Coarse-to-Fine Feature Mining for Video Semantic Segmentation

Guolei Sun\textsuperscript{1}  \hspace{1em} Yun Liu\textsuperscript{1,}\textsuperscript{*}  \hspace{1em} Henghui Ding\textsuperscript{1}  \hspace{1em} Thomas Probst\textsuperscript{1}  \hspace{1em} Luc Van Gool\textsuperscript{1,2}

\textsuperscript{1} Computer Vision Laboratory, ETH Zurich, \textsuperscript{2} VISICS, KU Leuven

Abstract

The contextual information plays a core role in semantic segmentation. As for video semantic segmentation, the contexts include static contexts and motional contexts, corresponding to static content and moving content in a video clip, respectively. The static contexts are well exploited in image semantic segmentation by learning multi-scale and global/long-range features. The motional contexts are studied in previous video semantic segmentation. However, there is no research about how to simultaneously learn static and motional contexts which are highly correlated and complementary to each other. To address this problem, we propose a Coarse-to-Fine Feature Mining (CFFM) technique to learn a unified presentation of static contexts and motional contexts. This technique consists of two parts: coarse-to-fine feature assembling and cross-frame feature mining. The former operation prepares data for further processing, enabling the subsequent joint learning of static and motional contexts. The latter operation mines useful information/contexts from the sequential frames to enhance the video contexts of the features of the target frame. The enhanced features can be directly applied for the final prediction. Experimental results on popular benchmarks demonstrate that the proposed CFFM performs favorably against state-of-the-art methods for video semantic segmentation. Our implementation is available at \url{https://github.com/GuoleiSun/VSS-CFFM}.

1. Introduction

Semantic segmentation aims at assigning a semantic label to each pixel in a natural image, which is a fundamental and hot topic in the computer vision community. It has wide range of applications in both academic and industrial fields. Thanks to the powerful representation capability of deep neural networks [28, 39, 71, 73] and large-scale image datasets [3, 14, 20, 59, 102], tremendous achievements have been seen for image semantic segmentation. However, video semantic segmentation has not been witnessed such tremendous progress [23, 35, 53, 60] due to the lack of large-scale datasets. For example, Cityscapes [14] and NYUDv2 [70] datasets only annotate one or several nonadjacent frames in a video clip. CamVid [2] only has a small scale and a low frame rate. The real world is actually dynamic rather than static, so the research on video semantic segmentation (VSS) is necessary. Fortunately, the recent establishment of the large-scale video segmentation dataset, VSPW [58], solves the problem of video data scarcity. This inspires us to denoting our efforts to VSS.

As widely accepted, the contextual information plays a central role in image semantic segmentation [5–7, 9, 16, 27, 33, 36, 37, 45, 50, 88, 90, 93, 96, 98, 99, 103, 107]. When considering videos, the contextual information is twofold: static contexts and motional contexts, as shown in Fig. 1. The former refers to the contexts within the same video frame or the contexts of unchanged content across different frames. Image semantic segmentation has exploited such contexts (for images) a lot, mainly accounting for multi-scale [6, 7, 9, 88] and global/long-range information [33, 96, 98, 107]. Such information is essential not only for understanding the static scene but also for perceiving the holistic environment of videos. The latter, also known as temporal information, is responsible for better parsing moving object/stuff and capturing more effective scene representations with the help of motions. The motional context

*The corresponding author: yun.liu@vision.ee.ethz.ch
learning has been widely studied in video semantic segmentation [4, 23, 32, 34, 35, 42, 47, 53, 54, 60, 69, 85, 105], which usually relies on optical flows [19] to model motional contexts, ignoring the static contexts. Although each single aspect, i.e., static or motional contexts, has been well studied, how to learn static and motional contexts simultaneously deserves more attention, which is important for VSS.

Furthermore, static contexts and motional contexts are highly correlated, not isolated, because both contexts are complementary to each other to represent a video clip. Therefore, the ideal solution for VSS is to jointly learn static and motional contexts, i.e., generating a unified representation of static and motional contexts. A naïve solution is to apply recent popular self-attention [18, 76, 80] by taking feature vectors at all pixels in neighboring frames as tokens. This can directly model global relationships of all tokens, of course including both static and motional contexts. However, this naïve solution has some obvious drawbacks. For example, it is super inefficient due to the large number of tokens/pixels in a video clip, making this naïve solution unrealistic. It also contains too much redundant computation because most content in a video clip usually does not change much and it is unnecessary to compute attention for the repeated content. Moreover, the too long length of tokens would affect the performance of self-attention, as shown in [12, 21, 29, 56, 84] where the reduction of the token length through downsampling leads to better performance. More discussion about why traditional self-attention is inappropriate for video context learning can be found in §3.1.

In this paper, we propose a new Coarse-to-Fine Feature Mining (CFFM) technique, which consists of two parts: coarse-to-fine feature assembling and cross-frame feature mining. Specifically, we first apply a lightweight deep network [83] to extract features from each frame. Then, we assemble the extracted features from neighbouring frames in a coarse-to-fine manner. Here, we use a larger receptive field and a more coarse pooling if the frame is more distant from the target. This feature assembling operation has two meanings. On one hand, it organizes the features in a multi-scale way, and the farthest frame would have the largest receptive field and the most coarse pooling. Since the content in a few sequential frames usually does not change suddenly and most content may only have a little temporal inconsistency, this operation is expected to prepare data for learning static contexts. On the other hand, this feature assembling operation enables a large perception region for remote frames because the moving objects may appear in a large region for remote frames. This makes it suitable for learning motional contexts. At last, with the assembled features, we use the cross-frame feature mining technique to iteratively mine useful information from neighbouring frames for the target frame. This mining technique is a specially-designed non-self attention mechanism that has two different inputs, unlike commonly-used self-attention that only has one input [18, 76]. The output features enhanced by the CFFM can be directly used for the final prediction. We describe the technical motivations for CFFM in detail in §3.1.

The advantages of this new video context learning mechanism are four-fold. (1) The proposed CFFM technique can learn a unified representation of static contexts and motional contexts, both of which are of vital importance for VSS. (2) The CFFM technique can be added on top of frame feature extraction backbones to generate powerful video contextual features, with low complexity and limited computational cost. (3) Without bells and whistles, we achieve state-of-the-art results for VSS on standard benchmarks by using the CFFM module. (4) The CFFM technique has the potential to be extended to improve other video recognition tasks that need powerful video contexts.

2. Related Work

2.1. Image Semantic Segmentation

Image semantic segmentation has always been a key topic in the vision community, mainly because of its wide applications in real-world scenarios. Since the pioneer work of FCN [68] which adopts fully convolution networks to make densely pixel-wise predictions, a number of segmentation methods have been proposed with different motivations or techniques [10, 30, 31, 52, 61, 63, 78, 81, 94, 97, 104]. For example, some works try to design effective encoder-decoder network architectures to exploit multi-level features from different network layers [1, 9, 25, 51, 62, 66, 68, 75]. Some works impose extra boundary supervision to improve the prediction accuracy of details [44, 74, 77, 89, 99]. Some works utilize the attention mechanism to enhance the semantic representations [8, 22, 33, 46, 67, 95, 101, 107]. Besides these talent works, we want to emphasize that most research aims at learning powerful contextual information [5, 17, 27, 36, 37, 45, 50, 90, 93, 96, 99, 103], including multi-scale [6, 7, 9, 26, 27, 45, 88] and global/long-range information [33, 96, 98, 107]. The contextual information is also essential for VSS, but the video contexts are different from the image contexts, as discussed above.

2.2. Video Semantic Segmentation

Since the real world is dynamic rather than static, VSS is necessary for pushing semantic segmentation into more practical deployments. Previous research on VSS was limited by the available datasets [58]. Specifically, three datasets were available: Cityscapes [14], NYUDv2 [70], and CamVid [2]. They either only annotate several nonadjacent frames in a video clip or have a small scale, a low frame rate and low resolution. In fact, these datasets are usually used for image segmentation. Fortunately, the recent establishment of the VSPW dataset [58] which is large-scale and
fully-annotated solves this problem. Most of the existing VSS methods utilize the optical flow to capture temporal relations [23, 34, 40, 42, 53, 54, 57, 60, 69, 85, 105]. These methods usually adopt different smart strategies to balance the trade-off between accuracy and efficiency [40, 57]. Among them, some works aim at improving the segmentation accuracy by exploiting the temporal relations using the optical flow for feature warping [23, 53, 60], or the GAN-like architecture [24] for predictive feature learning [35]. The other works aim at improving the segmentation efficiency by using temporal consistency for feature propagation and reuse [34, 47, 69, 105], or directly reusing high-level features [4, 69], or adaptively selecting the key frame [85], or propagating segmentation results to neighbouring frames [42], or extracting features from different frames with different sub-networks [32], or considering the temporal consistency as extra training constraints [54]. Zhu et al. [106] utilized video prediction models to predict future frames as well as future segmentation labels, which are used as augmented data for training better image semantic segmentation models, not for VSS. Different from the above approaches, STT [43] and LMANet [65] directly models the interactions between the target and reference features to exploit the temporal information.

The above VSS approaches explore the temporal relation, here denoted as motional contexts. However, video contexts include two highly-correlated aspects: static and motional contexts. Those methods ignore the static contexts that are important for segmenting complicated scenes. This paper addresses this problem by proposing a new video context learning mechanism, capable of joint learning a unified representation of static and motional contexts.

2.3. Transformer

Vision transformer, a strong competitor of CNNs, has been widely adopted in various vision tasks [18, 49, 55, 56, 72, 87, 91, 92, 100], due to its powerful ability of modeling global connection within all the input tokens. Specifically, ViT [18] splits images into patches, constructs tokens and processes tokens using typical transformer layers. Swin [56] improves ViT by introducing shifted windows when computing self-attention and a hierarchical architecture. Focal [87] introduces both fine-grained and coarse-grained attention in architecture design. The effectiveness of transformers has been validated in segmenta- tion [83, 100], tracking [11, 86], crowd counting [48, 72], multi-label classification [41] and so on. Despite the success of transformer in these tasks, the use of transformer layers in VSS is non-trivial due to the large number of tokens among video frames. Here, we propose a effective and efficient way to model the contextual information for VSS.

3. Methodology

3.1. Technical Motivation

Before introducing our method, we discuss our technical motivation to help readers better understand the proposed technique. As discussed above, video contexts include static contexts and motional contexts. The former is well exploited in image semantic segmentation [5–7, 9, 15, 26, 27, 33, 36, 37, 45, 50, 88, 90, 93, 96, 98, 99, 103, 107], while the latter is studied in video semantic segmentation [4, 23, 34, 40, 42, 47, 53, 54, 57, 60, 69, 85, 105]. However, there is no research touching the joint learning of both static and motional contexts which are both essential for VSS.

To address this problem, a naïve solution is to simply apply the recently popular self-attention mechanism [18, 76, 80] to the video sequence by viewing the feature vector at each pixel of each frame as a token. In this way, we can model global relationships by connecting each pixel with all others, so all video contexts can of course be constructed. However, this naïve solution has three obvious drawbacks. First, a video sequence has \( l \) times more tokens than a single image, where \( l \) is the length of the video sequence. This would lead to \( l^2 \) times more computational cost than a single image because the complexity of the self-attention mechanism is \( O(N^2C) \), where \( N \) is the number of tokens and \( C \) is the feature dimension [18, 56, 76]. Such high complexity is unaffordable, especially for VSS that needs on-time processing as video data stream comes in sequence. Second, such direct global modeling would be redundant. Despite that there are some motions in a video clip, the overall semantics/environment would not change suddenly and most video content is repeated. Hence, most of connections built by the direct global modeling are unnecessary, i.e., self-to-self connections. Last but not least, although self-attention can technically model global relationships, a too long sequence length would limit its performance, as demonstrated in [12, 21, 29, 56, 79, 84] where downsampling features into small scales leads to better performance than the original long sequence length.

Instead of directly modeling global relationships, we propose to model relationships only among necessary tokens for the joint learning of static and motional contexts. Our CFFM technique consists of two steps. The first step, Coarse-to-Fine Feature Assembling (CFFA), assembles the features extracted from neighbouring frames in a temporally coarse-to-fine manner based on three observations. First, the moving objects/stuff can only move gradually across frames in practice, and the objects/stuff cannot move from one position to another far position suddenly. Thus, the region of the possible positions of (an) moving object/stuff in a frame gradually gets larger for farther frames. In other words, for one pixel in a frame, the farther the frames, the larger the correlated regions. Second, although some
content may change across frames, the overall semantics and environment would not change much, which means that most video content may only have a little temporal inconsistency. Third, the little temporal inconsistency of the “static” content across neighbouring frames can be easily handled by the pooling operation which is scale- and rotation-invariant. Inspired by the second and third observations, a varied-size region sampling through the pooling operation in neighbouring frames can convey multi-scale contextual information. Therefore, the designed CFFA can perceive multi-scale contextual information (static contexts) and motional contexts. Specifically, each pixel in the target frame corresponds to a larger receptive field and a more coarse pooling in the farther frame, as depicted in Fig. 2. Note that the length of the sampled tokens is much shorter than that in the default self-attention.

The second step of CFFM, Cross-frame Feature Mining (CFM), is designed to mine useful information from the features of neighbouring frames. This is an attention-based process. However, unlike traditional self-attention [18,76,80] whose query, key, and value come from the same input, we propose to use a non-self-attention mechanism, where the query is from the target frame and the key and value are from neighbouring frames. Besides, we only update the query during the iterative running of non-self-attention, but we keep the context tokens unchanged. This is intuitive as our goal is to mine information from neighbouring frames and the update of context tokens is thus unnecessary. Compared with self-attention that needs to concatenate and process all assembled features, this non-self attention further reduces the computational cost.

### 3.2. Coarse-to-Fine Feature Assembling

Without loss of generalizability, we start our discussion on training data containing video frames \{I_{t-k}, \ldots , I_{t-1}, I_t\} with ground-truth segmentation of \{S_{t-k}, \ldots , S_{t-1}, S_t\}, and we focus on segmenting I_t. Specifically, I_t is the target frame and \{I_{t-k}, \ldots , I_{t-1}\} are \ell \ previous frames which are \{k_1, \ldots, k_{\ell}\} frames away from I_t, respectively. Let us denote \(U = \{t - k_1, \ldots , t - k_{\ell}, t\}\) as the set of all frame subscripts. We first process \{I_{t-k_1}, \ldots , I_{t-1}, I_t\} using an encoder to extract informative features \(F = \{F_{t-k_1}, \ldots , F_{t-1}, F_t\}\), each of which has the size of \(\mathbb{R}^{h \times w \times c}\) (\(h, w, c\) represent height, width, and the number of channels, respectively). We aim to exploit \(F\) to generate better features for segmenting \(I_t\) as relevant and valuable video contexts exist in previous frames.

To efficiently establish long-range interactions between the reference frame features \(\{F_{t-k_1}, \ldots , F_{t-1}\}\) and the target frame features \(F_t\), we propose the coarse-to-fine feature assembling module, as showed in Fig. 2. Inspired by previous works [56, 79, 87], we split the target frame features \(F_t\) into windows and each window attends to a shared set of context tokens. The reason behind this is that attending each location in \(F_t\) to a specific set of context tokens requires huge computation and memory cost. When using window size of \(s \times s\), \(F_t\) is partitioned into \(\frac{h}{s} \times \frac{w}{s}\) windows.
We obtain the new feature map $F^j_t$ as follows:

$$F^i_t \in \mathbb{R}^{h \times w \times c} \rightarrow F^j_t \in \mathbb{R}^{(\frac{h}{p_2}) \times (\frac{w}{p_1}) \times h \times w \times c} \rightarrow F'_{t} \in \mathbb{R}^{\frac{h}{p_2} \times \frac{w}{p_1} \times h \times w \times c}.$$  

(1)

Then, we generate context tokens from different frames. The main idea is to see a bigger receptive field and use a more coarse pooling if the frame is more distant from the target, which is why we call this step coarse-to-fine feature assembling. The motivation behind this is described in §3.1. Formally, we define two sets of parameters: the receptive fields $r = \{r_{t-k_1}, \cdots, r_{t-k_2}, r_t\}$ and the pooling kernel/window sizes $p = \{p_{t-k_1}, \cdots, p_{t-k_2}, p_t\}$, when generating corresponding context tokens. For $t - k_1 < t - k_2 < \cdots < t - k_i < t$, we have $r_{t-k_1} \geq r_{t-k_2} \geq \cdots \geq r_{t-k_i} \geq r_t$ and $p_{t-k_1} \geq p_{t-k_2} \geq \cdots \geq p_{t-k_i} \geq p_t$. With this definition, we partition the $\{F_{t-k_1}, \cdots, F_{t-k_i}, F_t\}$ using pooling windows $p = \{p_{t-k_1}, \cdots, p_{t-k_i}, p_t\}$ to pool the features, respectively. The result is processed by a fully connected layer (FC) for dimension reduction. This is formulated as

$$F^i_{j} \in \mathbb{R}^{h \times w \times c} \rightarrow E^i_{j} \in \mathbb{R}^{\frac{h}{p_1} \times \frac{w}{p_2} \times h \times w \times c} \rightarrow F'_{t} \in \mathbb{R}^{\frac{h}{p_2} \times \frac{w}{p_1} \times c}.$$  

(2)

where $j \in U$. In Fig. 2, we have $r = \{20, 12, 6, 4\}$ and $p = \{4, 3, 2, 1\}$ for all frames (3 reference and 1 target).

For each window partition $F^i_{t} \in \mathbb{R}^{h \times w \times c}$ (i.e., $i \in \{1, 2, \cdots, \frac{h \times w}{p_2 \times p_1}\}$) in the target features, we extract $c_{i,j}$ elements from $E^i_{j}$ around the area where the window lies in. This can be easily implemented using the unfold function in PyTorch [64]. Let $c_{i,j}$ denote the obtained context tokens from j-th frame and for i-th window partition in the target features. We concatenate $c_{i,j}$ into $c_i$ as follows,

$$c_i = \text{Concat}[c_{i,j}],$$  

(3)

where $j \in U$, $c_i \in \mathbb{R}^{m \times c}$ and $m = \sum_{j \in U} \frac{r^2_i}{p_1 \times p_2}$. The context tokens from the target frame are obtained by using parameter set $(r_i, p_i)$ to process the target features. In practice, we additionally use another parameter set $(r_i', p_i')$ to generate more contexts from the target since the target features are more important. For simplicity, we focus our discussion by omitting $(r_i', p_i')$ and using only $(r_i, p_i)$ for the target.

To sum up, $c_i$ contains the context information from all frames, which is used to refine the target frame features. As discussed in §3.1, on one hand, $c_i$ covers the tokens at possible positions that moving objects/stuff would appear, so it can be used for learning motional contexts. On the other hand, $c_i$ is a multi-scale sampling of neighbouring frames with the temporal inconsistency solved by the pooling operation, so it can be used for learning static contexts.

3.3. Cross-frame Feature Mining

After that we obtain the context tokens $c_i$, for each window partition in the target features, we propose a non-self attention mechanism to mine useful information from neighboring frames. Unlike the traditional self-attention mechanism that computes the query, key, and value from the same input, our non-self attention mechanism utilizes different inputs to calculate the query, key, and value. Since $F_t$ is the input to the first layer of our cross-frame feature mining module, we re-write it as $F_t^j = F'_t$. For the i-th window partition in $F_t^0$, the query $Q_i$, key $K_i$, and value $V_i$ are computed using three fully connected layers as follows:

$$Q_i = FC(F_t^i[i]), \quad K_i = FC(c_i), \quad V_i = FC(c_i),$$  

(4)

where $FC(\cdot)$ represents a FC layer. Next, we use non-self attention to update the target frame features, given by

$$F'_t[i] = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{c}} + B\right) V_i + F_t^i[i],$$  

(5)

where $B$ represents the position bias, following [56]. Note that we omit the formulation of the multi-head attention [18, 76] for simplicity. Equ. (4) and Equ. (5) are repeated for $N$ steps, and we finally obtain the enhanced feature $F_t^N \in \mathbb{R}^{\frac{h}{p_2} \times \frac{w}{p_1} \times h \times w \times c}$ for the target frame. Long-range static and motional contexts from neighbouring frames are continuously exploited to learn better features for segmenting the target frame. Note that in this process, we do not update the context tokens $c_i$ for simplicity/elegance and reducing computation. Since this step is to mine useful information from the reference frames, it is also unnecessary to update $c_i$. This is the advantage of non-self attention.

To generate segmentation predictions, we reshape $F_t^N$ into $\mathbb{R}^{h \times w \times c}$ and concatenate $F_t^N$ with $F_t$. Then, a simple MLP projects the features to segmentation logits. The common cross entropy (CE) is used as the loss function for training. Auxiliary losses on original features are also computed. During inference, our method does not need to extract features for all $l + 1$ frames when processing $I_l$. Instead, the features of the reference frames, which are the frames before the target frame, have already been extracted in previous steps. Only the target frame is passed to the encoder to generate $F_t$, and then features $\{F_{t-k_1}, \cdots, F_{t-k_i}, F_t\}$ for all frames are passed to the CFFM.

3.4. Complexity Analysis

Here, we formally analyze the complexity of the proposed CFFM and the recent popular self-attention mechanism [18, 76, 80] when processing video clip features $\{F_{t-k_1}, \cdots, F_{t-k_i}, F_t\}$. The coarse-to-fine feature assembling (Equ. (2)) has the complexity of $O((l+1)hwmc)$, which is irrespective of $p$. The cross-frame feature mining has two parts: Equ. (4) has the complexity of $O(hwmc^2) + O(mc^2)$, and Equ. (5) is with the complexity of $O(hwmc)$. As mentioned early, $m = \sum_{j \in U} \frac{r_i^2}{p_1 \times p_2}$. To sum over, the complexity of our method is given by

$$O(\text{CFFM}) = O(hwmc) + O(hwmc^2) + O(mc^2) + O((l + 1)hwmc) = O(hwmc) + O(hwmc^2),$$  

(6)
where the derivation is conducted by removing less significant terms. For the self-attention mechanism [18,76,80], the complexity is $O((l + 1)^2 h w^2 c) + O((l + 1) h w c^2)$. Since $m \ll (l + 1)^2 h w$, the complexity of the proposed approach is much less than the self-attention mechanism. Take the example in Fig. 2, $m = 66$ while $(l + 1)^2 h w = 6400$.

### 3.5. Difference with STT

We notice that a concurrent work STT [43] also utilizes bigger searching regions for more distant frames and self-attention mechanisms to establish connections across frames. While two works share these similarities, there are key differences between them. First, two methods have different motivations. We target at exploiting both static and motional contexts, while STT focuses on capturing the temporal relations among complex regions. Note that the concept of static/motional contexts is similar to the concept of simple/complex regions in STT. As a result, STT models only the motional contexts, while our method models both static and motional contexts. Second, the designs are different. For query selection, STT selects 50% of query locations in order to reduce the computation. However, our method splits the query features into windows and the query features in each window share the same contexts to reduce the computation. For key/value selection, STT operates in the same granularity, while our method processes the selected key/value into different granularity, which reduces the number of tokens and models the multi-scale information for static contexts. Third, our cross-frame feature mining can exploit multiple transformer layers to deeply mine the contextual information from the reference frames, but STT only uses one layer. The reason may be that STT only updates the query features of the selected locations and using multiple STT layers could lead to inconsistency in the query features in un-selected and selected locations.

### 4. Experiments

#### 4.1. Experimental Setup

**Implementation details.** We implement our approach based on the mmsegmentation [13] codebase and conduct all experiments on 4 NVIDIA GPUs. The backbones are the same as SegFormer [83], which are all pretrained on ImageNet [39]. For other parts of our model, we adopt random initialization. Our model uses 3 reference frames unless otherwise specified, and $\{k_1, k_2, k_3\} = \{9, 6, 3\}$, following [58]. We found that this selection of reference frames is enough to include rich context and achieve impressive performance. For the receptive field, pooling kernel and window size, we set $r = \{49, 20, 6, 7\}$, $p = \{7, 4, 2, 1\}$ and $s = 7$. For the target frame, we additionally have $r_t = 35$ and $p_t = 5$. During training, we adopt augmentations including random resizing, flipping, cropping, and photometric distortion. We use the crop size of 480 $\times$ 480 for the VSPW dataset [58] and 512 $\times$ 1024 for Cityscapes [14]. For optimizing parameters, we use the AdamW and “poly” learning rate schedule, with an initial learning rate of 6e-5. During testing, we conduct single-scale test and resize all images on VSPW to the size of 480 $\times$ 853 and 512 $\times$ 1024 for Cityscapes. Note that for efficiency and simplicity, the predicted mask is obtained by feeding the whole image to the network, rather than using sliding window as in [100]. We do not use any post-processing such as CRF [38].

**Datasets.** Our experiments are mainly conducted on the VSPW dataset [58], which is the largest video semantic segmentation benchmark. Its training, validation and test sets have 2,806 clips (198,244 frames), 343 clips (24,502 frames), and 387 clips (28,887 frames), respectively. It contains diverse scenarios including both indoor and outdoor scenes, annotated for 124 categories. More importantly, VSPW has dense annotations with a high frame rate of 15fps, making itself the best benchmark for video semantic segmentation till now. In contrast, previous datasets used for video semantic segmentation only have very sparse annotation, i.e., only one frame out of many consecutive frames is annotated. In addition, we also evaluate the proposed method on the Cityscapes dataset [14], which annotates one frame out of every 30 frames.

**Evaluation metrics.** Following previous works [68], we use mean IoU (mIoU), and weighted IoU to evaluate the segmentation performance. In addition, we also adopt video consistency (VC) [58] to evaluate the smoothness of the predicted segmentation maps across the temporal domain. Formally, for a video clips $\{I_c\}_{c=1}^C$ with ground truth masks $\{S_c\}_{c=1}^C$ and predicted masks $\{S'_c\}_{c=1}^C$, VC is computed as follows,

$$VC_n = \frac{1}{C - n + 1} \sum_{i=1}^{C-n+1} \frac{(\cap_{i}^{n} S_i) \cap (\cup_{i}^{n} S'_i)}{\cap_{i}^{n} S_i}$$

where $C \geq n$. After computing VC, for every video, we obtain the mean of VC$_n$ for all videos as mVC$_n$. The purpose of this metric is to evaluate the level of consistency in the predicted masks among those common areas (pixels’ semantic labels don’t change) across long-range frames. For more details, please refer to [58]. Note that to compute VC metric, the ground-truth masks for all frames are needed.

#### 4.2. Comparison with State-of-the-art Methods

We compare the proposed method with state-of-the-art algorithms on VSPW [58] in Tab. 1. The results are analyzed from different aspects. For small models (# of parameters < 20M), our method outperforms corresponding baseline with a clear margin, while introducing limited model complexity. For example, using the backbone MiT-B0, we obtain 2.5% mIoU gain over the strong baseline of SegFormer [83], with the cost of increasing the parameters from...
13.8M to 15.5M and reducing the FPS from 73.4 to 43.1. Our method also provides much more consistent predictions for the videos, outperforming the baseline with 5.0% and 5.6% in mVC₈ and mVC₁₆, respectively.

For large models, our approach achieves the new state-of-the-art performance in this challenging dataset and also generates visually consistent results. Specifically, our model with 26.5M parameters (slightly larger than SegFormer [83] with MiT-B2) achieves 44.9% mIoU at the frame rate of 23.8fps. Our large model (based on MiT-B5) achieves mIoU of 49.3% and performs best in terms of visual consistency, with mVC₈ and mVC₁₆ of 90.8% and 87.1%, respectively. For all backbones (MiT-B0, MiT-B1, MiT-B2 and MiT-B5), CFFM clearly outperforms the corresponding baselines, showing that the proposed modules are stable. The results validate the effectiveness of the proposed coarse-to-fine feature assembling and cross-frame feature mining in mining informative contexts from all frames.

For Cityscapes [14] dataset, our method is compared with recent efficient segmentation methods. Only using 4.6M parameters, our model obtains 74.0% mIoU with frame rate of 34.2fps, achieving an excellent balance on model size, performance and speed. Note that this dataset has sparse annotations, the excellent performance demonstrates that our method works well for both fully supervised and semi-supervised settings.

The qualitative results are shown in Fig. 3. For the given example, our method resolves the inconsistency existing in the predictions of the baseline, due to the use of rich contextual information from the all frames.

### 4.3. Ablation Study

All ablation studies are conducted on the large-scale VSPW [58] dataset and follows the same training strategies.
as described above, for fair comparison.

**Influence of the number of attention layers.** Tab. 3 shows the performance of our method with respect to the number of non-self attention layers in cross-frame feature mining module. For two backbones of MiT-B0 [83] and MiT-B1 [83], our method clearly outperforms the corresponding baseline (SegFormer) when only using a single attention layer and introducing a small amount of additional parameters. It demonstrates the effectiveness of the proposed coarse-to-fine feature assembling module and the attention layer. The former efficiently extracts the context information from the frames and the latter effectively mine the information to refine target features. In addition, we observe there is a trade-off between performance and the model complexity on MiT-B1 backbone. When using more attention layers, better mIoU is obtained while the model size linearly increases. For our method on MiT-B1, we choose N = 2 since better trade-off is obtained.

**Impact of selection of reference frames.** We study the impact of the selection of reference frames in Tab. 4. We start by using a single reference frame. There seems to be a trend that when increasing the distance between the reference frame and the target frame, better performance is obtained. The possible reason for this is that the far-away reference frame may contain richer and different context which complements the one of the target frame. When using more reference frames \((k_1 = 9, k_2 = 6, k_3 = 3)\), the best performance is achieved. It is worthy noting that the reference frames combination of \(k_1 = 3, k_2 = 2,\) and \(k_3 = 1,\) performs similarly as the cases when using single reference frame, possibly due to the fact that the close reference frames don’t give much new information for segmenting the target frame.

**Impact of CFFA and CFM** Starting from SegFormer [83], we first add CFFA by using MLP to process the context tokens and then merge them with the target features. We obtain mIoU of 37.6. Then we add both CFFA and CFM on the baseline, which is our final model. The segmentation mIoU is 38.5. It shows that both CFFA and CFM are valuable for the proposed CFFM mechanism.

## 5. Conclusion

The video contexts include *static contexts* and *motional contexts*, both of which are essential for video semantic segmentation. Previous methods pay much attention to *motional contexts* but ignore the *static contexts*. To this end, this paper proposes a Coarse-To-Fine Feature Mining (CFFM) technique to jointly learn a unified presentation of static and motional contexts, for precise and efficient VSS. CFFM contains two parts: coarse-to-fine feature assembling and cross-frame feature mining. The former summarizes contextual information with different granularity for different frames, according to their distance to the target frame. The latter efficiently mines the contexts from neighbouring frames to enhance the feature of the target frame. While adding limited computational resources, CFFM boosts segmentation performance in a clear margin.
References


[29] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial

[95] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. ACFNet: Attentional class feature network for semantic segmentation. In IEEE ICCV, pages 6798–6807, 2019. 2


[107] 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. 1, 2, 3