Exploring Temporal Feature Correlation for Efficient and Stable Video Semantic Segmentation

Matthieu Lin*[1], Jenny Sheng*[1], Yubin Hu[1], Yangguang Li[2], Lu Qi[3], Andrew Zhao[4], Gao Huang[4], Yong-Jin Liu[1]

[1] BNRist, Department of Computer Science and Technology, Tsinghua University
[4] BNRist, Department of Automation, Tsinghua University
[2] SenseTime Group Limited
[3] The University of California, Merced
{yh-lin21, cqq22, huyb20, zqc21}@mails.tsinghua.edu.cn, {liuyongjin, gaohuang}@tsinghua.edu.cn,
liyangguang256@gmail.com, luqi@cse.cuhk.edu.hk

Abstract

This paper tackles the problem of efficient and stable video semantic segmentation. While stability has been under-explored, prevalent work in efficient video semantic segmentation uses the keyframe paradigm. They efficiently process videos by only recomputing the low-level features and reusing high-level features computed at selected keyframes. In addition, the reused features stabilize the predictions across frames, thereby improving video consistency. However, dynamic scenes in the video can easily lead to misalignments between reused and recomputed features, which hampers performance. Moreover, relying on feature reuse to improve prediction consistency is brittle; an erroneous alignment of the features can easily lead to unstable predictions. Therefore, the keyframe paradigm exhibits a dilemma between stability and performance. We address this efficiency and stability challenge using a novel yet simple Temporal Feature Correlation (TFC) module. It uses the cosine similarity between two frames' low-level features to inform the semantic label's consistency across frames. Specifically, we selectively reuse label-consistent features across frames through linear interpolation and update others through sparse multi-scale deformable attention. As a result, we no longer directly reuse features to improve stability and thus effectively solve feature misalignment. This work provides a significant step towards efficient and stable video semantic segmentation. On the VSPW dataset, our method significantly improves the prediction consistency of image-based methods while being as fast and accurate.

1 Introduction

Video semantic segmentation (VSS) (Miao et al. 2021) is critical in enabling machines to understand dynamic visual scenes and plays an important role in various applications, such as AR/VR, live video, self-driving, and robotics. The explosion of deep learning techniques brought significant improvements to VSS. Nevertheless, current methods often use a per-frame segmentation network for VSS (Liu et al. 2020). As a result, predictions across frames are unstable/inconsistent (Miao et al. 2021), making image-based methods impractical for real-world video applications. However, incorporating the temporal information is expensive (Sun et al. 2022a,b), leading to the emerging focus on efficient video semantic segmentation (Zhu et al. 2017; Shelhamer et al. 2016).

While stability in VSS has been under-explored (Miao et al. 2021), prevalent work (Zhu et al. 2017; Shelhamer et al. 2016; Hu et al. 2023) in efficient video semantic segmentation uses the keyframe paradigm: the non-keyframes only need to compute their low-level features and reuse high-level features computed at selected keyframes. The reused features stabilize the predictions across frames, thereby improving video consistency. However, dynamic scenes in the video can easily cause misalignments between the reused and recomputed features in the non-keyframe feature pyramid (Jain, Wang, and Gonzalez 2019; Hu et al. 2020). Therefore, the keyframe paradigm hurts the performance of existing image-based methods (Jain, Wang, and Gonzalez 2019). Furthermore, feature reuse implicitly improves stability but is sensitive to incorrectly aligned features. Consequently, the keyframe paradigm suffers from a dilemma between stability and performance. Prior work addresses the performance drop by aligning the reused features with flow-based warping (Jain, Wang, and Gonzalez 2019). Yet, optical flow is prohibitively expensive, and erroneous warping leads to unstable predictions.

We address the efficiency and stability challenge using a novel yet simple **T**emporal **F**eature **C**orrelation (TFC) module. TFC uses the cosine similarity between two frames' low-level features to inform the semantic label's consistency across frames. As shown in this paper (Fig. 3), its prediction of the between-frame semantic label consistency is highly accurate. Specifically, we selectively reuse label-consistent features across frames while updating others to account for the feature misalignment.

In this paper, we make two main contributions. First, we use the cosine similarity as the weight of the linear interpolation between features of two frames before the class prediction head. If the cosine similarity is high, it would as-

*These authors contributed equally.

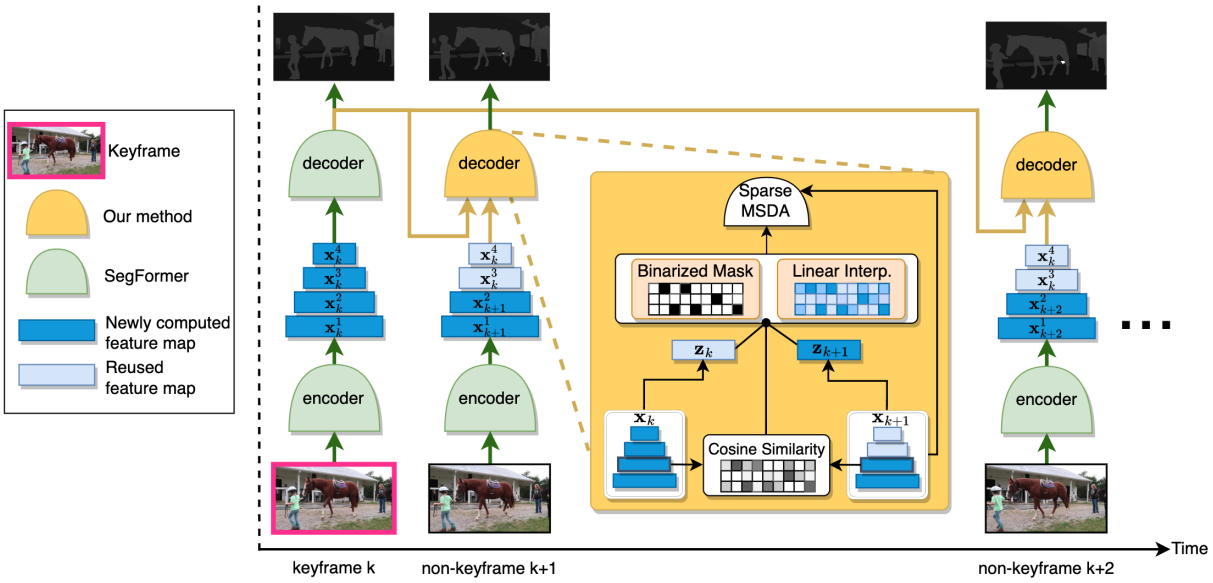

Figure 1: Outline of our method. At keyframes, the computation is kept the same as SegFormer (Xie et al. 2021). At non-keyframes, we propose two modules based on the cosine similarity. The yellow arrow going from one decoder to the next decoder represents $\mathbf{z}_k$ needed for linear interpolation at the next decoder. We only drew the diagram for one frame interval to keep the figure clean and easy to read. The above structure is repeated for every frame interval.

sign more weight to the keyframe features, and the prediction head would be biased toward predicting the same labels as the keyframe frame. Notably, linear interpolation requires no learnable parameters and only incurs a marginal computational cost. To further improve the prediction consistency, we add a binary cross-entropy loss based on focal loss (Lin et al. 2020) to guide the cosine similarity. We then obtain binary labels by comparing the semantic masks of two frames.

Second, we propose a *sparse* multi-scale deformable attention (MSDA) to efficiently solve the misalignment between reused and recomputed features. MSDA (Zhu et al. 2021) is a multi-scale feature fusion module that aggregates feature points at arbitrary locations and levels of the feature pyramid. It considers the feature misalignment and the semantic gap between features at different levels. Since the linear interpolation reuses feature points with high cosine similarity, we reduce the computation of MSDA by applying it to feature points with low cosine similarity. In VSPW (Miao et al. 2021), points with cosine similarity lower than 0.8 only account for about 10% of the feature points, which supports our sparse update strategy.

While prior work either focuses on stability (Sun et al. 2022a,b) or efficiency (Zhu et al. 2017), our work provides a significant step towards **efficient and stable** video semantic segmentation. Our method shown in Fig. 1 is novel yet simple. It solves for efficiency and stability with the cosine similarity between low-level features of two frames to inform the semantic label's consistency across frames. On this basis, we can selectively reuse label-consistent features across frames while updating others with MSDA to account for the feature misalignment. Our proposed method opens new considerations for the keyframe framework as a strong paradigm for efficient and stable VSS. On the VSPW (Miao et al. 2021) dataset, we significantly improve the prediction consistency of image-based methods while being as fast and accurate.

We organize this paper as follows. First, we summarize prior works in Sec. 2. Sec. 3 introduces the proposed linear interpolation and sparse MSDA. Experimental results in Sec. 4 illustrate our key design choices. Finally, Sec. 5 provides discussions.

2 Related Work

VSS Datasets & Prediction Consistency. Video semantic segmentation requires temporally dense annotations to study prediction consistency across frames. Yet, popular VSS datasets (Brostow, Fauqueur, and Cipolla 2009; Brostow et al. 2008; Cordts et al. 2016; Kim, Yim, and Kim 2018) are all sparsely-annotated or small-scale. For instance, Cityscapes (Cordts et al. 2016) and CamVid (Brostow, Fauqueur, and Cipolla 2009; Brostow et al. 2008) only annotate a single frame per video sequence. The Highway Driving dataset (Kim, Yim, and Kim 2018) is densely-annotated but small-scale; it contains a total of 1,200 annotated frames. As a result, much of prior work focuses on image semantic segmentation (Chen et al. 2017, 2018a,b; Fan et al. 2021; Li et al. 2019; Lin et al. 2017a; Liu et al. 2020; Long, Shelhamer, and Darrell 2015; Mehta et al. 2018; Nirkin, Wolf, and Hassner 2021; Orsic et al. 2019; Wang et al. 2020; Wu, Shen, and van den Hengel 2016; Xie et al. 2021; Yu et al. 2021, 2018; Zhao et al. 2017). The lack of large-scale datasets with temporally dense annotations has bottlenecked the study of prediction consistency in videos. Fortunately, VSPW (Miao et al. 2021) resolves this issue and provides dense annotations at 15 FPS for 3,536 videos. Therefore,

VSPW is so far the only viable dataset for studying stability and efficiency in VSS.

Flow-Guided Warping. Prior work focuses on warping the high-level features (e.g., optical flow (Jain, Wang, and Gonzalez 2019; Lee, Chen, and Peng 2021; Li, Shi, and Lin 2018; Xu et al. 2018; Zhu et al. 2017)) to mitigate the misalignment between the reused and recomputed features in the feature pyramid. This approach has two significant drawbacks in practice. First, flow-based warping struggles to alleviate the performance drop because it is inherently sensitive to occlusions. Second, previous methods improve prediction consistency through feature reuse, which assumes that the reused features remain the same. However, any erroneous warping would break that assumption and thus affect prediction consistency. MSDA (Zhu et al. 2021) has the following advantages over flow-guided warping. First and foremost, it considers the semantic gap in the feature pyramid due to the difference in the receptive field of each level. As a result, it provides a strong multi-scale feature fusion. Next, MSDA is less error-prone than flow-guided warping because it is not sensitive to occlusions. Instead of aggregating at the predicted flow, it aggregates feature points at arbitrary locations and levels of the feature pyramid. Nevertheless, MSDA is just as expensive as optical flow. Our work reduces computation by proposing a *sparse* MSDA.

Linear Interpolation between frames. While some work in VSS uses a linear interpolation between adjacent frames (Gadde, Jampani, and Gehler 2017; Jain and Gonzalez 2018; Mahasseni, Todorovic, and Fern 2017), they do not use it to improve the prediction consistency. NetWarp (Gadde, Jampani, and Gehler 2017) interpolates previously warped features with current features. Yet, NetWarp is not based on the keyframe paradigm and thus hampers its image-based counterpart with an expensive flow module. Jain *et al.* (Jain and Gonzalez 2018) interpolates between the current frame features and the warped features from adjacent keyframes, but it is not an online method because it looks ahead to the next keyframe. Mahasseni *et al.* (Mahasseni, Todorovic, and Fern 2017) directly interpolates between the previous frames' predicted pixels to produce labels for the current frame. Note that this method uses learnable parameters for the interpolation. In contrast with prior work, our linear interpolation does not rely on any learnable parameters, but uses the already computed low-level features. Based on the keyframe paradigm, we explicitly use our linear interpolation to improve prediction consistency across frames.

Multi-scale Feature Fusion. The difference in the receptive field of each level of a feature pyramid has the following effect. High-level features contain rich semantic information but lack fine-grained spatial information. Low-level features contain rich fine-grained spatial information but lack semantic information. Furthermore, Zhang *et al.* (Zhang et al. 2018) observed increased image segmentation performance by reducing the semantic gap. Although there exist many different flavors of multi-scale feature fusion (Lin et al. 2017b; Liu et al. 2018; Ghiasi, Lin, and Le 2019; Tan, Pang, and Le 2020; Zhang et al. 2018; Zhu et al. 2021; Roh et al. 2022), none of them use it to mitigate feature misalignment in keyframe-based methods.

3 Method

Our method uses a simple encoder-decoder architecture that predicts a semantic mask given an image. Akin to previous keyframe-based methods, our method follows SegFormer's computation (Xie et al. 2021) at keyframes and reuses the computed high-level features for non-keyframes. At non-keyframes, we use TFC to improve the prediction consistency across frames and to efficiently mitigate feature misalignment.

For clarity of explanation, we explain the computation for keyframes and non-keyframes separately. At non-keyframes, we address both efficiency and stability using the cosine similarity between low-level features of two frames, which is highly informative of the semantic label's consistency across frames. Based on the cosine similarity, we derive two operations: (1) a linear interpolation and (2) a *sparse* MSDA. The linear interpolation selectively reuses label-consistent features across frames while *sparse* MSDA updates others to account for the feature misalignment. Note that the parameters are shared across all frames except for the two proposed modules, which are only used at non-keyframes. We illustrate our method in Figs. 1 & 2.

3.1 Keyframe

As mentioned earlier, the computation at keyframes follows exactly that of SegFormer (Xie et al. 2021). SegFormer is an encoder-decoder network with a hierarchical Transformer encoder and a lightweight MLP decoder. Let k be the time step of the keyframe. Then, given an image I_k, it predicts semantic mask $y_k \in \mathbb{R}^{C\times H\times W}$ as follows.

First, the encoder produces feature maps at four levels with resolutions $\{1/4; 1/8; 1/16; 1/32\}$ of the original image. In formulae,

$$(\mathbf{x}_k^1, \mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_k^4) = \text{encoder}(I_k). \tag{1}$$

Next, we describe the decoder. A feature fusion step combines these feature maps into $\mathbf{x}_k = (\mathbf{x}_k^1, \mathbf{x}_k^2, \mathbf{x}_k^3, \mathbf{x}_k^4)$ by projecting and concatenating these features after up-sampling them at $1/4$ resolution. In formulae,

$$\mathbf{z}_k = \text{FC}(\text{concat}(\text{resize}(\text{FC}(\mathbf{x}_k)))), \tag{2}$$

Finally, a class prediction head predicts semantic classes at resolution $1/4$, and we obtain predictions at the original image resolutions by upsampling the semantic class predictions. Given $\mathbf{z}_k \in \mathbb{R}^{D\times H/4\times W/4}$, it yields

$$y_k = \text{resize}(\text{FC}(\mathbf{z}_k)). \tag{3}$$

3.2 Non-keyframe

Let $t > k$ be the time step of the non-keyframe. Then, at non-keyframe, the encoder only extracts the low-level features $(\mathbf{x}_t^1, \mathbf{x}_t^2)$ and reuses the high-level features $(\mathbf{x}_k^3, \mathbf{x}_k^4)$ from the keyframe to form the feature pyramid

$$\mathbf{x}_t = (\mathbf{x}_t^1, \mathbf{x}_t^2, \mathbf{x}_k^3, \mathbf{x}_k^4). \tag{4}$$

Two issues arise at non-keyframes. First, video scenes may be dynamic, thus $(\mathbf{x}_t^1, \mathbf{x}_t^2)$ and $(\mathbf{x}_k^3, \mathbf{x}_k^4)$ become inconsistent in Eq. (4). Second, relying only on feature reuse

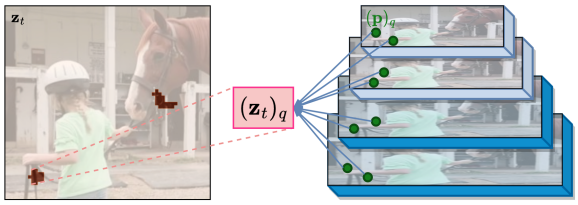

Figure 2: Sparse MSDA Diagram. The brown patches on the image represent the points i that satisfy $(\mathbf{w}_t)_i < 0.8$. We apply MSDA to only these specific points to reduce the computation cost while alleviating temporal misalignment.

to improve the prediction consistency across frames is brittle. It marginally improves the stability and it assumes that $(\mathbf{x}_k^3, \mathbf{x}_k^4)$ remain the same. Consequently, erroneous feature alignments lead to unstable predictions.

Define the binary operator on feature maps $\langle \cdot, \cdot \rangle$ and the operator $(\cdot)_i$ indexing a point $i = (h, w)$ on a feature map. We also denote the projected and resized version of $\mathbf{x}^l$ in Eq. (2) as $\hat{\mathbf{x}}^l$. The binary operator computes the cosine similarity between each feature point on $\hat{\mathbf{x}}_k^l \in \mathbb{R}^{D \times H/4 \times W/4}$ and its corresponding feature point at time step t. In formulae,

$$\mathbf{w}_t = \langle \hat{\mathbf{x}}_k^l, \hat{\mathbf{x}}_t^l \rangle \in \mathbb{R}^{1 \times H/4 \times W/4}, \tag{5}$$

where the cosine similarity $(\langle \cdot, \cdot \rangle)_i$ is defined as follows

$$(\mathbf{w}_t)_i = \frac{(\hat{\mathbf{x}}_k^l)_i}{\|(\hat{\mathbf{x}}_k^l)_i\|_2} \cdot \frac{(\hat{\mathbf{x}}_t^l)_i}{\|(\hat{\mathbf{x}}_t^l)_i\|_2}. \tag{6}$$

In Sec. 4.3, we empirically find that $l = 2$ provides better results than $l = 1$. Eq. (6) is highly informative of the semantic label's consistency across frames. Intuitively, for $(\hat{\mathbf{x}}_k^l)_i$ and $(\hat{\mathbf{x}}_t^l)_i$ with consistent semantic label, we expect $(\mathbf{w}_t)_i$ to be close to 1.

Linear Interpolation. We leverage linear interpolation to maintain prediction consistency across frames by weighing between keyframe feature $\mathbf{z}_k$ and non-keyframe feature $\mathbf{z}_t$. Define the Hadamard product on matrices with $\odot$. We use the linear interpolation based on the cosine similarity as follows

$$\mathbf{z}_t' = \hat{\mathbf{w}}_t \odot \mathbf{z}_k + (\mathbf{1} - \hat{\mathbf{w}}_t) \odot \mathbf{z}_t, \tag{7}$$

where $\hat{\mathbf{w}}_t = \text{clamp}(\hat{\mathbf{w}}_t, \text{min} = 0, \text{max} = 1)$. Intuitively, the extent to which $(\mathbf{z}_k)_i$ and $(\mathbf{z}_t)_i$ weigh depends on the extent that low-level features have changed at i since the keyframe. If the cosine similarity is high, then $(\mathbf{z}_t)_i$ will be biased towards $(\mathbf{z}_k)_i$. Consequently, semantic prediction at $(\mathbf{z}_k)_i$ is kept and prediction consistency is improved.

MSDA. Next, we describe how MSDA (Zhu et al. 2021) performs multi-scale feature fusion (readers familiar with MSDA can skip this part). Given $\mathbf{z}_t$ from Eq. (2) and sparse locations $\mathbf{p}_t$ on the feature pyramid $\mathbf{x}_t$, MSDA yields

$$\text{MSDA} : (\mathbf{z}_t, \mathbf{x}_t, \mathbf{p}_t) \rightarrow \mathbf{z}_t. \tag{8}$$

The resulting feature points $\mathbf{z}_t$ contain semantic information from multiple levels and locations (given by $\mathbf{p}_t$) on $\mathbf{x}_t$. The total number of sampling points is defined by the number of levels l, heads m, and keys k. Let q be a point on $\mathbf{z}_t$, denoted as $(\mathbf{z}_t)_q$. Let $\mathbf{v}_t$ be the values obtained from sampling points $(\mathbf{p}_t)_{qmlk}$ on the multi-scale feature map $\mathbf{x}_t$. Then, MSDA for a single point is

$$(\mathbf{z}_t)_q = \sum_m \mathbf{W}_m \left[\sum_l \sum_k (A_t)_{qmlk} \cdot \mathbf{v}_t((\mathbf{p}_t)_{qmlk}) \right], \tag{9}$$

where

$$\mathbf{v}_t((\mathbf{p}_t)_{qmlk}) = \mathbf{W}'_m \mathbf{x}_t^l((\mathbf{p}_t)_{qmlk}). \tag{10}$$

We obtain sampling points $(\mathbf{p}_t)_{qmlk}$ and attention weights $(A_t)_{qmlk}$ with a linear layer on top of $(\mathbf{z}_t)_q$. Moreover, the attention weights are normalized such that $\sum_m \sum_l \sum_k (A_t)_{qmlk} = 1$. The strength of MSDA comes from the fact that $(\mathbf{p}_t)_{qmlk}$ and $(A_t)_{qmlk}$ are data-dependent. In this work, we advocate for MSDA as an efficient way to mitigate feature misalignment.

Sparse MSDA. To sparsify MSDA, we propose applying MSDA only on feature points of $\mathbf{z}_t$ with low cosine similarity. In formulae,

$$\text{sparseMSDA} : (\mathbf{z}_t, \mathbf{x}_t, \mathbf{p}_t, \mathbf{w}_t) \rightarrow \mathbf{z}_t. \tag{11}$$

Sparse MSDA only applies MSDA to feature points $(\mathbf{z}_t)_q$ for which $(\mathbf{w}_t)_q < 0.8$. In VSPW (Miao et al. 2021), these points account for about 10% of points, which makes *sparse* MSDA much cheaper than MSDA (Zhu et al. 2021). More specifically, the complexity of MSDA w.r.t the number points N_q is linear, i.e., $O(N_q D^2 + 5N_q KD + 3N_q DK)$, where K is the number of sampling points $|(\mathbf{p})_q|$ and D is the feature dimension. As a comparison, in MSDA, $N_q = HW/4^2$ while in *sparse* MSDA, $N_q \ll HW/4^2$.

3.3 Implementation Details

This section provides implementation details on the keyframe interval and training loss. Methods focusing on selecting keyframes (Shelhamer et al. 2016) are orthogonal to our work. Hence, we select every 4th frame as a keyframe for simplicity. In addition, the linear interpolation and cosine similarity are based on the keyframe $\mathbf{x}_k$ and not the previous frames $\mathbf{x}_{t-1}$. This prevents errors from accumulating. For training, we do not tune the keyframe interval as increasing it makes training prohibitively expensive. We perform all our experiments using 16 GeForce GTX 1080, where a keyframe interval of 4 fits well for all models (Xie et al. 2021).

We can trivially obtain a binary mask for supervising the (clamped) cosine similarity by comparing semantic labels between two frames. Furthermore, we find that the binary mask is imbalanced since the VSPW (Miao et al. 2021) dataset contains temporally dense annotations at 15 FPS. To account for the class imbalance, we implement the binary cross entropy based on the focal loss (Lin et al. 2020) with $\gamma = 1$.

Define L_{seg} as the cross entropy loss for supervising the segmentation mask. Similarly, define L_{bce} as the binary cross entropy based on focal loss (Lin et al. 2020) for supervising the (clamped) cosine similarity. We train the whole model in an end-to-end manner to minimize

$$L = L_{\text{seg}} + L_{\text{bce}}, \tag{12}$$

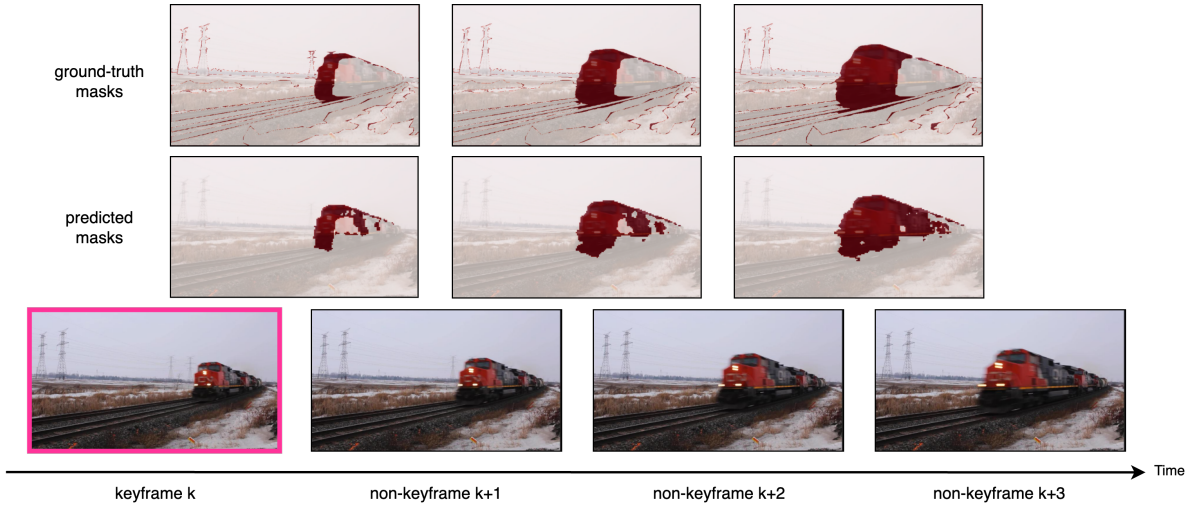

Figure 3: Predicted masks on the validation set. Based on the MiT-B5 (Xie et al. 2021) backbone, we show the predicted binary mask for points with cosine similarity lower than 0.8. The binary masks are computed between the keyframe and the non-keyframes. Because videos in VSPW (Miao et al. 2021) are very densely annotated, changes between frames are less visually apparent than in other datasets. We set the spacing between frames to be higher than usual for better visualization.

where L_{seg} takes the average of the cross entropy loss on the keyframe and the non-keyframes.

4 Experiments

Dataset. As summarized in Sec. 2, VSPW (Miao et al. 2021) is so far the only dataset that provides temporally dense annotations (15 FPS) for VSS. Since the study of prediction consistency across frames requires a temporally dense annotation, we perform all experiments on VSPW. In addition, VSPW is the largest VSS benchmark, with 198,244 training frames and 24,502 validation frames.

Training and Inference. We train all models on 16 Nvidia GeForce GTX 1080 with a batch size of 1 on each GPU (Contributors 2020). Training takes approximately $1-2$ days depending on the model size. Our backbone uses the pretrained SegFormer on ImageNet (Deng et al. 2009). Accordingly, we keep all hyper-parameters as in SegFormer (Xie et al. 2021). We use random resizing, flipping, cropping, and photometric distortions during training. The cropping size is set to 480×480. In addition, we adopt AdamW with a "poly" learning schedule with an initial learning rate of $6e^{-5}$. At inference, the keyframe interval is kept the same as for training unless specified, and images are resized to 480×853.

Evaluation. Following VSPW (Miao et al. 2021), we use mean IoU (mIoU), weighted IoU (wIoU), and mean Video Consistency (mVC). In particular, IoU measures the accuracy of the semantic mask, and video consistency measures the prediction consistency across frames. For a video clip $\{I^{(c)}\}_{c=1}^{C}$ of length C with ground-truth masks $\{y^{(c)}\}_{c=1}^{C}$ and predicted masks $\{\hat{y}^{(c)}\}_{c=1}^{C}$, video consistency $\text{VC}_C \in [0,1]$ is computed as follows,

$$\text{VC}_C = \frac{(y^{(1)} \cap ... \cap y^{(C)}) \cap (\hat{y}^{(1)} \cap ... \cap \hat{y}^{(C)})}{(y^{(1)} \cap ... \cap y^{(C)})} \quad (13)$$

Intuitively, $(y^{(1)} \cap ... \cap y^{(C)})$ computes the common area along the video for ground-truth masks and similarly for predicted masks. Then, we wish the common area of the ground-truth and predicted mask to be as high as possible.

4.1 Baselines

This section summarizes the baseline methods in Tab. 1. To our knowledge, there are no keyframe-based methods on the VSPW dataset. We describe next a set of reimplemented baselines that focus on efficient VSS.

CFFM & MRCFA. These methods (Sun et al. 2022a,b) do not belong to efficient VSS methods. Therefore, a direct comparison with these methods is unfair. In particular, they do not leverage the temporal prior to reduce the computation of the image-based methods. Orthogonal to the keyframe paradigm, they focus on improving a per-frame segmentation model using extra computation to extract temporal information at multiple frames. In contrast, keyframe-based methods reduce the computation at non-keyframes by reusing features computed at keyframes. As a result, keyframe-based methods process videos more efficiently than their image-based counterpart. We add these non-efficient VSS methods for reference.

Feature Reuse. Our Feature Reuse (Shelhamer et al. 2016) baseline implements the vanilla version of the keyframe paradigm, which corresponds to our method without linear interpolation and sparse MSDA.

TDNet. We reimplement TDNet (Hu et al. 2020) and remove the knowledge distillation component for a fair comparison. We note that in TDNet, removing knowledge distil-

Methods	Backbone	Type	mIoU ↑	Stability		Efficiency	
				mVC_8 ↑	mVC_{16} ↑	Params (M) ↓	FPS (f/s) ↑
SegFormer	MiT-B0	Image	32.9	82.7	77.3	3.8	37.5
Segformer + TFC	MiT-B0	E & S	32.9	85.3 (+2.6)	80.0 (+2.7)	4.5 (+0.7)	38.2 (+0.7)
SegFormer + Feature Reuse	MiT-B0	E	31.5	84.3	78.7	3.8	39.9
SegFormer + TDNet	MiT-B0	E	31.74	83.4	78.8	5.8	14.1
SegFormer + MRCFA	MiT-B0	S	35.2	88.0	83.2	5.2	27.3
SegFormer + CFFM	MiT-B0	S	35.4	87.7	82.9	4.7	30.3
SegFormer	MiT-B1	Image	36.5	84.7	79.9	13.8	34.9
Segformer + TFC	MiT-B1	E & S	36.7	86.8 (+2.1)	81.5 (+1.5)	14.5 (+0.7)	37.1 (+2.2)
SegFormer + MRCFA	MiT-B1	S	38.9	88.8	84.4	16.2	22.1
SegFormer + CFFM	MiT-B1	S	38.5	88.6	84.1	15.5	21.5
SegFormer	MiT-B5	Image	48.2	87.8	83.7	82.1	26.2
SegFormer + TFC	MiT-B5	E & S	48.3	90.5 (+2.7)	86.7 (+3.0)	82.8 (+0.7)	37.8 (+11.6)
SegFormer + CFFM	MiT-B5	S	49.3	90.8	87.1	85.5	20.0
SegFormer + MRCFA	MiT-B5	S	49.9	90.9	87.4	84.5	23.0

Table 1: Main results on VSPW (Miao et al. 2021). We show in parenthesis the difference with the image-based counterpart. E stands for Efficiency and S for Stability. In particular, prior works focus on either Efficiency (Shelhamer et al. 2016) or Stability (Sun et al. 2022a,b). We report the average FPS on a single key-frame interval.

lation results in a performance drop of 1.2 mIoU. In particular, we are interested in comparing our linear interpolation with their attention propagation module.

4.2 Results

Tab. 1 reports the main results on the VSPW evaluation set (Miao et al. 2021). Previous state-of-the-art methods increase stability at the cost of inference speed (Sun et al. 2022b,a). In contrast, our method boosts both stability and efficiency. Furthermore, we find that relying on feature reuse increases the VC by 1.6 and 1.4 for mVC_8 and mVC_{16}, respectively. We also make gains in VC compared to the backbone. On the MiT-B0 backbone, mVC_8 and mVC_{16} increases by 2.6 and 2.7, respectively. Compared to TDNet (Hu et al. 2020), our linear interpolation significantly raises the VC without requiring any trainable parameter.

Compared to the image-based counterpart, SegFormer (Xie et al. 2021), our method significantly improves stability while being as fast and accurate. For reference, at non-keyframe, our method's GFLOPs with B0 vs MiT-B0 at non-keyframes is: 6.4 vs 6.8. Similarly with B5 vs MiT-B5 at non-keyframes is: 53.2 vs 95.7. More importantly, enhancing stability without hampering speed and accuracy is a significant milestone toward efficient and stable VSS.

4.3 Ablation

We perform ablation studies to show the importance of some key design choices, including (1) the computation of cosine similarity, (2) the sparsity of MSDA, (3) the weighting in linear interpolation, (4) the cosine similarity supervision, and (5) the keyframe interval.

Cosine Similarity. The goal of the cosine similarity is to inform the semantic label's consistency across frames. In particular, it is the main contribution of our method, which uses it to derive a linear interpolation and sparse MSDA. As described in Sec. 3.2, we compute the cosine similarity using $\hat{\mathbf{x}}_k^2$ and $\hat{\mathbf{x}}_t^2$. This choice is not arbitrary. In Tab. 2, we

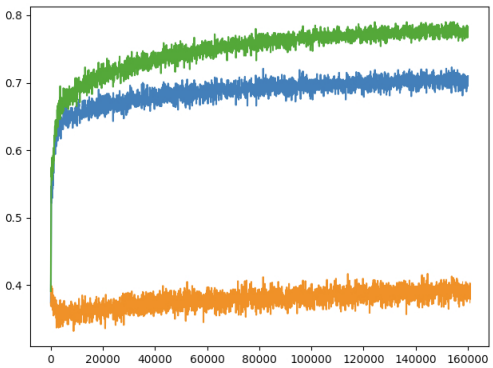

Figure 4: Accuracy of the binary mask. We display the binary mask accuracy for MiT-B0 (blue and orange) & MiT-B5 (green) during training. For MiT-B0, we plot the accuracy of the clamped cosine similarity with (blue) and without (orange) the binary cross entropy loss based on the focal loss. Supervising the cosine similarity is essential for learning an accurate binary mask.

empirically show that using $\hat{\mathbf{x}}_k^1$ and $\hat{\mathbf{x}}_t^1$ gives weaker results in both prediction accuracy and prediction consistency. We briefly described some variants that we have tried that did not work. First, we observed that the model does not converge when we use $\mathbf{z}_k$ and $\mathbf{z}_t$, probably because the reused high-level features introduce significant noise. Second, we found that using a neural network to directly output the binary mask is non-trivial. For both cases, the model could not even overfit to a single batch. In contrast, the cosine similarity provides a simple and efficient way to obtain the semantic label's consistency across frames.

Sparse MSDA. Tab. 3 compares the inference speed at different MSDA sparsity levels against the inference speed of computing high-level Segformer features (Xie et al.

	Methods	mIoU	mVC_8	mVC_{16}
Linear Interp.	Feature reuse	31.5	84.3	78.7
	+ Linear interp.	**32.0**	85.0	79.3
	+ Hard interp.	32.0	83.7	78.0
	+ Fixed interp.	31.0	**85.9**	**80.5**
Sparse MSDA	Feature reuse	31.5	**84.3**	78.7
	+ Sparse MSDA (0.8)	32.7	84.2	78.7
	+ Sparse MSDA (1.0)	**33.8**	79.0	74.3
	+ Sparse MSDA (0.5)	31.9	**84.3**	**79.0**
	+ Flow-guided Warping	32.7	81.0	75.0
Cosine Similarity	$\hat{\mathbf{x}}^2$	**32.9**	**85.3**	**80.0**
	$\hat{\mathbf{x}}^1$	31.6	84.5	79.0
Focal Loss	With Focal Loss	**32.9**	**85.3**	**80.0**
	Without Focal Loss	32.1	83.1	77.7
Frame Interval	Interval 4	**32.9**	85.3	80.0
	Interval 5	32.9	85.8	80.1
	Interval 6	32.8	85.8	80.4
	Interval 7	32.7	86.0	80.7
	Interval 8	32.6	86.4	80.9
	Interval 9	32.4	**86.6**	**81.1**

Table 2: Ablation results. We highlight the setup used in our method in gray. For Sparse MSDA, we show in parenthesis the threshold. Our method is the only one that is efficient and stable.

	Methods	FPS f/s
SegFormer (Xie et al. 2021)	MiT-B0	342.3
	MiT-B1	319.1
	MiT-B5	31.7
Sparse MSDA	MiT-B0 (20%)	522.9
	MiT-B0 (40%)	455.5
	MiT-B0 (100%)	302.6

Table 3: sparse MSDA inference speed. We compare the inference speed of sparse MSDA against the last two layers of the backbone on an A100 GPU.

2021). We find that the computation at non-keyframes is lower than at keyframes. In Tab. 2, we compare MSDA for different thresholds on $\mathbf{w}_t$. The results show that sparsity is important not only for improving efficiency but also for preserving stability. When all points are selected (threshold 1.0), the performance increases while stability decreases. This is because the performance at non-keyframes becomes greater than at keyframes. Flow-guided warping (Teed and Deng 2020) also follows a similar trend. The erroneous warping breaks the assumption of feature reuse methods, leading to unstable predictions.

Linear Interpolation. We compare two other linear interpolation variants to show the effectiveness of our proposed linear interpolation. The first variant binarizes the weight by applying a threshold (the same threshold as in sparse MSDA). The linear interpolation either reuses the features completely or forgets them completely. Tab. 4.3 shows that binarizing the weight significantly hurts the stability. The second variant fixes the weight of the linear interpolation. Since approximately 10% of the feature points change between two frames, we set the weight to be 0.1. As shown in Tab. 4.3, the results are much more stable, but it leads to a considerable performance drop.

Focal Loss. Intuitively, we would expect the network to naturally learn an accurate binary mask. However, the results in Fig. 4 and Tab. 2 empirically demonstrate that supervision with the binary cross entropy loss based on focal foss is essential. In addition, in Fig. 4, we observe that better-performing models (MiT-B0 vs MiT-B5) tend to learn a better binary mask. We also show some qualitative binary masks in Fig. 3.

Keyframe Interval. Tab. 2 shows that our method can easily generalize to longer sequences when trained on a keyframe interval 4. In particular, we can increase the VC by 0.7 by just adding 3 frames at inference.

5 Discussion and Conclusion

Previous work in video semantic segmentation either focuses on efficiency or stability. In contrast, our work proposes tackling efficiency and stability together. While the two problems are seemingly unrelated, we find that they can be solved at the same time. In particular, we reveal that the cosine similarity between two frames' low-level features is highly informative of the semantic label's consistency across frames. Leveraging this finding, we propose a linear interpolation to selectively reuse features based on the cosine similarity. Then, we propose a sparse MSDA to efficiently update feature points. As a result, we can effectively preserve efficiency while enhancing stability. We note here that a potential limitation of our method is the propagation of error from keyframes to non-keyframes, which we leave for future work. We believe future work can use the proposed cosine similarity for scheduling keyframes.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (Project Number 62332019).

References

Brostow, G. J.; Fauqueur, J.; and Cipolla, R. 2009. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.*, 30(2): 88–97.

Brostow, G. J.; Shotton, J.; Fauqueur, J.; and Cipolla, R. 2008. Segmentation and Recognition Using Structure from Motion Point Clouds. In Forsyth, D.; Torr, P.; and Zisserman, A., eds., *Computer Vision – ECCV 2008*, 44–57. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-88682-2.

Chen, L.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR*, abs/1706.05587.

Chen, L.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018a. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, 833–851. Springer.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018b. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 834–848.

Contributors, M. 2020. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Fan, M.; Lai, S.; Huang, J.; Wei, X.; Chai, Z.; Luo, J.; and Wei, X. 2021. Rethinking BiSeNet for Real-Time Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 9716–9725. Computer Vision Foundation / IEEE.

Gadde, R.; Jampani, V.; and Gehler, P. V. 2017. Semantic video cnns through representation warping. In *Proceedings of the IEEE International Conference on Computer Vision*, 4453–4462.

Ghiasi, G.; Lin, T.-Y.; and Le, Q. V. 2019. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, P.; Caba, F.; Wang, O.; Lin, Z.; Sclaroff, S.; and Perazzi, F. 2020. Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8818–8827.

Hu, Y.; He, Y.; Li, Y.; Li, J.; Han, Y.; Wen, J.; and Liu, Y.-J. 2023. Efficient Semantic Segmentation by Altering Resolutions for Compressed Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22627–22637.

Jain, S.; and Gonzalez, J. E. 2018. Fast Semantic Segmentation on Video Using Block Motion-Based Feature Interpolation. In Leal-Taixé, L.; and Roth, S., eds., *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, volume 11132 of *Lecture Notes in Computer Science*, 3–6. Springer.

Jain, S.; Wang, X.; and Gonzalez, J. E. 2019. Accel: A corrective fusion network for efficient semantic segmentation on video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8866–8875.

Kim, B.; Yim, J.; and Kim, J. 2018. Highway Driving Dataset for Semantic Video Segmentation. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, 140. BMVA Press.

Lee, S.-P.; Chen, S.-C.; and Peng, W.-H. 2021. GSVNET: Guided Spatially-Varying Convolution for Fast Semantic Segmentation on Video. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.

Li, H.; Xiong, P.; Fan, H.; and Sun, J. 2019. DFANet: Deep Feature Aggregation for Real-Time Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 9522–9531. Computer Vision Foundation / IEEE.

Li, Y.; Shi, J.; and Lin, D. 2018. Low-latency video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5997–6005.

Lin, G.; Milan, A.; Shen, C.; and Reid, I. 2017a. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2020. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2): 318–327.

Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017b. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; and Jia, J. 2018. Path Aggregation Network for Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y.; Shen, C.; Yu, C.; and Wang, J. 2020. Efficient semantic video segmentation with per-frame inference. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, 352–368. Springer.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mahasseni, B.; Todorovic, S.; and Fern, A. 2017. Budget-Aware Deep Semantic Video Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2077–2086. IEEE Computer Society.

Mehta, S.; Rastegari, M.; Caspi, A.; Shapiro, L. G.; and Hajishirzi, H. 2018. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, volume 11214 of *Lecture Notes in Computer Science*, 561–580. Springer.

Miao, J.; Wei, Y.; Wu, Y.; Liang, C.; Li, G.; and Yang, Y. 2021. VSPW: A Large-scale Dataset for Video Scene Parsing in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4133–4143.

Nirkin, Y.; Wolf, L.; and Hassner, T. 2021. HyperSeg: Patch-Wise Hypernetwork for Real-Time Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 4061–4070. Computer Vision Foundation / IEEE.

Orsic, M.; Kreso, I.; Bevandic, P.; and Segvic, S. 2019. In Defense of Pre-Trained ImageNet Architectures for Real-Time Semantic Segmentation of Road-Driving Images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 12607–12616. Computer Vision Foundation / IEEE.

Roh, B.; Shin, J.; Shin, W.; and Kim, S. 2022. Sparse DETR: Efficient End-to-End Object Detection with Learnable Sparsity. In *ICLR*.

Shelhamer, E.; Rakelly, K.; Hoffman, J.; and Darrell, T. 2016. Clockwork convnets for video semantic segmentation. In *European Conference on Computer Vision*, 852–868. Springer.

Sun, G.; Liu, Y.; Ding, H.; Probst, T.; and Van Gool, L. 2022a. Coarse-to-Fine Feature Mining for Video Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3126–3137.

Sun, G.; Liu, Y.; Tang, H.; Chhatkuli, A.; Zhang, L.; and Van Gool, L. 2022b. Mining Relations Among Cross-Frame Affinities for Video Semantic Segmentation. In *European Conference on Computer Vision*, 522–539. Springer.

Tan, M.; Pang, R.; and Le, Q. V. 2020. EfficientDet: Scalable and Efficient Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, 402–419. Springer.

Wang, L.; Li, D.; Zhu, Y.; Tian, L.; and Shan, Y. 2020. Dual Super-Resolution Learning for Semantic Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 3773–3782. Computer Vision Foundation / IEEE.

Wu, Z.; Shen, C.; and van den Hengel, A. 2016. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *CoRR*, abs/1611.10080.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34: 12077–12090.

Xu, Y.-S.; Fu, T.-J.; Yang, H.-K.; and Lee, C.-Y. 2018. Dynamic Video Segmentation Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; and Sang, N. 2021. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation. *Int. J. Comput. Vis.*, 129(11): 3051–3068.

Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; and Sang, N. 2018. BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, volume 11217 of *Lecture Notes in Computer Science*, 334–349. Springer.

Zhang, Z.; Zhang, X.; Peng, C.; Xue, X.; and Sun, J. 2018. Exfuse: Enhancing feature fusion for semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 269–284.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid Scene Parsing Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable {DETR}: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*.

Zhu, X.; Xiong, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2349–2358.