Efficient Semantic Segmentation by Altering Resolutions for Compressed Videos

Yubin Hu[1] Yuze He[1] Yanghao Li[1] Jisheng Li[1] Yuxing Han[2] Jiangtao Wen[3] Yong-Jin Liu[1]*
[1]BNRist, Department of Computer Science and Technology, Tsinghua University
[2]Shenzhen International Graduate School, Tsinghua University
[3]Eastern Institute for Advanced Study
{huyb20, hyz22, liyangha18}@mails.tsinghua.edu.cn, jas0n1ee@icloud.com,
yuxinghan@sz.tsinghua.edu.cn, jtwen@eias.ac.cn, liuyongjin@tsinghua.edu.cn

Abstract

Video semantic segmentation (VSS) is a computationally expensive task due to the per-frame prediction for videos of high frame rates. In recent work, compact models or adaptive network strategies have been proposed for efficient VSS. However, they did not consider a crucial factor that affects the computational cost from the input side: ***the input resolution****. In this paper, we propose an altering resolution framework called AR-Seg for compressed videos to achieve efficient VSS. AR-Seg aims to reduce the computational cost by using low resolution for non-keyframes. To prevent the performance degradation caused by downsampling, we design a Cross Resolution Feature Fusion (CReFF) module, and supervise it with a novel Feature Similarity Training (FST) strategy. Specifically, CReFF first makes use of motion vectors stored in a compressed video to warp features from high-resolution keyframes to low-resolution non-keyframes for better spatial alignment, and then selectively aggregates the warped features with local attention mechanism. Furthermore, the proposed FST supervises the aggregated features with high-resolution features through an explicit similarity loss and an implicit constraint from the shared decoding layer. Extensive experiments on CamVid and Cityscapes show that AR-Seg achieves state-of-the-art performance and is compatible with different segmentation backbones. On CamVid, AR-Seg saves 67% computational cost (measured in GFLOPs) with the PSPNet18 backbone while maintaining high segmentation accuracy. Code:* `https://github.com/THU-LYJ-Lab/AR-Seg`.

1. Introduction

Video semantic segmentation (VSS) aims to predict pixel-wise semantic labels for each frame in a video sequence. In contrast to a single image, a video sequence is a series of consecutive image frames recorded at a certain frame rate (usually 25fps or higher). Applying image-based segmentation methods [6, 25, 48, 52, 55] to a video frame by frame consumes considerable computational resources. To improve the efficiency of VSS, existing methods mainly focus on the design of network architectures. A class of methods proposes compact and efficient image-based architectures to reduce the computational overhead per-frame [22, 23, 28, 49, 51, 52, 54]. Another class of methods applies a deep model to keyframes and a shallow network for non-keyframes to avoid the repetitive computation [19, 24, 27, 34] for videos.

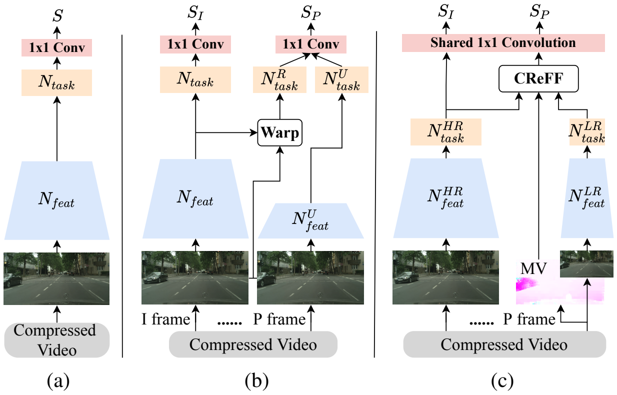


Figure 1. Comparison of different VSS methods: (a) per-frame framework, (b) Accel [19] that alters the depth of models, and (c) our AR-Seg. AR-Seg reduces the computational cost for non-keyframes by lowering the input resolution (depicted by narrow blocks), which is a dimension orthogonal to the depth of networks.

The work presented in this paper is based on an important observation: the above existing works ignored a crucial factor that affects the computational cost from the input side: **the input resolution**. For image-related tasks, the input resolution directly determines the amount of computation, e.g., the computational cost of 2D convolution is proportional to the product of image width and height. Once we

*Corresponding author.

downsample the input frame by 0.5×0.5, the computational overhead can be reduced by 75%. On the other hand, decreasing resolution often leads to worse segmentation accuracy due to the loss of information [43,54]. In this paper, we propose to prevent the accuracy degradation by using temporal correlation in the video. Since the contents of video frames are temporally correlated, the local features lacking in low-resolution (LR) frames can be inferred and enriched by finding correspondences in sparse high-resolution (HR) reference frames based on motion cues. In a compressed video, the motion vectors contain such motion cues and can be obtained along with the video frames from video decoding at almost no additional cost.

Motivated by the above observation, we propose an altering resolution framework for compressed videos, named AR-Seg, to achieve efficient VSS. As shown in Figure 1(c), AR-Seg uses an HR branch to process keyframes at high resolution and an LR branch to process non-keyframes at low resolution. In particular, to prevent performance drop caused by downsampling, we insert a novel Cross Resolution Feature Fusion (CReFF) module into the LR branch and supervise the training with a Feature Similarity Training (FST) strategy to enrich local details in the LR features. CReFF fuses the HR keyframe features into LR non-keyframe features in two steps: 1) Align the spatial structures of features from different frames by feature warping with motion vectors, which can be readily obtained from compressed videos at almost no additional cost; 2) Selectively aggregate the warped features (which may be noisy after warping) into LR features with the aid of local attention mechanism. Since local attention assigns different importance to each location in the neighborhood, it is an effective way to avoid misleading by noisy warped features.

Furthermore, our proposed FST strategy guides the learning of the CReFF aggregated features. FST consists of an *explicit* similarity loss (between the aggregated features and HR features inferred from non-keyframes) and an *implicit* constraint from the shared decoding layer across the HR and LR branches. Such a training strategy helps the LR branch to learn from features extracted from the HR branch, which is reliable and effective. Integrated with CReFF and FST, AR-Seg efficiently compensates for the accuracy loss of LR frames. Overall, AR-Seg significantly reduces the computational cost of VSS by altering input resolutions, while maintaining high segmentation accuracy.

To sum up, we make three contributions in this paper: **1)** We propose an efficient framework for compressed videos, named AR-Seg, that uses altering input resolution for VSS and significantly reduces the computational cost without losing segmentation accuracy. **2)** We design an efficient CReFF module to prevent the accuracy loss by aggregating HR keyframe features into LR non-keyframe features. **3)** We propose a novel FST strategy that supervises the LR branch to learn from the HR branch through both explicit and implicit constraints. Experiment results demonstrate the effectiveness of AR-Seg with different resolutions, backbones, and keyframe intervals. On both CamVid [3] and Cityscapes [9] datasets, compared to the constant-resolution baselines, AR-Seg reduces the computational cost by nearly 70% without compromising accuracy.

2. Related Works

As a fundamental task of scene understanding, semantic segmentation has been an active research area for many years [8, 13, 21, 41], which also attracts considerable attention in the study of deep neural networks, e.g., FCN [25], DeepLabs [4, 5] and PSPNet [55]. In order to obtain accurate results in real-time applications, several methods have been proposed to improve the efficiency of semantic segmentation, which we summarize as follows.

Efficient Image Segmentation Methods. Many compact architectures have been proposed for efficient image segmentation. DFANet [22] adopted a lightweight backbone to reduce computational cost and designed cross-level aggregation for feature refinement. DFNet [23] utilized a partial order pruning algorithm to search segmentation models for a good trade-off between speed and accuracy. ICNet [54] used a cascade fusion module and transformed part of the computation from high-resolution to low-resolution. Wang et al. [43] designed super-resolution learning to improve image segmentation performance. BiSeNets [11,51,52] used two-stream paths for low-level details and high-level context information, respectively. ESPNet [28] used an efficient spatial pyramid to accelerate the convolution computation. These efficient backbone networks reduce the computational burden of single-image segmentation, and can be applied to temporal or spatial frameworks in VSS.

Temporally Correlated Video Segmentation. Another group of methods focus on utilizing temporal redundancy in videos. They proposed various mechanisms that propagate the deep features extracted from keyframes to reduce the computation for non-keyframes. Clockwork [36] directly reused the segmentation result from keyframes, while Mahasseni et al. [27] interpolated segmentation results in the neighborhood. Noticing the lack of information from non-keyframes, Li et al. [24] extracted shallow features from non-keyframes, and fused them into the propagated deep features by spatially variant convolution. To compensate for the spatial misalignment between video frames, Zhu et al. [56] and Xu et al. [50] warped the intermediate features from keyframes by optical flow to produce segmentation results for non-keyframes. Jain et al. [19] fused the shallow features of non-keyframe into the warped features, and decoded them into better results. With global attention mechanism, TD-Net [16] aggregated the features from different time stamps and replaced the deep model with several shallow models distributed across the timeline. All the above methods mainly reduced the depth of backbone networks, but neglected the factor of input resolution considered in this paper. Instead of processing the image frames as a whole, Verelst et al. [42] split the frame into blocks and chose to copy or process them by a policy network. This block-based method reduces computational overhead from the spatial dimension, but lacks global information on non-keyframes. Kim et al. [20] attempted to improve efficiency by reducing resolution. But they directly used the LR segmentation results, thus suffering from severe performance degradation. Compared to these methods, our proposed AR-Seg keeps the global information of LR non-keyframes, and enhances LR frames by selectively aggregating intermediate features from HR keyframes.

Compressed-Domain Video Analysis. Compressed video formats have been recently utilized in computer vision tasks. The motion vectors and residual maps are treated as additional modalities and directly fed into networks for video action recognition [1, 17, 37, 47] and semantic segmentation [39]. Such motion information also helps compensate for the spatial misalignment of features from different frames. Wang et al. [44] leveraged motion vectors to warp features in previous frames for object detection. Fan et al. [12] conditionally executed the backbone network for pose estimation depending on the residual values. For VSS, several methods have been proposed for efficient segmentation in the compressed domain. Jain et al. [18] warped the former and latter keyframe features using motion vectors, and predicted the non-keyframe features by interpolation. Tanujaya et al. [40] warped the results of keyframe segmentation for non-keyframes, and refined the warped segmentation by guided inpainting. Feng et al. [14] replaced a block of warped features with a local non-keyframe feature patch for further refinement. These methods reduced the computational cost of VSS, but suffered from performance degradation due to the limited capability of their feature refinement modules designed for non-keyframes. In our proposed AR-Seg, the warped features are refined by local attention mechanism according to the LR features of non-keyframes. Our attention-based refinement selectively aggregates the warped features and effectively suppresses the noise in motion vectors, achieving good segmentation accuracy with little computational overhead.

3. Method

In order to achieve efficient VSS for compressed videos, we propose an altering resolution framework named AR-Seg (Section 3.1). AR-Seg uses two branches to process HR keyframes and LR non-keyframes in the video separately. To compensate for the information loss due to downsampling the non-keyframes, we design a novel CReFF module that aggregates HR keyframe features into LR non-keyframe features (Section 3.2). To guide the learning of aggregated features, we further propose a novel feature similarity training (FST) strategy containing both explicit and implicit constraints (Section 3.3).

3.1. AR-Seg Framework

For image/video semantic segmentation tasks, the input resolution directly determines the amount of computation, no matter what type of algorithms are applied, e.g., Conditional Random Fields, CNNs, and Transformers. Although reducing resolution has been studied in the context of video recognition [26, 29], altering resolution in dense prediction tasks like VSS remains unexplored. Based on this observation, we design the AR-Seg framework for VSS, which inputs video frames with altering resolutions; i.e., in AR-Seg, only a few keyframes are processed at high resolution to preserve fine details, while other non-keyframes are processed at low resolution to reduce the computational cost.

To identify the keyframes, we make use of the frame structure in a group of pictures (GOP) encoded in compressed videos [38, 46]. A GOP includes L consecutive frames of three types: I frame, P frame, and B frame. I frames are encoded in intra mode, while P and B frames are encoded in inter mode that computes motion vectors for motion compensation. In each GOP, we treat the first I frame as a keyframe and process it at high resolution. The remaining $L-1$ frames in GOP are non-keyframes and processed at low resolution. To simplify the description of our method, following the previous works [14, 47], we only consider the GOP structure without B frames in this section. The full treatment including B frames is presented in Appendix A4.

Due to the domain gap between images with different resolutions, it is difficult for a single network to extract features suitable for both HR and LR resolution images. Thus we design two branches with shared backbone architecture in the model and train them separately for each resolution. Figure 1(c) summarizes the proposed AR-Seg framework. It consists of two branches: an HR branch for keyframes and an LR branch for non-keyframes.

The HR branch adapts an image segmentation network, which consists of a feature sub-network N_{feat}, a task sub-network N_{task}, and a final 1x1 convolutional layer[1]. It predicts segmentation results in high-resolution and simultaneously provides intermediate features before the final convolution as a reference for the LR branch. The LR branch is equipped with the same backbone network as the HR branch. To prevent performance degradation caused by downsampling, we design a CReFF module that aggregates HR features of the keyframe into the LR branch and place it before the final convolution layer of the backbone network.

[1] We follow the modular division of backbone networks in [19]. Two backbones are discussed and compared in Section 4.

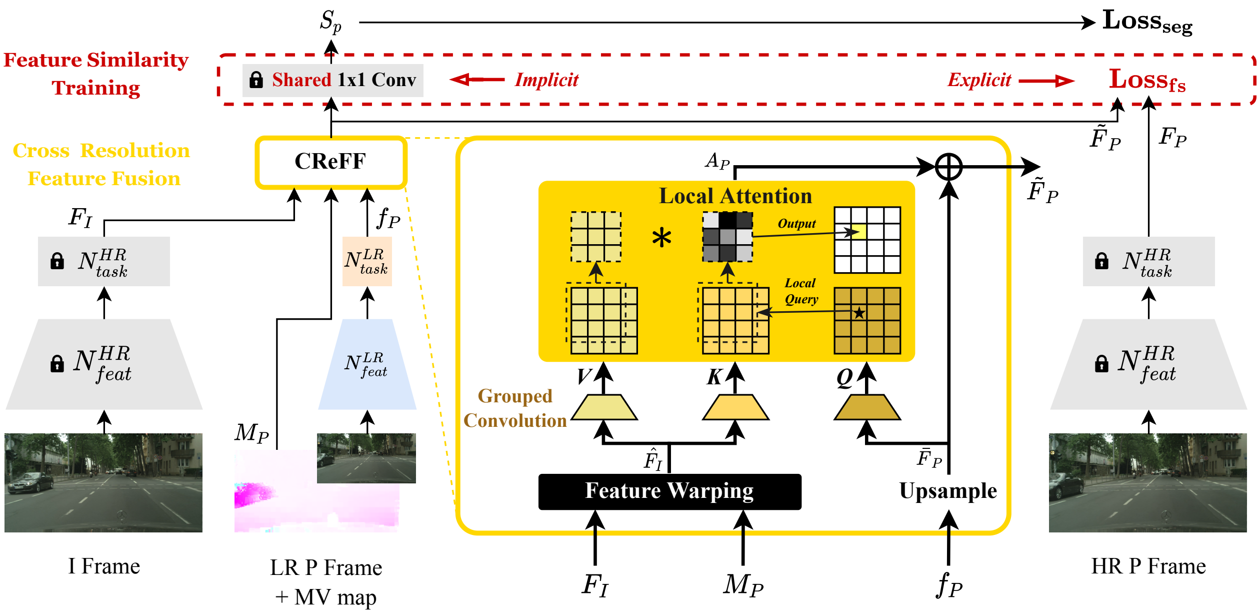


Figure 2. The CReFF module in the network architecture and feature similarity training (FST) strategy. CReFF consists of feature warping $\mathcal{W}_{MV}$ and local-attention-based feature fusion $\mathcal{F}_{LA}$ (Section 3.2). FST includes the explicit supervision by the feature similarity loss and the implicit supervision from shared convolution (Section 3.3). Parameters in gray blocks are **fixed** during the training of the LR branch.

CReFF aggregates the HR reference features into the extracted LR features, yielding estimated HR features for the non-keyframes, which are further converted into pixel-wise semantic labels by the final convolution layer.

As illustrated in Figure 1, different from the previous Accel framework [19], AR-Seg performs feature fusion before the last 1x1 convolution layer, instead of before the task sub-network N_{task}. The reason is twofold: 1) Since feature maps before the final convolution have basically the same spatial layout as the input images and segmentation outputs, we can utilize motion vectors to compensate for the spatial misalignment of features at such position; and 2) As the CReFF module actually upsamples the LR features, such a placement allows almost all convolution layers to benefit from the low resolution, thus reducing most of the computational cost.

3.2. CReFF: Cross Resolution Feature Fusion

In the AR-Seg framework, CReFF aims to prevent the performance degradation caused by the lack of fine local details in LR non-keyframes. Unlike a single image, video frames are intrinsically temporally correlated, so missing details in LR non-keyframes can be retrieved from the corresponding regions in HR keyframes according to motion cues. Motion vectors (MVs) in the compressed video exactly provide such motion cues at the block level, i.e., pixels inside a macroblock share the same motion vector. Almost all mainstream video compression standards use motion vectors for inter-prediction, including H.26x series [38, 46], AOMedia series [7] and AVS series [10, 15, 53]. Such block-wise MVs are readily available in compressed videos and can be used to assist the LR branch.

Specifically, as depicted in Figure 2, the HR branch of AR-Seg extracts the feature $F_I \in \mathcal{R}^{C\times H\times W}$ from an I frame, and the LR branch extracts the feature $f_P \in \mathcal{R}^{C\times h\times w}$ from a P frame. Although P frames are processed in low resolution, CReFF takes F_I, M_P, and f_P as input to generate the aggregated feature $\tilde{F}_P$, where $M_P \in \mathcal{R}^{2\times H\times W}$ denotes the MVs from P frame to I frame. The two channels of M_P correspond to x and y dimensions of motion vectors, denoted by c_x and c_y. Inside the CReFF module, the MV-based feature warping operation $\mathcal{W}_{MV}$ firstly warps F_I to the spatial layout of the P frame, which can be formulated as per-pixel shifting:

$$\hat{F}_I^{(x,y)} = F_I^{(x+M_P^{(c_x,x,y)},\,y+M_P^{(c_y,x,y)})}, \tag{1}$$

where $\hat{F}_I \in \mathcal{R}^{C\times H\times W}$ denotes the warped HR feature that will be further fused into LR features.

Due to the coarse-grained MVs (block-level instead of pixel-level) and the varying occlusion relationships across video frames, the warped features $\hat{F}_I$ are often noisy and misleading. Inspired by the success of non-local operation [45] and attention mechanism [16, 31, 33] in video-based applications, we propose to assign different fusion importance weights to the (x, y) locations in noisy features $\hat{F}_I$ by attention mechanism. Since $\hat{F}_I$ is roughly spatially aligned to f_p after warping, we use local attention to efficiently fuse the features as follows.

In the local-attention-based feature fusion module $\mathcal{F}_{LA}$, we firstly generate the ***Value*** and ***Key*** feature maps from the warped HR features $\hat{F}_I$, and the ***Query*** maps from the upsampled LR features $\bar{F}_P$. The 3×3 grouped convolution $\mathbf{Conv}_g$ with $groups = C$ is selected to efficiently encode the feature maps into attention representations $V_I, K_I, Q_P \in \mathcal{R}^{C \times H \times W}$. Note that attention representations share the same channel size as the intermediate features. Denote the $n \times n$ neighborhood centered at (x, y) as $Nbhd_{(x,y)}$. Within $Nbhd_{(x,y)}$, the output of local attention A_P at the position (x, y) is generated by

$$A_P{}^{(x,y)} = \overline{V_I{}^{Nbhd_{(x,y)}}} \mathcal{S}(\overline{K_I{}^{Nbhd_{(x,y)}}}, Q_P{}^{(x,y)}), \tag{2}$$

where $A_P{}^{(x,y)}$, $Q_P{}^{(x,y)} \in \mathcal{R}^{C \times 1}$ denote the feature vectors at the position (x, y) of A_P and Q_P respectively, and $\overline{V_I{}^{Nbhd_{(x,y)}}}$, $\overline{K_I{}^{Nbhd_{(x,y)}}} \in \mathcal{R}^{C \times n^2}$ are the re-arranged feature vectors within $Nbhd_{(x,y)}$ in V_I and K_I, respectively. The similarity operation $\mathcal{S}(K, Q)$ is formulated as

$$\mathcal{S}(K, Q) = Softmax(\frac{K^T Q}{\sqrt{C}}). \tag{3}$$

Furthermore, the aggregated feature $\tilde{F}_P$ for P frame is obtained in a residual fashion:

$$\tilde{F}_P = \bar{F}_P + A_P = Upsample(f_P) + A_P. \tag{4}$$

In summary, using the CReFF module, the feature details from the I frame are firstly aligned to the P frame, and then aggregated into the LR branch according to the pixel-wise similarity between Q_P and K_I. The verification of the architecture design of CReFF is presented in Section 4.3. The reader is referred to Appendix A1 for more details on the visualization of attention weights in $\mathcal{F}_{LA}$.

3.3. FST: Feature Similarity Training

In order to effectively train the CReFF module, we propose a feature similarity training (FST) strategy. FST utilizes the HR features of P frame F_P (extracted from the HR branch) to guide the learning of the aggregated features $\tilde{F}_P$ in the LR branch. Since F_P contains sufficient details to produce high-quality segmentation results, CReFF can learn how to aggregate $\bar{F}_P$ and $\hat{F}_I$ into effective HR features from it under the supervision of FST. Specifically, FST supervises the training process of the LR branch both *explicitly* and *implicitly* in the following ways.

The *explicit* constraint is to use the feature similarity loss $\mathcal{L}_{fs}$. We use mean square error (MSE) to measure the difference between $\tilde{F}_P$ and F_P, which serves as an additional regularization for the LR model:

$$\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{fs} = CE(S_P, G_P) + MSE(\tilde{F}_P, F_P), \tag{5}$$

where $S_P \in \mathcal{R}^{C_{out} \times H \times W}$ denotes the segmentation result produced by LR branch, $G_P \in \mathcal{R}^{H \times W}$ denotes the ground-truth segmentation of P frame and the segmentation loss $\mathcal{L}_{seg}$ is the standard cross entropy loss $CE(S_P, G_P)$.

The *implicit* constraint of FST is the shared decoding layer of $\tilde{F}_P$ and F_P. In the segmentation backbone model trained on the HR images, the final convolution layer acts as the segmentation decoder, which contains deep semantic information about high-quality HR features. To utilize such information, we directly transfer the final 1×1 convolution layer of the HR branch to the LR branch with fixed parameters. Since the parameters are trained on HR features, they produce better segmentation results S_P when $\tilde{F}_P$ is closer to the HR feature F_P.

In summary, with the *explicit* and *implicit* constraints, FST effectively transfers the knowledge of HR features from the HR branch to the LR branch, enabling high-quality segmentation based on the aggregated features of CReFF. Figure 2 shows the overall training strategy for the LR branch. The HR I frame provides the features F_I for feature fusion in CReFF, and the HR P frame provides the features F_P for the *explicit* supervision in FST. Parameters of the LR branch are trained using the total loss $\mathcal{L}$ via backpropagation, with fixed parameters of the HR branch and the shared final convolution layer.

4. Experiments

We evaluate the proposed AR-Seg framework on CamVid [3] and Cityscapes [9] datasets for street-view video semantic segmentation. Below we present experiments to demonstrate the efficiency of AR-Seg and its compatibility with different backbone models, resolutions, and video compression configurations.

4.1. Experimental Setup

Datasets & Pre-processing. The *CamVid* [3] dataset consists of 4 videos of 720×960 resolution captured at 30 fps, with semantic annotations at 1Hz and, in part, 15Hz. The *Cityscapes* [9] dataset contains street view videos of 1024×2048 resolution captured in 17 fps, from which 5,000 images are densely annotated. We use the official train/validation/test split for both datasets. Following the previous works [16, 51], we select 11 and 19 classes for training and evaluation on these two datasets, respectively.

To simulate a real video compressed scenario, we compress the videos at reasonable bit-rates of 3Mbps for CamVid and 5Mbps for Cityscapes with the HEVC/H.265 [38] standard. The reader is referred to Appendix A3.1 for the detailed pre-processing steps.

Models & Baselines. To demonstrate the compatibility of AR-Seg with different backbones, we select two representative image segmentation models in our experiments: PSPNet [55] and BiseNet [52], which is similar to settings in the previous work [16]. The former is a widely used classical model, and the latter is a lightweight model that achieves state-of-the-art performance. We use **AR**$^{\alpha}$**-** as the prefix of AR-Seg with specified backbone networks, where α denotes the downsample scale for the LR branch.

Training & Evaluation Details. Given a GOP length L, we train the LR branch with image pairs (i, p), where p refers to the P frame with annotation and $i = p - (L - 1)$ refers to the I frame as a reference. We denote the distance between the annotated and the reference frames as d, then $d = L - 1$ for the training pairs.

For evaluation, we test AR-Seg with different distances d between the target frame p and the reference keyframe i. For $d = 0$, we treat frame p as the keyframe and process it by the HR branch. Otherwise, we feed frame p into the LR branch for $d \in (0, L - 1]$. The average of $mIoU_d$ for each distance d is reported as the mIoU result. We measure FLOPs by PyTorch-OpCounter [35] following the previous methods [30, 32]. All the comparisons are evaluated on the compressed videos. More training and evaluation details are presented in Appendix A3.2 and A3.3.

4.2. Experiment Results

Comparison with image-based methods. We first compare AR-Segs (with PSPNet [55] and BiseNet [52] as backbone) to their image-based counterparts of 1.0x resolution. As shown in Table 1, on both CamVid and Cityscapes datasets, the proposed AR$^{0.5}$- models achieve on-par or better performance than the 1.0x resolution baselines while saving 67% computational cost. Different from the low-resolution baselines that lead to significant performance degradation, AR-Seg successfully preserves the segmentation accuracy with the help of the CReFF module and the FST strategy. More comparisons between AR-Seg and the LR baselines under different resolutions are presented in Appendix A2.1. Furthermore, these experiment results with PSPNet and BiseNet also demonstrate the compatibility of AR-Seg for different backbone networks.

Comparison with video-based methods. Taking temporal coherence into consideration, we compare AR-Seg with the recent state-of-the-art video-based methods for efficient VSS. Besides the accuracy and computational cost, we also follow the previous work [33] to report the relative changes compared to their single-frame backbone models. Specifically, $\widetilde{\Delta}$mIoU denotes the relative change of mIoU, and $\widetilde{\Delta}$GFLOPs denotes the relative change of GFLOPs. As shown in Table 2, existing video-based methods usually improve accuracy ($\widetilde{\Delta}$mIoU>0) at the cost of more computation ($\widetilde{\Delta}$GFLOPs>0), e.g., TDNet [16] and Accel [19]. Other methods, including BlockCopy [42], TapLab [14] and Jain et al. [18], reduce the computational cost ($\widetilde{\Delta}$GFLOPs<0) but the accuracy also decreases 3%-7% ($\widetilde{\Delta}$mIoU<0). As a comparison, our proposed models AR$^{0.6}$- can reduce the computational cost (**$\widetilde{\Delta}$GFLOPs<0**) by more than 55% and preserve the accuracy of single-frame backbone models (**$\widetilde{\Delta}$mIoU≥0**). With the lightweight backbone model BiseNet, AR$^{0.6}$-Bise18 achieves good performance in both accuracy and computational cost. More results of video-based methods and their single-frame backbone models are presented in Appendix A2.2.

Table 1. Comparison to the image-based methods on CamVid *test* set and Cityscapes *valid* set.

	Method	PSPNet18 [55]		BiseNet18 [52]	
		mIoU(%)↑	GFLOPs ↓	mIoU(%)↑	GFLOPs ↓
CamVid	1.0x	69.43	309.02	71.57	58.83
	AR$^{0.7}$	**71.23**	169.86	**71.78**	31.89
	AR$^{0.6}$	70.82	133.09	71.60	24.68
	AR$^{0.5}$	70.48	**101.98**	70.38	**18.96**
Cityscapes	1.0x	69.00	560.97	70.09	178.96
	AR$^{0.7}$	**70.23**	302.95	**70.86**	97.10
	AR$^{0.6}$	69.45	234.91	70.72	76.06
	AR$^{0.5}$	69.03	**177.44**	70.57	**57.00**

Table 2. Comparison to the video-based methods on CamVid *test* set and Cityscapes *valid* set. $\widetilde{\Delta}x = \frac{\Delta x}{|x|}$ denotes the relative change compared to their single-frame backbone models. The best results are bold and the second best results are underlined.

	Method	mIoU↑	GFLOPs↓	$\widetilde{\Delta}$mIoU↑	$\widetilde{\Delta}$GFLOPs↓
CamVid	Accel-DL18 [19]	66.15	397.70	**+13.8%**	+61.9%
	TD^4-PSP18 [16]	70.13	363.70	+1.0%	+17.7%
	BlockCopy [42]	66.75	107.52	-5.2%	-45.7%
	TapLab-BL2 [14]	67.57	117.73	-3.1%	-50.2%
	Jain et al. [18]	67.61	146.97	-4.3%	-53.8%
	AR$^{0.6}$-PSP18	<u>70.82</u>	<u>101.98</u>	<u>+2.0%</u>	<u>-57.0%</u>
	AR$^{0.6}$-Bise18	**71.60**	**24.68**	+0.0%	**-58.0%**
Cityscapes	Accel-DL18 [19]	68.25	1011.75	**+18.4%**	+96.0%
	TD^4-PSP18 [16]	<u>70.11</u>	673.06	<u>+1.6%</u>	+20.0%
	BlockCopy [42]	67.69	294.20	-6.7%	-41.2%
	TapLab-BL2 [14]	68.90	237.29	-4.1%	-50.6%
	Jain et al. [18]	68.57	342.67	-5.1%	-52.5%
	AR$^{0.6}$-PSP18	69.45	<u>234.91</u>	+0.7%	**-58.1%**
	AR$^{0.6}$-Bise18	**70.72**	**76.06**	+0.9%	<u>-57.5%</u>

4.3. Ablation Study

We perform ablation studies to show the importance of each component in CReFF and FST, as well as the location of CReFF. We also evaluate AR-Seg in terms of different resolutions and GOP lengths, which reflects the influence of hyper-parameters α and L, respectively. We conducted ablation studies on the 30fps CamVid dataset, and used PSPNet18 as the default backbone model, $L = 12$ as the default GOP length, and HEVC@3Mbps as the default codec.

Architecture of CReFF. We first validate the necessity of $\mathcal{W}_{MV}$ and $\mathcal{F}_{LA}$. The method without CReFF directly applies FST to the upsampled features $\bar{F}_P$ and serves as a baseline. As shown in Table 3, simply warping the keyframe features (+ $\mathcal{W}_{MV}$) saves the most amount of computation by skipping processing non-keyframes with the

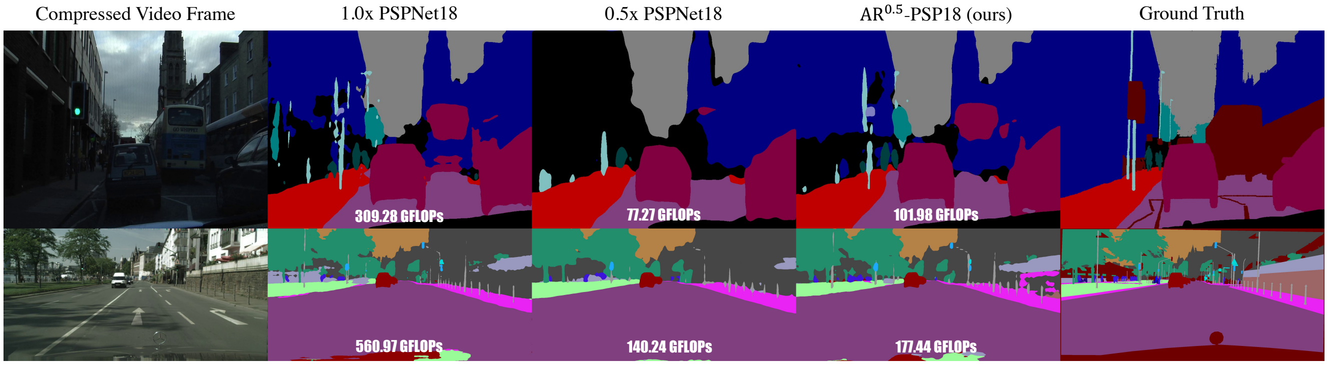


Figure 3. Semantic segmentation on CamVid (top) and Cityscapes (bottom) with $d = 11$. Note that AR$^{0.5}$-PSP18 predicts more semantic details than the constant-resolution PSPNet18 working on 0.5x resolution. Compared to the 1.0x baseline, AR$^{0.5}$-PSP18 generates similar segmentation results, but consumes only 33.0% computational cost (measured in GFLOPs).

segmentation network, but receives poor accuracy. Directly fusing the keyframe features using local attention (+ $\mathcal{F}_{LA}$) does not perform very well, because the feature maps are not well-aligned due to object motion in videos.

We further evaluate the design of $\mathcal{F}_{LA}$ by replacing 7x7 local attention (LA) with other operations, including LA with different neighborhood sizes, LA with non-grouped convolution encoders ($\mathcal{F}_{LA^*}$), global attention ($\mathcal{F}_{GA}$) and one-layer convolution ($\mathcal{F}_{Conv}$). For $\mathcal{F}_{GA}$, we downsample the ***Value*** and ***Key*** maps by $\frac{1}{32}$ to save the computation. $\mathcal{F}_{Conv}$ processes the concatenated feature $[\hat{F}_I, \bar{F}_P]$ with a 3x3 convolution. Due to the large channel number in deep layers, such an operation brings considerable computation overhead. Results in Table 3 show that $\mathcal{F}_{LA}$ with a 7x7 neighborhood achieves a good balance between computation and accuracy. Other designs increase the computational cost without improving the accuracy. Furthermore, removing the direct connection (DC) path from CReFF reduces the mIoU to 69.14%. This result implies that CReFF is more likely to learn the residuals of HR features than the absolute values.

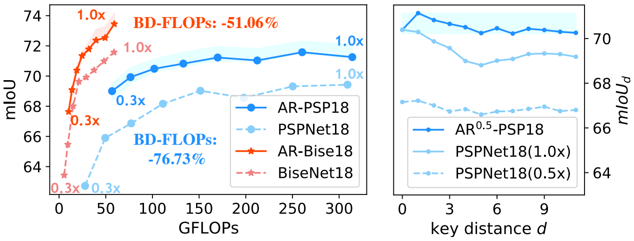
Location of CReFF. As specified in Section 3.1, we place CReFF before the final 1x1 convolution layer, which is different from the previous Accel framework [19]. To evaluate the influence of the split point location, we insert CReFF before different layers and evaluate the performance. Split points include the final convolution layer ($C_{1\times1}$), the task sub-network N_{task} and the feature sub-network N_{feat}. For the N_{feat} case, we insert CReFF after the first convolution layer of ResNet. As shown in Table 3, placing CReFF before $C_{1\times1}$ results in the best performance, which affirms our design in Section 3.1. We note that fusing features at an early stage does not improve accuracy, but rather considerably increases computational cost.

Feature Similarity Training (FST). The proposed FST strategy consists of the MSE Loss and the shared final convolution layer $C_{1\times1}$. We train AR$^{0.5}$-PSP18 with or without these components and report the results in Table 3. Both components improve the segmentation performance compared to the model trained without FST. We also replace the MSE Loss with the Kullback-Leibler (KL) Divergence Loss, but the resulting segmentation performance is poor.

Table 3. Ablation experiments on CamVid dataset with PSPNet18. Settings used in our final model are underlined.

Experiment	Method	mIoU(%)	GFLOPs
Baseline	PSPNet18 (1.0x)	69.43	309.02
	PSPNet18 (0.5x)	66.51	77.27
Architecture of CReFF	$\underline{+ \mathcal{W}_{MV} + \mathcal{F}_{LA}\ (7x7)}$	**70.48**	**101.98**
	w/o CReFF	67.14	96.60
	+ $\mathcal{W}_{MV}$	57.64	25.75
	+ $\mathcal{F}_{LA}$ (7x7)	67.93	101.98
	+ $\mathcal{W}_{MV}$ + $\mathcal{F}_{LA}$ (3x3)	70.30	98.74
	+ $\mathcal{W}_{MV}$ + $\mathcal{F}_{LA}$ (11x11)	70.48	107.32
	+ $\mathcal{W}_{MV}$ + $\mathcal{F}_{LA^*}$ (7x7)	69.99	170.96
	+ $\mathcal{W}_{MV}$ + $\mathcal{F}_{GA}$ (1/32)	67.11	113.58
	+ $\mathcal{W}_{MV}$ + $\mathcal{F}_{Conv}$	70.45	143.63
	+ CReFF w/o DC	69.14	101.98
Location of CReFF	$\underline{\text{before } C_{1\times1}}$	**70.48**	**101.98**
	before N_{task}	68.60	214.76
	before N_{feat}	68.31	308.46
Feature Similarity Training (FST)	$\underline{\text{+ MSE Loss + Shared } C_{1\times1}}$	**70.48**	101.98
	w/o FST	69.21	101.98
	+ Shared $C_{1\times1}$	69.57	101.98
	+ MSE Loss	70.17	101.98
	+ KL Loss + Shared $C_{1\times1}$	68.91	101.98
Keyframe Interval	$\underline{\text{AR}^{0.5}\text{-PSP18, L=12}}$	**70.48**	101.98
	AR$^{0.5}$-PSP18, L=15	70.28	97.88
	AR$^{0.5}$-PSP18, L=20	70.28	94.11
	AR$^{0.5}$-PSP18, L=30	69.67	**90.34**

Resolution of LR Branch. By adjusting the resolution of the LR branch, AR-Seg can tailor the setting adaptive to



(a) (b)

Figure 4. (a) Performance of AR-Seg with different resolutions for the LR branch. (b) $mIoU_d$ for annotated frames with different distances to key frame. Value intervals of $mIoU_d$ when d varies from 1 to $L-1$ are depicted as color bars in (a).

different computational budgets. Here, we train and evaluate AR-Seg with different resolutions for the LR branch, ranging from 0.3x to 1.0x. We also train and evaluate the constant-resolution baselines of each resolution for comparison. As shown in Figure 4(a), AR-Seg improves both backbones under all resolutions, demonstrating the effectiveness of CReFF and FST. To quantify the average improvement, we utilize two metrics *BD-FLOPs* and *BD-mIoU* following the design of BD-Rate and BD-PSNR [2] in video compression. The results show that (1) with the same computational budget, AR-Seg improves the absolute accuracy for two backbones by 3.67% and 3.02% respectively, and (2) with the same accuracy, AR-Seg reduces the computational cost by 76.73% and 51.06% respectively. Both metrics are described in detail in Appendix A5.

The Temporal Gap. To investigate the influence of the distance d to the keyframe, we plot the $mIoU_d$ results for AR$^{0.5}$-PSP18 and the constant-resolution baselines in Figure 4(b). As $mIoU_0$ is determined by the HR branch, AR$^{0.5}$-PSP18 shares the same point with PSPNet18(1.0x) at $d = 0$. The $mIoU_0$ for PSPNet18(0.5x) is much lower due to downsampling. When $d > 0$, the accuracy of PSPNet18(1.0x) decreases since the compression artifacts in P frames are more severe than those in I frames. As a comparison, the AR$^{0.5}$-PSP18 benefits from the CReFF module and maintains high accuracy for all the d values.

Keyframe Intervals. To validate the long-range reference, we extend our evaluation to different keyframe intervals without re-training. As shown in Table 3, AR$^{0.5}$-PSP18 trained with $L = 12$ maintains good performance with different GOP lengths. Moreover, even for $L = 30$, which stands for 1s in 30fps videos and is larger than the discussion in previous works [19, 56], AR-Seg outperforms the 1.0x baseline using only 29.2% FLOPs.

Video Compression Configurations. As shown in Table 4, we also train and evaluate our model with different realistic bit-rates (3Mbps and 1Mbps) and configurations for HEVC/H.265 encoders. We use x265-*medium* and x265-*ultrafast* to represent different presets for x265, which apply simplified motion search algorithms and larger macroblocks. These configurations are widely used in traditional video encoders. The results show that AR$^{0.5}$-PSP18 consistently outperforms the 1.0x constant resolution counterpart using only 33% GFLOPs under different configurations.

Table 4. Performance of AR-Seg on videos compressed by different codecs. AR-Seg achieves comparable or even better accuracy than its image-based constant-resolution counterparts.

Codec	Method	mIoU(%)	GFLOPs
HEVC@3Mbps	PSPNet18(1.0x)	69.43	309.02
	AR$^{0.5}$-PSP18	**70.48**	**101.98**
HEVC@1Mbps	PSPNet18(1.0x)	65.76	309.02
	AR$^{0.5}$-PSP18	**67.89**	**101.98**
x265-*medium*@3Mbps	PSPNet18 (1.0x)	68.19	309.02
	AR$^{0.5}$-PSP18	**69.53**	**101.98**
x265-*ultrafast*@3Mbps	PSPNet18 (1.0x)	67.69	309.02
	AR$^{0.5}$-PSP18	**68.78**	**101.98**

4.4. Running Time

We measure the running time of AR-Seg with PSPNet18 on both CamVid and Cityscapes datasets, and the results are reported in Table 5. Our AR-Seg models run 2x-3x times faster than the constant resolution counterparts. All tests are executed on a single GeForce RTX 3090 GPU.

Table 5. Running time of AR-PSP18 on 720x960 CamVid and 1024x2048 Cityscapes datasets.

Dataset	1.0x baseline	AR$^{0.5}$	AR$^{0.3}$
CamVid	31.2 ms (32fps)	14.7 ms (68fps)	9.0 ms (111fps)
Cityscapes	95.4 ms (10fps)	30.7 ms (33fps)	19.9 ms (50fps)

5. Conclusion

In this paper, we propose AR-Seg, an altering resolution framework for compressed video semantic segmentation, which innovatively improves the efficiency of video segmentation from the perspective of input resolution. By jointly considering the design of architecture and training strategy, our proposed CReFF module and FST strategy effectively prevent the accuracy loss caused by downsampling. Results evaluated on two widely used datasets show that AR-Seg can achieve competitive segmentation accuracy with a reduction of up to 67% computational cost. Our current study only uses two alternating resolutions (i.e., HR and LR). Future work that applies more complicated scheduling of multi-resolutions and keyframe gaps will be considered to further improve the VSS performance.

Aknowledgements: This work was partially supported by Beijing Natural Science Foundation (L222008), Tsinghua University Initiative Scientific Research Program, the Natural Science Foundation of China (61725204), BNRist, and MOE-Key Laboratory of Pervasive Computing.

References

[1] Barak Battash, Haim Barad, Hanlin Tang, and Amit Bleiweiss. Mimic the raw domain: Accelerating action recognition in the compressed domain. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 2926–2934. Computer Vision Foundation / IEEE, 2020. 3

[2] G. Bjontegaard. *Calculation of average PSNR differences between RD-curves*. VCEG-M33, Austin, TX, USA, Apr. 2001. 8

[3] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.*, 30(2):88–97, 2009. 2, 5

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[5] 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 Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018. 2

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 1

[7] Yue Chen, Debargha Mukherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, Urvang Joshi, Ching-Han Chiang, Yunqing Wang, Paul Wilkins, Jim Bankoski, Luc N. Trudeau, Nathan E. Egge, Jean-Marc Valin, Thomas Davies, Steinar Midtskogen, Andrey Norkin, and Peter De Rivaz. An overview of core coding tools in the AV1 video codec. In *2018 Picture Coding Symposium, PCS 2018, San Francisco, CA, USA, June 24-27, 2018*, pages 41–45. IEEE, 2018. 4

[8] Dan C. Ciresan, Alessandro Giusti, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2852–2860, 2012. 2

[9] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223. IEEE Computer Society, 2016. 2, 5

[10] Liang Fan, Siwei Ma, and Feng Wu. Overview of AVS video standard. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo, ICME 2004, 27-30 June 2004, Taipei, Taiwan*, pages 423–426. IEEE Computer Society, 2004. 4

[11] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 9716–9725. Computer Vision Foundation / IEEE, 2021. 2

[12] Zhipeng Fan, Jun Liu, and Yao Wang. Motion adaptive pose estimation from compressed videos. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 11699–11708. IEEE, 2021. 3

[13] Clément Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1915–1929, 2013. 2

[14] Junyi Feng, Songyuan Li, Xi Li, Fei Wu, Qi Tian, Ming-Hsuan Yang, and Haibin Ling. Taplab: A fast framework for semantic video segmentation tapping into compressed-domain knowledge. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(3):1591–1603, 2022. 3, 6

[15] Wen Gao and Siwei Ma. An overview of avs2 standard. *Advanced Video Coding Systems*, pages 35–49, 2014. 4

[16] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8815–8824. Computer Vision Foundation / IEEE, 2020. 2, 4, 5, 6

[17] Yuqi Huo, Xiaoli Xu, Yao Lu, Yulei Niu, Mingyu Ding, Zhiwu Lu, Tao Xiang, and Ji-Rong Wen. Lightweight action recognition in compressed videos. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12536 of *Lecture Notes in Computer Science*, pages 337–352. Springer, 2020. 3

[18] Samvit Jain and Joseph E. Gonzalez. Fast semantic segmentation on video using block motion-based feature interpolation. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, volume 11132 of *Lecture Notes in Computer Science*, pages 3–6. Springer, 2018. 3, 6

[19] Samvit Jain, Xin Wang, and Joseph E. Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8866–8875. Computer Vision Foundation / IEEE, 2019. 1, 2, 3, 4, 6, 7, 8

[20] Byungju Kim, Junho Yim, and Junmo Kim. Highway driving dataset for semantic video segmentation. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 140. BMVA Press, 2018. 3

[21] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip H. S. Torr. Associative hierarchical crfs for object class im-

age segmentation. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, pages 739–746. IEEE Computer Society, 2009. 2

[22] Hanchao Li, Pengfei Xiong, Haoqiang Fan, and Jian Sun. 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*, pages 9522–9531. Computer Vision Foundation / IEEE, 2019. 1, 2

[23] Xin Li, Yiming Zhou, Zheng Pan, and Jiashi Feng. Partial order pruning: for best speed/accuracy trade-off in neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9145–9153, 2019. 1, 2

[24] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5997–6005. Computer Vision Foundation / IEEE Computer Society, 2018. 1, 2

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3431–3440. IEEE Computer Society, 2015. 1, 2

[26] Chuofan Ma, Qiushan Guo, Yi Jiang, Zehuan Yuan, Ping Luo, and Xiaojuan Qi. Rethinking resolution in the context of efficient video recognition. *CoRR*, abs/2209.12797, 2022. 3

[27] Behrooz Mahasseni, Sinisa Todorovic, and Alan Fern. 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*, pages 2077–2086. IEEE Computer Society, 2017. 1, 2

[28] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda G. Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, volume 11214 of *Lecture Notes in Computer Science*, pages 561–580. Springer, 2018. 1, 2

[29] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogério Feris. Ar-net: Adaptive frame resolution for efficient action recognition. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VII*, volume 12352 of *Lecture Notes in Computer Science*, pages 86–104. Springer, 2020. 3

[30] Yuval Nirkin, Lior Wolf, and Tal Hassner. 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*, pages 4061–4070. Computer Vision Foundation / IEEE, 2021. 6

[31] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9225–9234. IEEE, 2019. 4

[32] Marin Orsic, Ivan Kreso, Petra Bevandic, and Sinisa Segvic. 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*, pages 12607–12616. Computer Vision Foundation / IEEE, 2019. 6

[33] Matthieu Paul, Martin Danelljan, Luc Van Gool, and Radu Timofte. Local memory attention for fast video semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pages 1102–1109. IEEE, 2021. 4, 6

[34] Matthieu Paul, Christoph Mayer, Luc Van Gool, and Radu Timofte. Efficient video semantic segmentation with labels propagation and refinement. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 2862–2871. IEEE, 2020. 1

[35] PyTorch-OpCounter Contributors. THOP: PyTorch-OpCounter, 11 2018. 6

[36] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In Gang Hua and Hervé Jégou, editors, *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III*, volume 9915 of *Lecture Notes in Computer Science*, pages 852–868, 2016. 2

[37] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1268–1277. Computer Vision Foundation / IEEE, 2019. 3

[38] Gary J. Sullivan, Jens-Rainer Ohm, Woojin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12):1649–1668, 2012. 3, 4, 5

[39] Zhentao Tan, Bin Liu, Qi Chu, Hangshi Zhong, Yue Wu, Weihai Li, and Nenghai Yu. Real time video object segmentation in compressed domain. *IEEE Trans. Circuits Syst. Video Technol.*, 31(1):175–188, 2021. 3

[40] Stefanie Tanujaya, Tieh Chu, Jia-Hao Liu, and Wen-Hsiao Peng. Semantic segmentation on compressed video using block motion compensation and guided inpainting. In *IEEE International Symposium on Circuits and Systems, ISCAS 2020, Sevilla, Spain, October 10-21, 2020*, pages 1–5. IEEE, 2020. 3

[41] Joseph Tighe and Svetlana Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision - ECCV 2010 - 11th European Conference*

on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V, volume 6315 of *Lecture Notes in Computer Science*, pages 352–365. Springer, 2010. 2

[42] Thomas Verelst and Tinne Tuytelaars. Blockcopy: High-resolution video processing with block-sparse feature propagation and online policies. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5138–5147. IEEE, 2021. 3, 6

[43] Li Wang, Dong Li, Yousong Zhu, Lu Tian, and Yi Shan. 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*, pages 3773–3782. Computer Vision Foundation / IEEE, 2020. 2

[44] Shiyao Wang, Hongchao Lu, and Zhidong Deng. Fast object detection in compressed video. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7103–7112. IEEE, 2019. 3

[45] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7794–7803. Computer Vision Foundation / IEEE Computer Society, 2018. 4

[46] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, 2003. 3, 4

[47] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Compressed video action recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6026–6035. Computer Vision Foundation / IEEE Computer Society, 2018. 3

[48] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12077–12090, 2021. 1

[49] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12077–12090, 2021. 1

[50] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6556–6565. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[51] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet V2: bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vis.*, 129(11):3051–3068, 2021. 1, 2, 5

[52] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, volume 11217 of *Lecture Notes in Computer Science*, pages 334–349. Springer, 2018. 1, 2, 5, 6

[53] Jiaqi Zhang, Chuanmin Jia, Meng Lei, Shanshe Wang, Siwei Ma, and Wen Gao. Recent development of AVS video coding standard: AVS3. In *Picture Coding Symposium, PCS 2019, Ningbo, China, November 12-15, 2019*, pages 1–5. IEEE, 2019. 4

[54] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pages 418–434. Springer, 2018. 1, 2

[55] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239. IEEE Computer Society, 2017. 1, 2, 5, 6

[56] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4141–4150. IEEE Computer Society, 2017. 2, 8