adjacent frames. Concretely, for a clip with C frames, there are common areas where the category of the pixel does not change, *i.e.*, the intersection of ground-truth in the clip. We compute the intersection of predictions in the common area over the common area to evaluate the video consistency. Formally, for a clip of frames in one video $\{I_{t+i}\}_{i=1}^{C}$, the ground-truth masks are $\{\mathbf{Q}_{t+i}\}_{i=1}^{C}$ and the predicted masks are $\{\mathbf{Q}'_{t+i}\}_{i=1}^{C}$. Thus the video consistency of one clip over C frames is defined as

$$\mathrm{VC}_C = \frac{(\mathbf{Q}'_{t+i} \cap ... \cap \mathbf{Q}'_{t+C}) \cap (\mathbf{Q}_{t+i} \cap ... \cap \mathbf{Q}_{t+C})}{(\mathbf{Q}_{t+i} \cap ... \cap \mathbf{Q}_{t+C})}. \tag{4}$$

We apply a clip-based sliding window to scan the entire video with a step of 1, and acquire the mean VC_C (mVC_C) over all clips. Fig. 5 shows how to compute the video consistency, and takes a clip with $C = 3$ as an example.

5.3. Segmentation Results and Analysis

We conduct experiments on our VSPW using our TCB method, image-based semantic segmentation methods (PSPNet [75], UperNet [65], Deeplabv3+ [11] and OCRNet [72]) and video-based state-of-the-art methods (NetWarp [20], ETC [45]).

5.3.1 Quantitative and Qualitative Comparisons

Table. 2 shows the segmentation performance and video stability scores. The first/second/third group in the table denotes the image-based/video-based/our methods. **TCB** $_{\text{st-ppm}}$ and **TCB** $_{\text{st-ocr}}$ denote our TCB method with the spatial-temporal PPM and OCR modules, respectively. **TCB** $_{\text{st-ocr mem}}$ denotes **TCB** $_{\text{st-ocr}}$ with the memory mechanism during the inference stage.

It can be observed that our method achieves the state-of-the-art on both segmentation performance (mIoU) and video stability (TC and mVC). For the image-based segmentation models, OCRNet [72] achieves the best performance on mIoU, which indicates that the spatial object-contextual information benefits the scene parsing on our VSPW dataset, and performs better than multi-scale context aggregation methods [75, 65]. Comparing with the best competing image-based method, OCRNet [72], TCB $_{\text{st-ocr mem}}$ surpasses it by +1.14% mIoU on the validation set and +1.60% mIoU on the test set, indicating that leveraging the temporal object-contextual information improves the segmentation performance. For the video stability, TCB $_{\text{st-ocr mem}}$ outperforms OCRNet [72] by +7.42% (+4.78%) TC score on the validation (test) set, and +3.89% (+3.26%) mVC_8 score on the validation (test) set, respectively, indicating that our method obtains significantly better stability across both adjacent frames and long-range frames. Comparing with the video-based methods (ETC [45] and NetWarp [20]), TCB $_{\text{st-ocr mem}}$ slightly outperforms them on mIoU performance but surpasses them by a large margin on the stability (TC and mVC). Compared with OCR-based NetWarp [20], our model outperforms it by +4.74% TC score and +3.86% mVC_8 score on the validation set.

For TCB $_{\text{st-ppm}}$, comparing with its corresponding image-based method (PSPNet [75]), TCB $_{\text{st-ppm}}$ surpasses it by +0.99% mIoU, +4.41% TC score and +2.79% mVC_8 score on the validation set. Above results indicate that harnessing the long-range spatial-temporal contextual information can effectively improves the segmentation performance and video stability.

Compared with image-based methods, previous video-based state-of-the-arts (ETC [45] and NetWarp [20]) improves TC but achieves similar mVC score on the validation set. One reason is that ETC [45] and NetWarp [20] utilize optical flow between two adjacent frames, and ignore the long-range temporal information.

Segmentation examples from the validation set are shown in Fig. 6. Compared with the competing methods, the results from TCB are more detailed and accurate. For instance, our TCB predicted the correct "dog" object while other methods failed.

5.3.2 Ablation Study

Selection of the Support Frames. Table. 3 shows the impact of the number and the dilation of support frames on the validation set. The results demonstrate that adding the support frames can consistently improve mIoU and the temporal stability (mVC and TC) compared with the baselines (OCRNet [72] or PSPNet [75]). In addition, when there is only one support frame, the video stability (TC and mVC) perform worse than three support frames. When the support frames cover a long-range (d1, d2, d3 are 3, 6, 9), the segmentation performance and mVC score are better than a shot-range (1, 2, 3). For the TC score, which considers the consistency between two adjacent frames, a short-range support clip (1, 2, 3) performs better.

Impact of the Memory Aggregation. As demonstrated in Section. 4.1.1, the spatial-temporal object representations is recorded and aggregated in a memory. This op-

Method	Backbone	Validation Set					Test Set				
		mIOU	Weighted IOU	TC	mVC_8	mVC_{16}	mIOU	Weighted IOU	TC	mVC_8	mVC_{16}
DeepLabv3+ [11]	ResNet-101	34.67%	58.81%	65.45%	83.24%	78.24%	32.15%	57.08%	70.01%	80.98%	75.02%
UperNet [65]	ResNet-101	36.46%	58.60%	63.10%	82.55%	76.08%	33.46%	54.84%	66.32%	79.33%	73.29%
PSPNet [75]	ResNet-101	36.47%	58.08%	65.89%	84.16%	79.63%	33.78%	56.38%	70.29%	83.35%	78.29%
OCRNet [72]	ResNet-101	36.68%	59.24%	66.21%	83.97%	79.04%	34.02%	56.78%	69.55%	82.94%	77.42%
ETC [45]	PSPNet [75]	36.55%	58.29%	67.94%	84.10%	79.22%	33.84%	56.51%	69.43%	82.81%	77.06%
NetWarp [20]	PSPNet [75]	36.95%	57.93%	67.85%	84.36%	79.42%	33.68%	56.61%	69.10%	82.55%	77.09%
ETC [45]	OCRNet [72]	37.46%	59.13%	68.99%	84.10%	79.10%	34.55%	57.27%	69.25%	83.12%	78.00%
NetWarp [20]	OCRNet [72]	37.52%	58.94%	68.89%	84.00%	78.97%	35.00%	57.67%	70.23%	83.15%	77.21%
TCB $_{\text{st-ppm}}$	ResNet-101	37.46%	58.57%	70.30%	86.95%	82.12%	34.61%	57.25%	72.02%	85.19%	80.23%
TCB $_{\text{st-ocr}}$	ResNet-101	37.40%	59.26%	72.20%	86.88%	82.04%	35.12%	58.11%	73.86%	85.11%	80.12%
TCB $_{\text{st-ocr mem}}$	ResNet-101	**37.82%**	**59.49%**	**73.63%**	**87.86%**	**83.99%**	**35.62%**	**58.19%**	**74.33%**	**86.21%**	**81.90%**

Table 2. Comparison on the validation set and the test set. mVC_C means we use a clip with C frames.

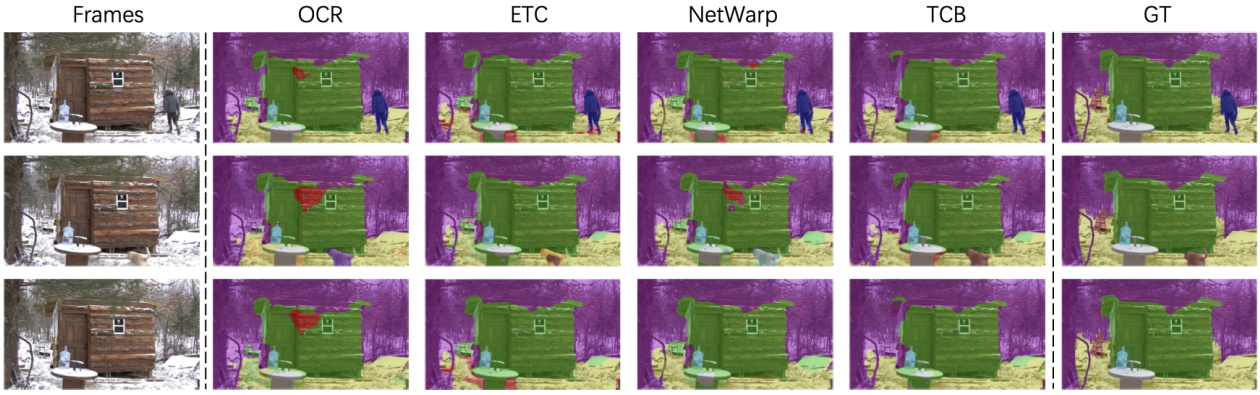

Figure 6. Qualitative comparisons of the segmentation results.

d1	d2	d3	Method	mIoU	TC	mVC_8
-	-	-	PSPNet	36.47%	65.89%	84.16%
1	-	-	TCB $_{\text{st-ppm}}$	36.94%	69.13%	85.35%
3	-	-	TCB $_{\text{st-ppm}}$	36.99%	69.52%	85.92%
6	-	-	TCB $_{\text{st-ppm}}$	36.84%	69.92%	85.50%
9	-	-	TCB $_{\text{st-ppm}}$	37.12%	70.01%	85.88%
1	2	3	TCB $_{\text{st-ppm}}$	37.40%	74.12%	86.31%
1	3	5	TCB $_{\text{st-ppm}}$	37.42%	72.04%	86.40%
3	6	9	TCB $_{\text{st-ppm}}$	37.46%	70.30%	86.95%
-	-	-	OCRNet	36.68%	66.21%	83.97%
1	-	-	TCB $_{\text{st-ocr mem}}$	37.20%	70.45%	86.52%
3	-	-	TCB $_{\text{st-ocr mem}}$	37.34%	69.72%	86.63%
6	-	-	TCB $_{\text{st-ocr mem}}$	37.21%	69.98%	86.65%
9	-	-	TCB $_{\text{st-ocr mem}}$	37.35%	69.52%	86.50%
1	2	3	TCB $_{\text{st-ocr mem}}$	37.45%	74.33%	87.39%
1	3	5	TCB $_{\text{st-ocr mem}}$	37.52%	72.19%	87.25%
3	6	9	TCB $_{\text{st-ocr mem}}$	37.82%	73.63%	87.86%

Table 3. The impact of the selection of the support frames.

eration can accumulate the contextual information for a long-range video clip or even the entire video. As shown in Table 2, adding the memory mechanism (TCB $_{\text{st-ocr mem}}$) can further improve the performance on both the segmentation and video stability. Compared with TCB $_{\text{st-ocr}}$, TCB $_{\text{st-ocr mem}}$ outperforms it by +0.42% (+0.50%) mIoU, +1.43% (+0.47%) TC and +0.98% (+1.10%) mVC_8 score on the validation (test) set, indicating that long-range contextual information is beneficial for the video scene parsing.

6. Conclusion and Future Work

This paper contributes a large-scale dataset for Video Scene Parsing in the Wild (VSPW), with diverse scenarios, high resolution, and a high frame rate. As far as we know, VSPW is the first attempt to tackle the challenging video scene parsing task by considering a wide range of diverse scenarios. Besides, we further propose a TCB network to harness long-range contextual information, which outperforms previous image-/video-based methods on VSPW.

However, there are several remained problems for the future work on our VSPW benchmark: (1) Efficiency and low-latency. Algorithms for the video scene parsing is required to use less computation cost and balanced latency. VSPW is a large-scale dataset with high resolutions. Thus the methods for low computation cost and low GPU-memory are needed to study. (2) Motion Blur. The learned models often fail to segment those frames with motion blurs, since the data distribution is different from the frames without motion blurs. (3) Appearance change or occlusion during motion. Even without motion blurs, when objects or cameras move faster, the appearance changes very fast, and the objects are easy to be occluded. Thus it is hard to keep video stability.

In sum, our VSPW poses many new challenges that are not well explored before. By releasing VSPW, we hope the new challenges can be extensively studied in the future.

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: real-time instance segmentation. In *IEEE ICCV*, pages 9157–9166, 2019. 2

[2] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, pages 44–57, 2008. 1, 2, 3, 4

[3] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295, 2010. 2

[4] 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. 2, 12

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *IEEE CVPR*, pages 1209–1218, 2018. 1, 2

[6] Joao Carreira, Viorica Patraucean, Laurent Mazare, Andrew Zisserman, and Simon Osindero. Massively parallel video networks. In *ECCV*, pages 649–666, 2018. 3

[7] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *IEEE ICCV*, 2020. 2

[8] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *IEEE CVPR*, pages 4013–4022, 2018. 2

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017. 2, 3

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 2, 3

[11] 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, 3, 5, 7, 8, 14, 15

[12] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *IEEE ICCV*, pages 2061–2069, 2019. 2

[13] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *IEEE CVPR*, pages 1189–1198, 2018. 2

[14] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas S Huang, Wen-Mei Hwu, and Honghui Shi. Spgnet: Semantic prediction guidance for scene parsing. In *IEEE ICCV*, pages 5218–5228, 2019. 2

[15] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *IEEE CVPR*, 2020. 2

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE CVPR*, pages 3213–3223, 2016. 1, 2, 3, 4

[17] 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. 5, 14

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1, 2

[19] Qingnan Fan, Fan Zhong, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *ACM Trans. Graph.*, 34(6):195–1, 2015. 2

[20] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video cnns through representation warping. In *ICCV*, pages 4453–4462, 2017. 2, 3, 7, 8, 14, 15

[21] Fabio Galasso, Naveen Shankar Nagaraja, Tatiana Jimenez Cardenas, Thomas Brox, and Bernt Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, pages 3527–3534, 2013. 2

[22] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *IEEE CVPR*, pages 5356–5364, 2019. 2

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE ICCV*, pages 2980–2988, 2017. 2

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 14

[25] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, pages 8818–8827, 2020. 3

[26] Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, and Thomas S. Huang. Ccnet: Criss-cross attention for semantic segmentation. *IEEE TPAMI*, 2020. 1, 2, 3, 5

[27] 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. 14

[28] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *CVPR*, pages 8866–8875, 2019. 3

[29] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, pages 656–671, 2014. 2

[30] Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, et al. Video scene parsing with predictive feature learning. In *ICCV*, pages 5580–5588, 2017. 3

[31] Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella X Yu. Adaptive affinity field for semantic segmentation. *ECCV*, 2018. 2

[32] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *IEEE CVPR*, pages 6399–6408, 2019. 2

[33] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 2

[34] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018. 6

[35] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, pages 2192–2199, 2013. 2

[36] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018. 2

[37] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *IEEE ICCV*, 2019. 2

[38] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *IEEE CVPR*, pages 7026–7035, 2019. 2

[39] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *IEEE CVPR*, pages 2359–2367, 2017. 2

[40] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *CVPR*, pages 5997–6005, 2018. 3

[41] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE CVPR*, 2017. 2

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 2

[43] Si Liu, Changhu Wang, Ruihe Qian, Han Yu, Renda Bao, and Yao Sun. Surveillance video parsing with single frame supervision. In *CVPR*, pages 1013–1021. IEEE Computer Society, 2017. 3

[44] Si Liu, Changhu Wang, Ruihe Qian, Han Yu, Renda Bao, and Yao Sun. Surveillance video parsing with single frame supervision. In *CVPR*, pages 413–421, 2017. 3

[45] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *ECCV*, 2020. 2, 3, 6, 7, 8, 13, 14, 15

[46] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE CVPR*, pages 3431–3440, 2015. 1, 2, 6

[47] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE TPAMI*, 41(6):1515–1530, 2018. 2

[48] Jiaxu Miao, Yunchao Wei, and Yi Yang. Memory aggregation networks for efficient interactive video object segmentation. In *IEEE CVPR*, 2020. 2

[49] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE CVPR*, pages 891–898, 2014. 2

[50] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *IEEE ICCV*, pages 4990–4999, 2017. 2

[51] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *CVPR*, pages 6819–6828, 2018. 3

[52] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE TPAMI*, 36(6):1187–1200, 2013. 2

[53] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters—improve semantic segmentation by global convolutional network. In *IEEE CVPR*, pages 1743–1751, 2017. 2

[54] 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, 14

[55] 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. 12

[56] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, pages 5639–5647, 2018. 14

[57] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *ECCV*, pages 852–868. Springer, 2016. 3

[58] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 1, 2, 3, 4

[59] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *IEEE ICCV*, pages 7355–7363, 2019. 2

[60] 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. 2, 3, 12

[61] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *IEEE CVPR*, pages 9481–9490, 2019. 2

[62] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In *CVPR Workshop*, volume 5, 2017. 2

[63] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE CVPR*, pages 7794–7803, 2018. 2, 3, 5

[64] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene

recognition from abbey to zoo. In *IEEE CVPR*, pages 3485–3492, 2010. 2

[65] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 2, 7, 8, 14, 15

[66] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *IEEE CVPR*, pages 8818–8826, 2019. 2

[67] 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. 2, 12

[68] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, pages 5188–5197, 2019. 2

[69] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *IEEE CVPR*, pages 3684–3692, 2018. 2

[70] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *arXiv preprint arXiv:1902.05093*, 2019. 2

[71] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, pages 332–348. Springer, 2020. 4, 12

[72] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, August 2020. 2, 3, 5, 7, 8, 14, 15

[73] Yuhui Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. 2, 3

[74] Rui Zhang, Sheng Tang, Yongdong Zhang, Jintao Li, and Shuicheng Yan. Scale-adaptive convolutions for scene parsing. In *IEEE ICCV*, pages 2031–2039, 2017. 2

[75] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE CVPR*, 2017. 1, 2, 3, 5, 6, 7, 8, 14, 15

[76] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, pages 270–286, 2018. 2

[77] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, pages 487–495, 2014. 3

[78] 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, 2, 3, 12

[79] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *CVPR*, pages 2349–2358, 2017. 3

[80] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *IEEE ICCV*, pages 593–602, 2019. 3

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 [4] and propagation based methods [71]. In this work, we tried both solutions. We follow [4] 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. 8 (a). As the propagation based method, we adopt the state-of-the-art method CFBI [71] and modified it to adapt our setting, as shown in Fig. 8 (b).

Concretely, for the finetuning-based method, we use HRNetV2 [60] pre-trained by ADE20k [78] 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 [71]. 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 [67] and DAVIS [55] jointly. For detail of the model architecture and training setting, please refer to [71].

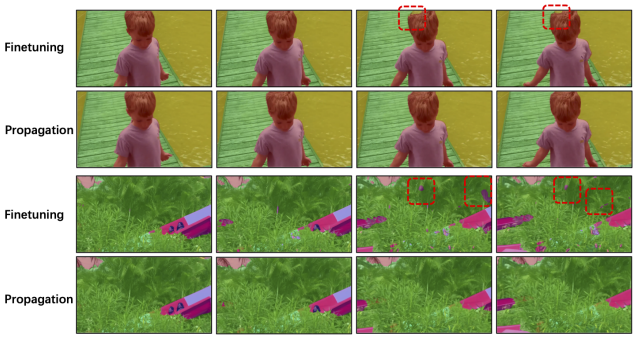

Figure 7. Qualitative comparison between the finetuning-based model and the propagation-based model.

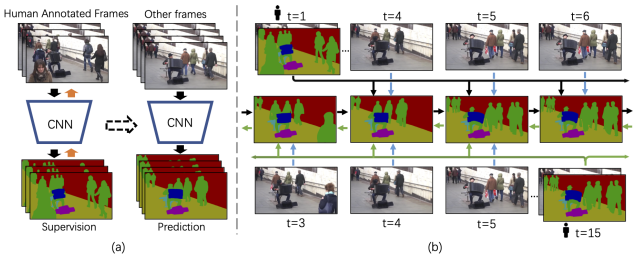

Figure 8. (a) Fine-tuning based model. (b) Propagation based model.

Table 4. Comparison of the finetuning-based model and the propagation based model.

Method	Mean IOU	Weighted IOU	Pixel Acc.	Pixel Acc. per Class
Finetuning [4]	82.76%	93.01%	96.19%	87.97%
Propagation [71]	89.82%	95.87%	97.86%	95.16%

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. 7 shows the qualitative comparisons.

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. 4 in the paper shows the comparison and the propagation model [71] significantly outperforms another one [4]. 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. 9 shows the ranked object category frequencies in the frame/video/pixel level, respectively. 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. 10 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. 11 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.

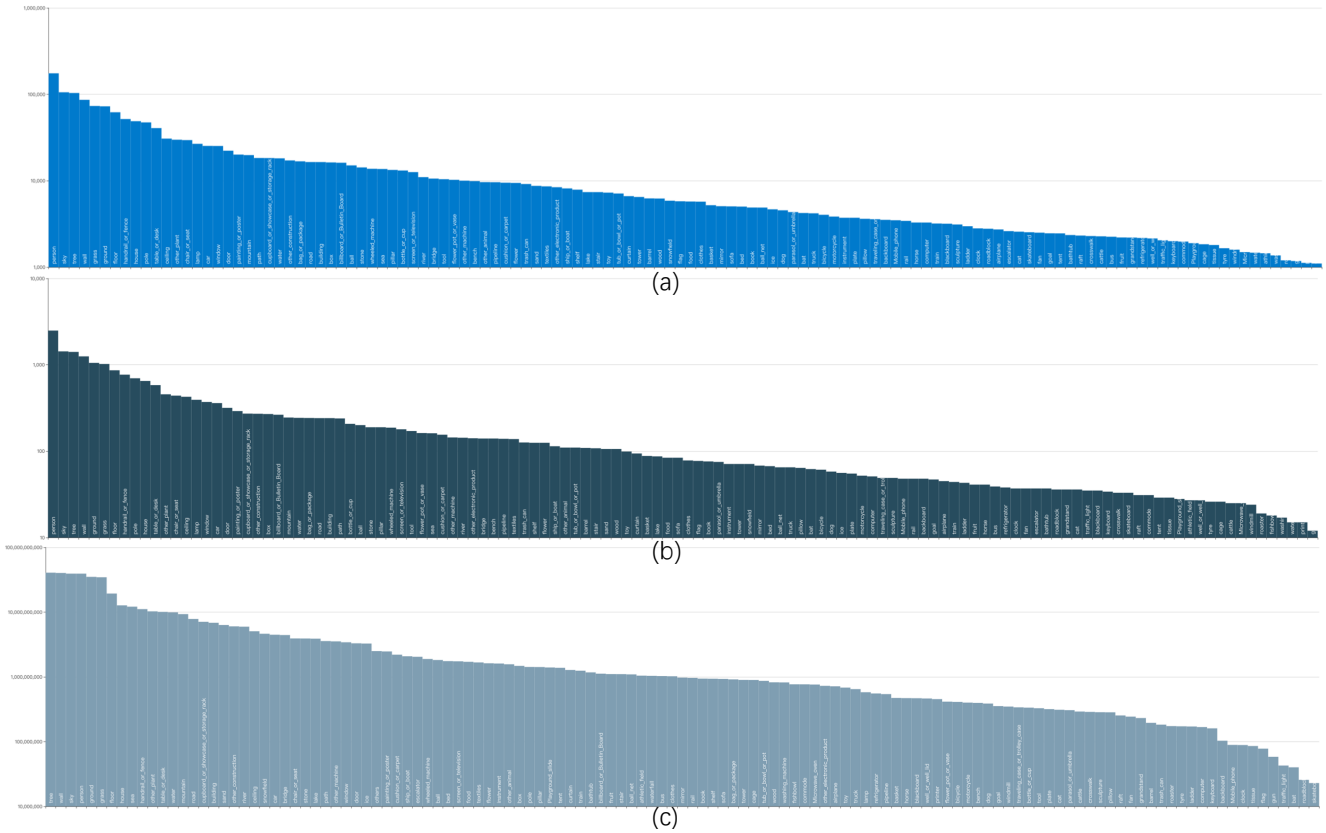

Figure 9. The ranked category frequencies in the (a) frame (b) video (c) pixel level.

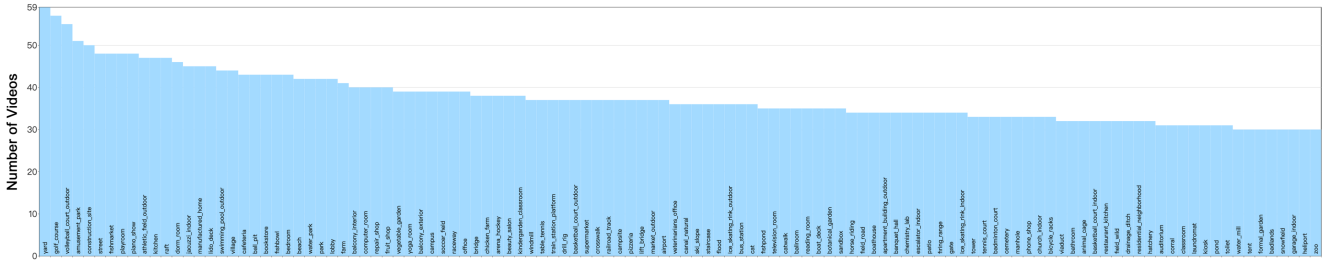

Figure 10. The distribution of videos per scene.

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. 6.

Following [45], 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,

$$TC(\mathbf{Q}_{t-1}, \mathbf{Q}_t) = \frac{\mathbf{Q}_t \cap \hat{\mathbf{Q}}_{t-1}}{\mathbf{Q}_t \cup \hat{\mathbf{Q}}_{t-1}}, \tag{5}$$

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:

$$TC = \frac{1}{N}\sum_{n=1}^{N}\frac{\mathcal{Q}_n \cap \hat{\mathcal{Q}}_n}{\mathcal{Q}_n \cup \hat{\mathcal{Q}}_n}, \tag{6}$$

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

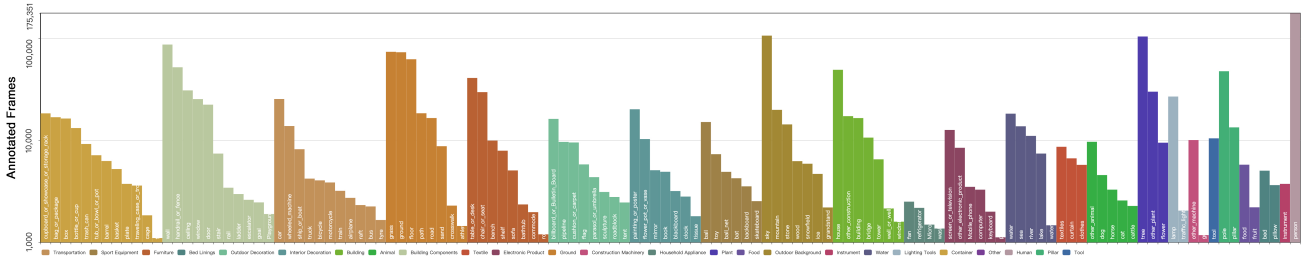

Figure 11. The histogram of pixels for parent classes and their subclasses.

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 [54]. In [54], 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. 5. TCB achieves better TS than image-based methods while similar to Netwarp and ETC.

Score	TCB$_{ocr}$	OCR	TCB$_{psp}$	PSP	Netwarp	ETC
TS $\downarrow$	0.299	0.351	0.303	0.344	0.296	0.301

Table 5. Comparison on temporal stability (VOS-metrics).

A.4 Implementation Details

We use ResNet-101 [24] as our backbone and initialize the backbone by the ImageNet [17] 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 [27] layer is replaced by the Synchronize BatchNorm [56] to collect the mean and standard-deviation of BatchNorm across multiple GPUs during training. For fair comparisons, all the comparable methods (PSPNet [75], UperNet [65], Deeplabv3+ [11] , OCRNet [72],NetWarp [20],ETC [45]) use the same training settings. For ETC [45], we use the ResNet-101 as the backbone without distillation, because we do not compare the efficiency in this paper.

(a) Results on the validation set.

Method	Backbone	mIOU	Weighted IOU	Pixel Acc.	Pixel Acc. per Class	TC	VC_8	VC_{16}
DeepLabv3+ [11]	ResNet-101	34.67%	58.81%	72.82%	45.48%	65.45%	83.24%	78.24%
UperNet [65]	ResNet-101	36.46%	58.60%	72.64%	47.35%	63.10%	82.55%	76.08%
PSPNet [75]	ResNet-101	36.47%	58.08%	72.34%	46.33%	65.89%	84.16%	79.63%
OCRNet [72]	ResNet-101	36.68%	59.24%	73.14%	47.12%	66.21%	83.97%	79.04%
ETC [45]	PSPNet [75]	36.55%	58.29%	72.41%	46.58%	67.94%	84.10%	79.22%
NetWarp [20]	PSPNet [75]	36.95%	57.93%	72.14%	47.09%	67.85%	84.36%	79.42%
ETC [45]	OCRNet [72]	37.46%	59.13%	72.99%	47.94%	68.99%	84.10%	79.10%
NetWarp [20]	OCRNet [72]	37.52%	58.94%	72.93%	47.72%	68.89%	84.00%	78.97%
TCB $_{\text{st-ppm}}$	ResNet-101	37.46%	58.57%	72.50%	47.59%	70.30%	86.95%	82.12%
TCB $_{\text{st-ocr}}$	ResNet-101	37.40%	59.26%	73.22%	48.55%	72.20%	86.88%	82.04%
TCB $_{\text{st-ocr memory}}$	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+ [11]	ResNet-101	32.15%	57.08%	70.86%	42.76%	70.01%	80.98%	75.02%
UperNet [65]	ResNet-101	33.46%	54.84%	70.26%	44.77%	66.32%	79.33%	73.29%
PSPNet [75]	ResNet-101	33.78%	56.38%	72.34%	46.23%	70.29%	83.35%	78.29%
OCRNet [72]	ResNet-101	34.02%	56.78%	70.91%	44.97%	69.55%	82.94%	77.42%
ETC [45]	PSPNet [75]	33.84%	56.51%	70.80%	44.28%	69.43%	82.81%	77.06%
NetWarp [20]	PSPNet [75]	33.68%	56.61%	70.82%	44.41%	69.10%	82.55%	77.09%
ETC [45]	OCRNet [72]	34.55%	57.27%	71.25%	45.67%	69.25%	83.12%	78.00%
NetWarp [20]	OCRNet [72]	35.00%	57.67%	71.63%	45.94%	70.23%	83.15%	77.21%
TCB $_{\text{st-ppm}}$	ResNet-101	34.61%	57.25%	71.31%	45.85%	72.02%	85.19%	80.23%
TCB $_{\text{st-ocr}}$	ResNet-101	35.12%	58.11%	72.17%	46.53%	73.86%	85.11%	80.12%
TCB $_{\text{st-ocr memory}}$	ResNet-101	35.62%	58.19%	72.21%	46.88%	74.33%	86.21%	81.90%

Table 6. Comparison on the validation set and the test set. VC_C means we use a clip number C.