JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

THU-PM-193

Efficient Semantic Segmentation by Altering Resolutions for Compressed Videos

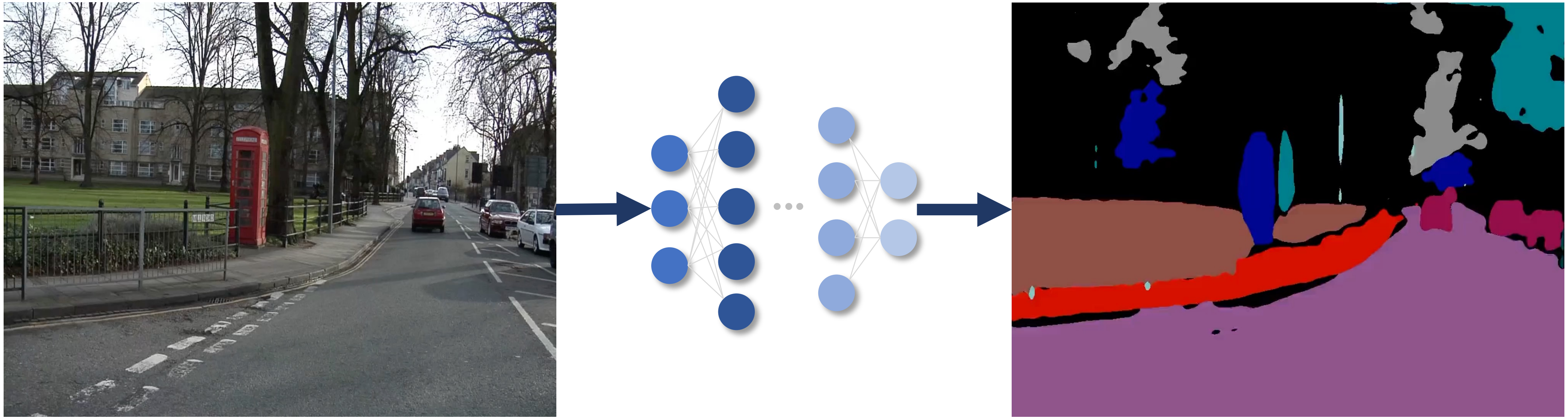
Yubin Hu[1], Yuze He[1], Yanghao Li[1], Jisheng Li[1], Yuxing Han[2], Jiangtao Wen[3] and Yong-jin Liu[1]

1, 2 清華大學 Tsinghua University

3 东方理工高等研究院 EASTERN INSTITUTE FOR ADVANCED STUDY

Overview

- Video semantic segmentation (VSS) is a **computationally expensive** task.

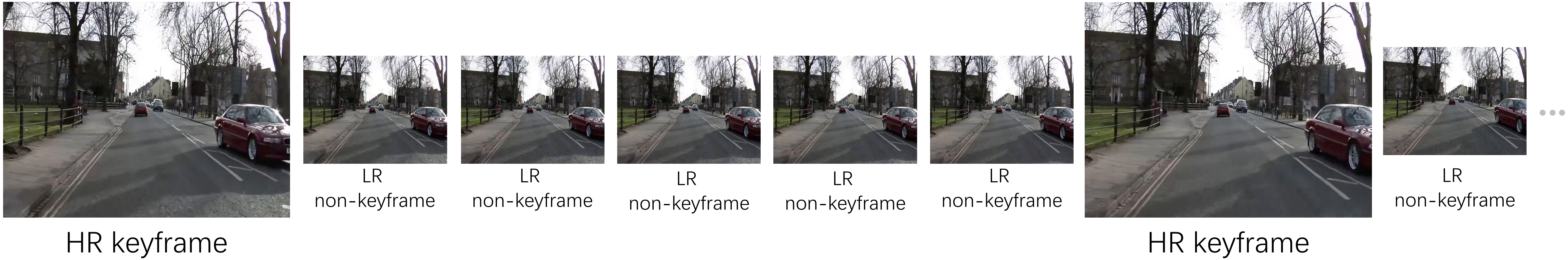


Input

Segmentation results of *PSPNet (ResNet-18).*

Overview

- Video semantic segmentation (VSS) is a **computationally expensive** task.
  - We propose to reduce the GLOPs by **altering the input resolution**.



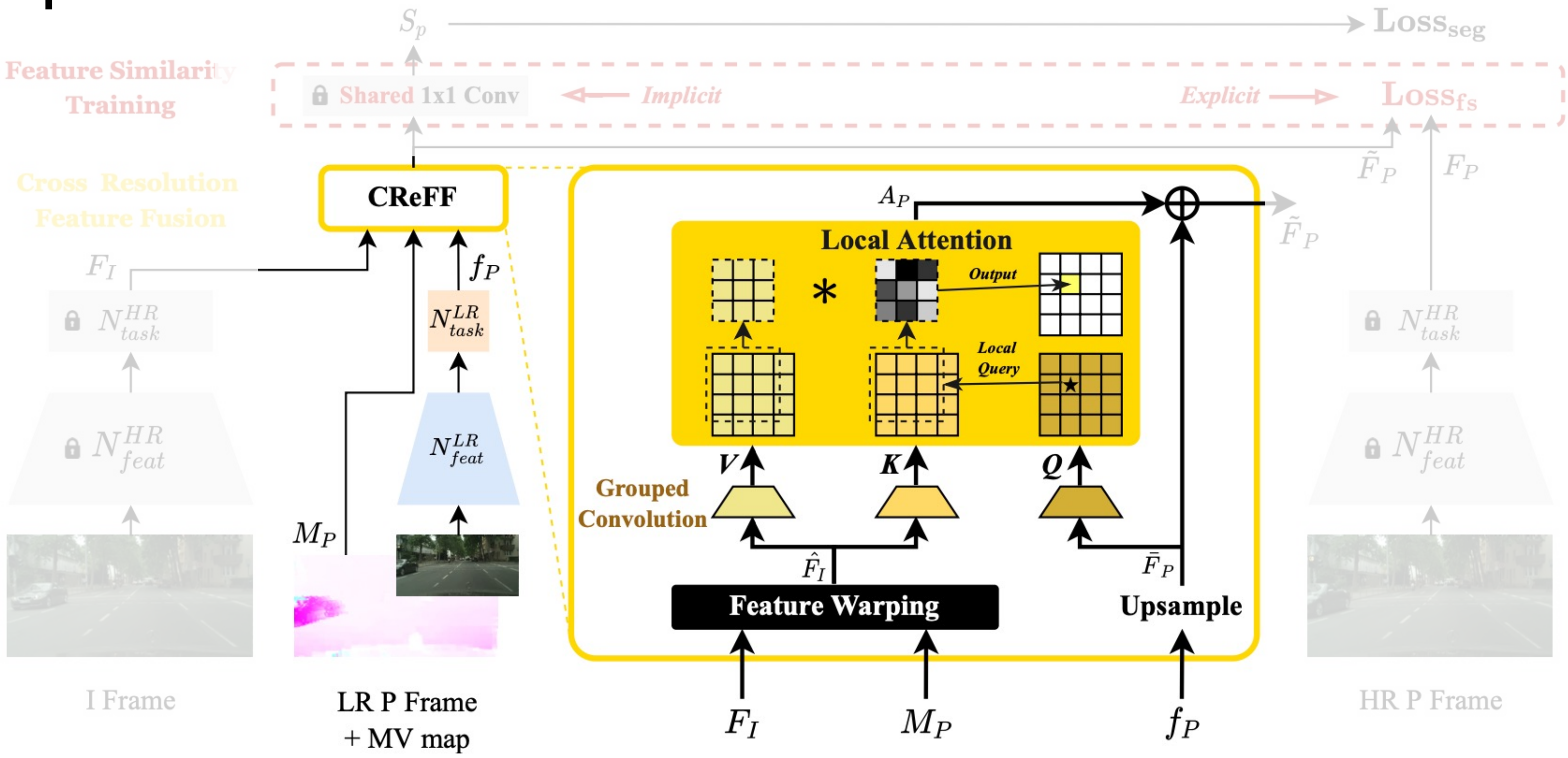
Overview

- Video semantic segmentation (VSS) is a **computationally expensive** task.
  - We propose to reduce the GLOPs by **altering the input resolution**.
  - The presented AR-Seg **reduces 60% GFLOPs while maintaining the accuracy**.

	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**

Overview

- Video semantic segmentation (VSS) is a **computationally expensive** task.
  - We propose to reduce the GLOPs by **altering the input resolution**.
  - The presented AR-Seg **reduces 60% GFLOPs while maintaining the accuracy**.
  - Our utilization of **motion vectors** can be adopted to **other applications related to compressed videos**.

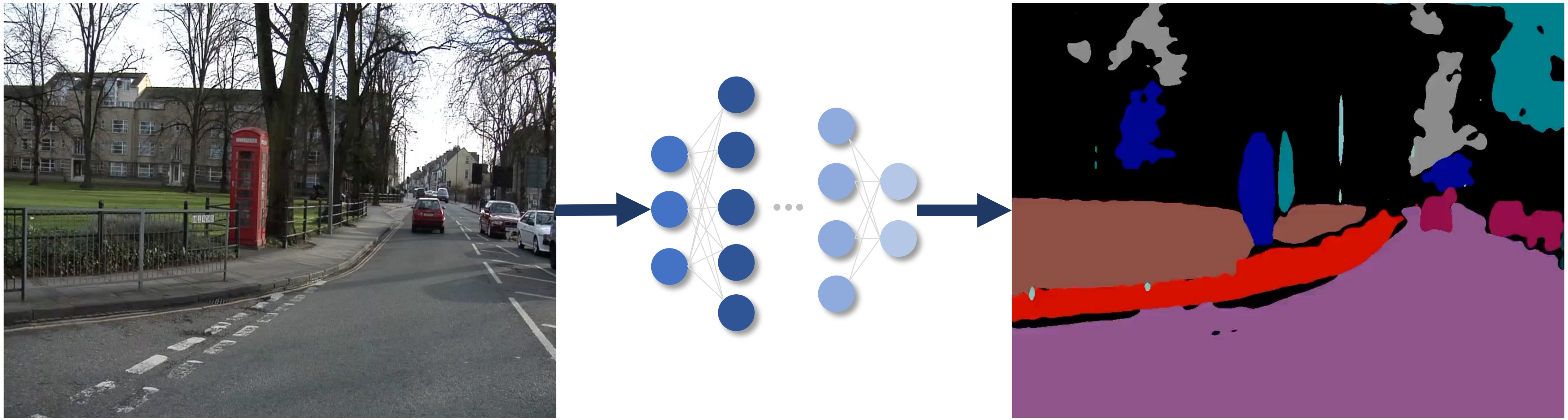


Efficient Semantic Segmentation by Altering Resolutions for Compressed Videos

More Details

Background

- Video semantic segmentation (VSS) is a computationally expensive task.
  - Applying image-based models is expensive.
  - Compact image-based or temporally varying models are proposed.
  - **What about improving efficiency from the input side ?**



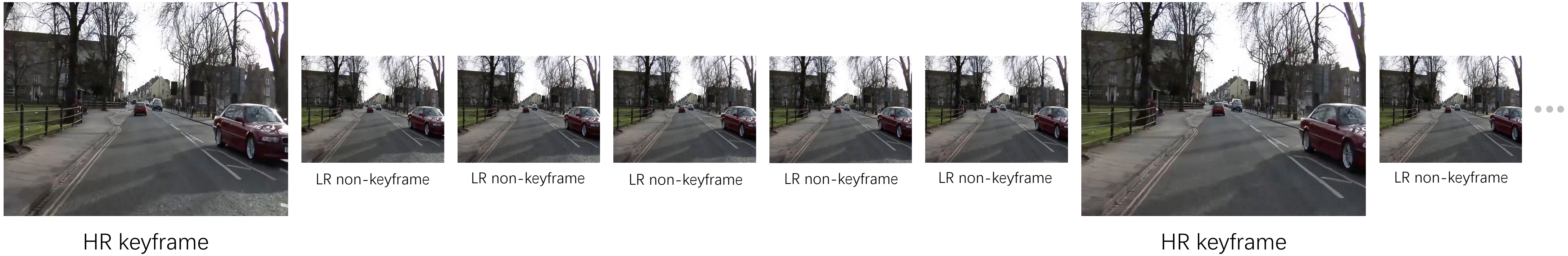
Input

Segmentation results of *PSPNet (ResNet-18).*

Motivation

- They ignored a crucial factor from the input side: **the input resolution**.
  - **The input resolution** determines the amount of computation for image-related tasks.
  - E.g. 0.5x0.5 down-sampling reduces the cost of convolution by 75%.

- Process keyframes in **high-resolution** and non-keyframes in **low-resolution**.
  - With temporal correlation, <u>the performance drop in LR frames can be mitigated by HR frames</u>.



Motivation

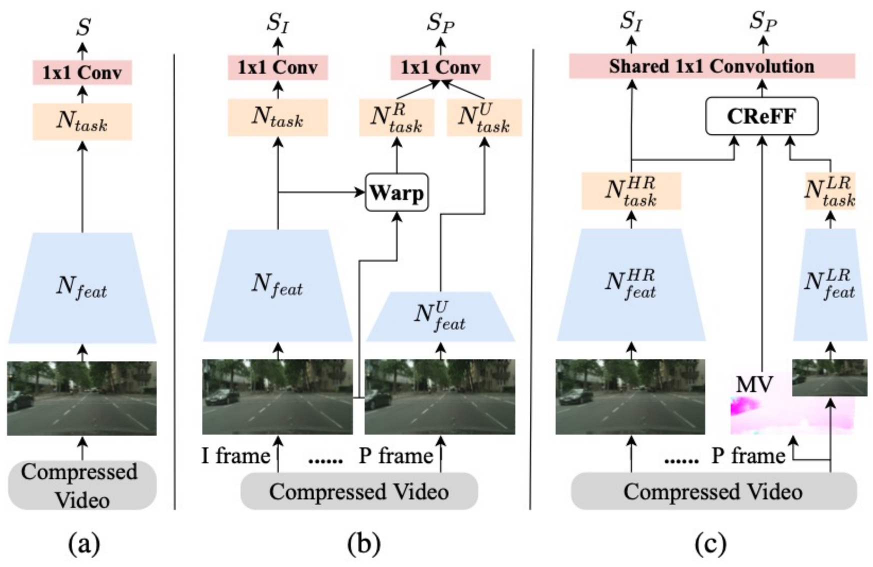
- How to use the temporal correlation and improve the accuracy of LR frames?
  - Aggregate the HR features into LR frames. 😏
  - **Spatial misalignment** for frames at different timesteps. 🤨
  - Guide the feature aggregation with some **motion cues**. 🤔
    - Optical flow can provide such motion cues. **But expensive.** 😤

    - Most videos are compressed by video encoders, e.g. H.264, H.265, AV1.
    - Motion vectors in the compressed videos can also provide such motion cues. **With almost no cost.** 🤩

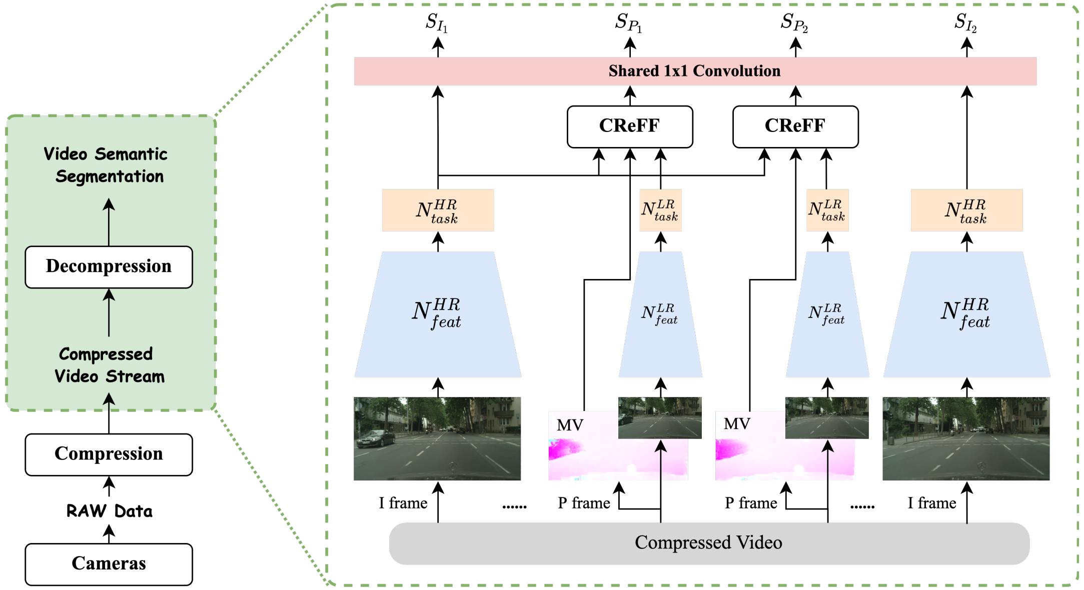
Keyframe   Non-keyframe   Motion vectors   Non-keyframe   Motion vectors

Method

- We propose **an efficient framework**, AR-Seg, for VSS of compressed videos.
  - It alters the input resolution of video frames to reduce the computational cost. 🤓
  - And maintains the overall segmentation accuracy. 😎



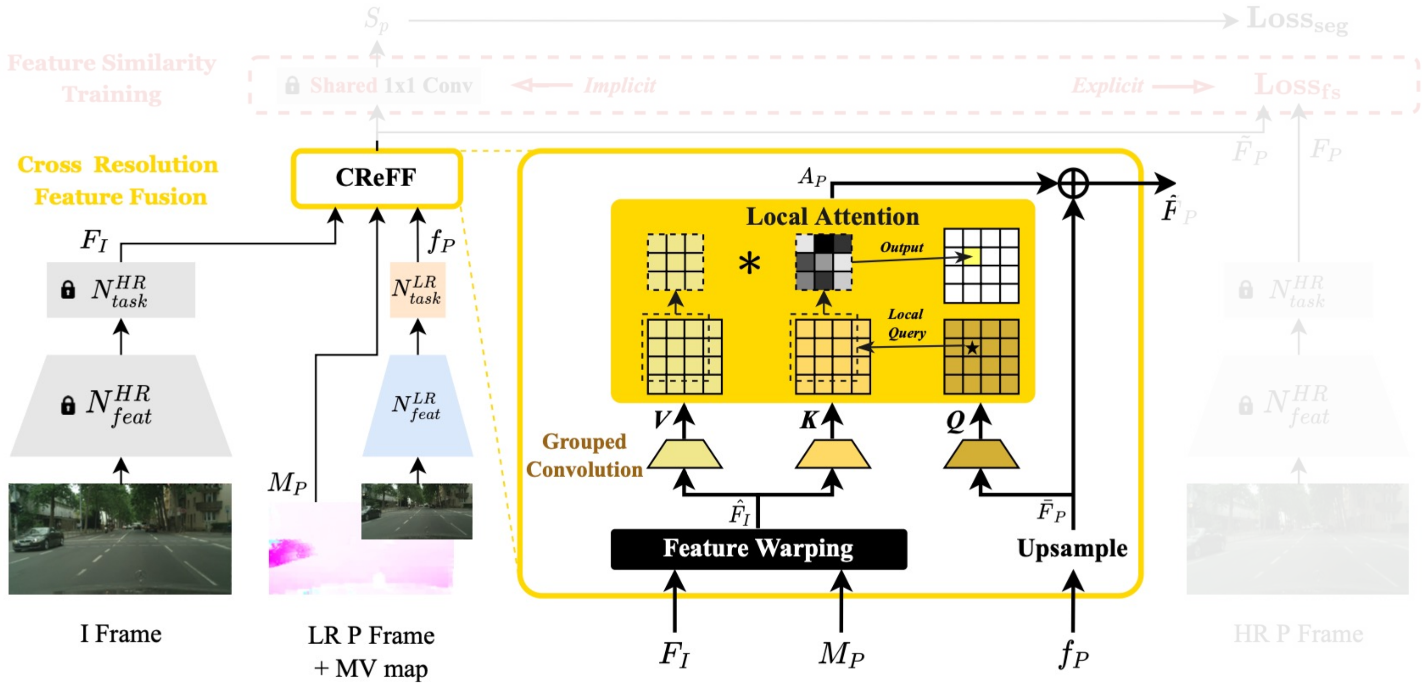
Comparison between AR-Seg and existing methods.



The overall pipeline of AR-Seg.

Method

- The proposed cross resolution feature fusion (CReFF) module.
  - Fuse the information inside HR features into the LR branch.
1. Warp the HR feature using motion vectors.
2. Aggregate the warped features with local attention mechanism.

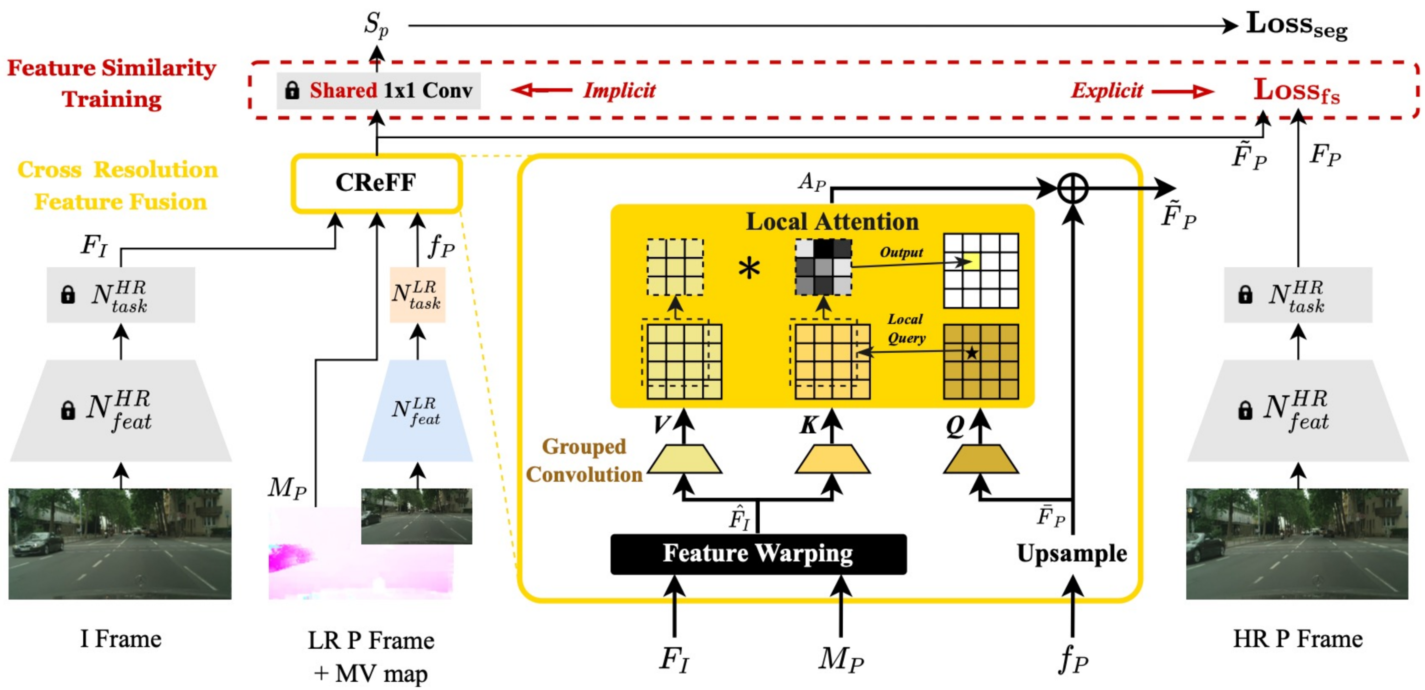


The cross resolution feature fusion (CReFF) module.

Method

- The proposed feature similarity training (FST) strategy.
  - Guide the aggregated features in the LR branch.

1. An **explicit** constraint: feature similarity loss.
2. An **implicit** constraint: the shared decoding layer.



The CReFF module in the network architecture and feature similarity training (FST) strategy.

Experiments

- Comparison with image-based methods. L=12.

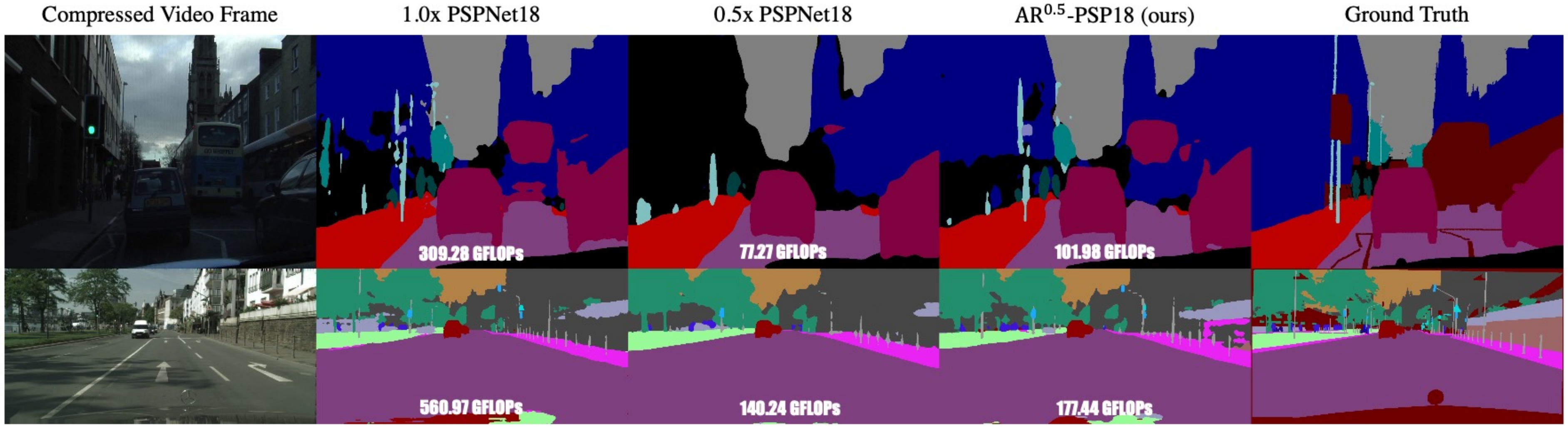


Table 1. Comparison to the image-based methods on CamVid *test* set and Cityscapes *valid* set.

	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 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)

Experiments

- Comparison with video-based methods.
  - AR-Seg is the only method that saves computation and maintains accuracy.
  - $\tilde{\Delta} GFLOPs \leq 0\ \&\ \tilde{\Delta} mIoU \geq 0$

		Single-frame baseline			Video approach			
	Method	Backbone	mIoU(%)↑	GFLOPs ↓	mIoU(%)↑	GFLOPs ↓	$\tilde{\Delta}$mIoU↑	$\tilde{\Delta}$GFLOPs ↓
CamVid	Accel-DL18 [10]	DeepLab18 [3]	58.13	245.65	66.15	397.70	**+13.8%**	+61.9%
	TD^4-PSP18 [8]	PSPNet18 [23]	69.43	309.02	70.13	363.70	+1.0%	+17.7%
	BlockCopy [18]	SwiftNet-RN50 [13]	70.41	215.90	66.75	107.52	-5.2%	-45.7%
	TapLab-BL2 [6]	MobileNetV2 [16]	69.93	236.40	67.57	117.73	-3.1%	-50.2%
	Jain et al. [9]	DeepLab50 [3]	70.65	318.12	67.61	146.97	-4.3%	-53.8%
	$AR^{0.6}$-PSP18	PSPNet18 [23]	69.43	309.02	70.82	101.98	+2.0%	-57.0%
	$AR^{0.6}$-Bise18	BiseNet18 [22]	**71.57**	**58.83**	**71.60**	**24.68**	+0.0%	**-58.0%**
Cityscapes	Accel-DL18 [10]	DeepLab18 [3]	57.64	516.20	68.25	1011.75	**+18.4%**	+96.0%
	TD^4-PSP18 [8]	PSPNet18 [23]	69.00	560.97	70.11	673.06	+1.6%	+20.0%
	BlockCopy [18]	SwiftNet-RN50 [13]	**72.47**	500.35	67.69	294.20	-6.7%	-41.2%
	TapLab-BL2 [6]	MobileNetV2 [16]	71.85	480.34	68.90	237.29	-4.1%	-50.6%
	Jain et al. [9]	DeepLab50 [3]	72.26	721.41	68.57	342.67	-5.1%	-52.5%
	$AR^{0.6}$-PSP18	PSPNet18 [23]	69.00	560.97	69.45	234.91	+0.7%	**-58.1%**
	$AR^{0.6}$-Bise18	BiseNet18 [22]	70.09	**178.96**	**70.72**	**76.06**	+0.9%	-57.5%

Experiments

- The design of CReFF and FST, and the keyframe interval.

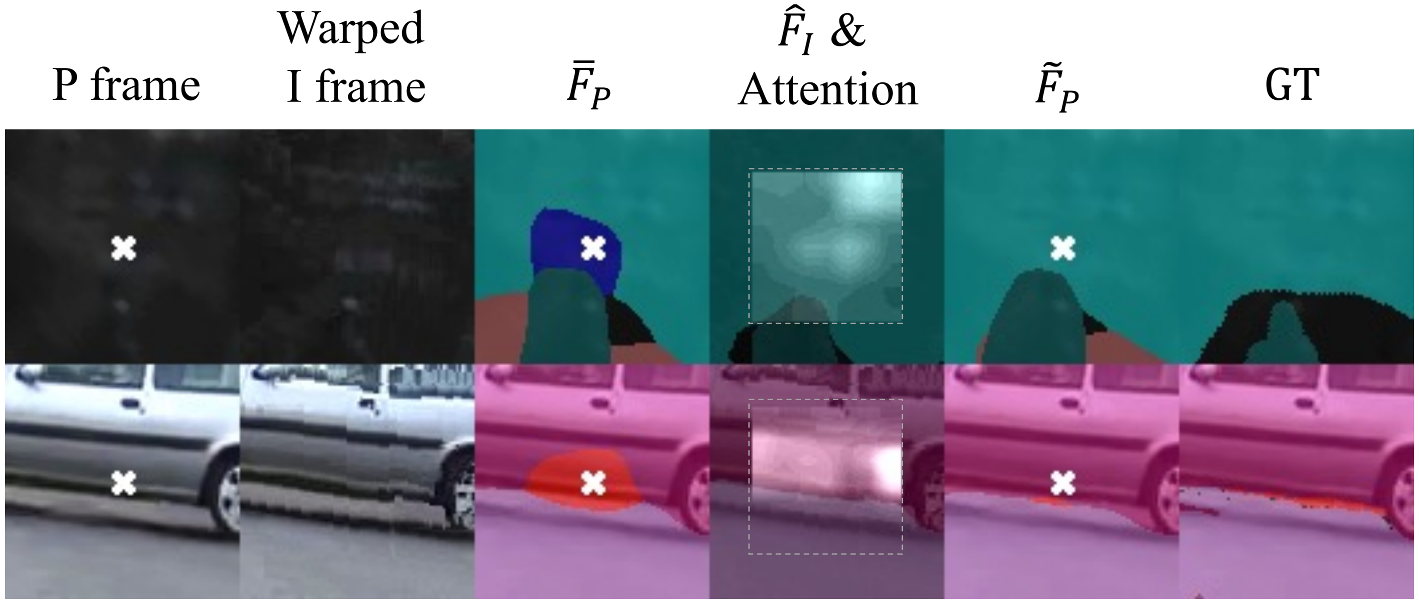
Experiment	Method	mIoU(%)	GFLOPs
Baseline	PSPNet18 (1.0x)	69.43	309.02
	PSPNet18 (0.5x)	66.51	77.27
Architecture of CReFF	$\underline{+ \mathcal{W}_{MV} + \mathcal{F}_{LA}\ (7\text{x}7)}$	**70.48**	**101.98**
	w/o CReFF	67.14	96.60
	$+ \mathcal{W}_{MV}$	57.64	25.75
	$+ \mathcal{F}_{LA}$ (7x7)	67.93	101.98
	$+ \mathcal{W}_{MV} + \mathcal{F}_{LA}$ (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

Experiment	Method	mIoU(%)	GFLOPs
Baseline	PSPNet18 (1.0x)	69.43	309.02
	PSPNet18 (0.5x)	66.51	77.27
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
	$\text{AR}^{0.5}$-PSP18, L=15	70.28	97.88
	$\text{AR}^{0.5}$-PSP18, L=20	70.28	94.11
	$\text{AR}^{0.5}$-PSP18, L=30	69.67	**90.34**

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

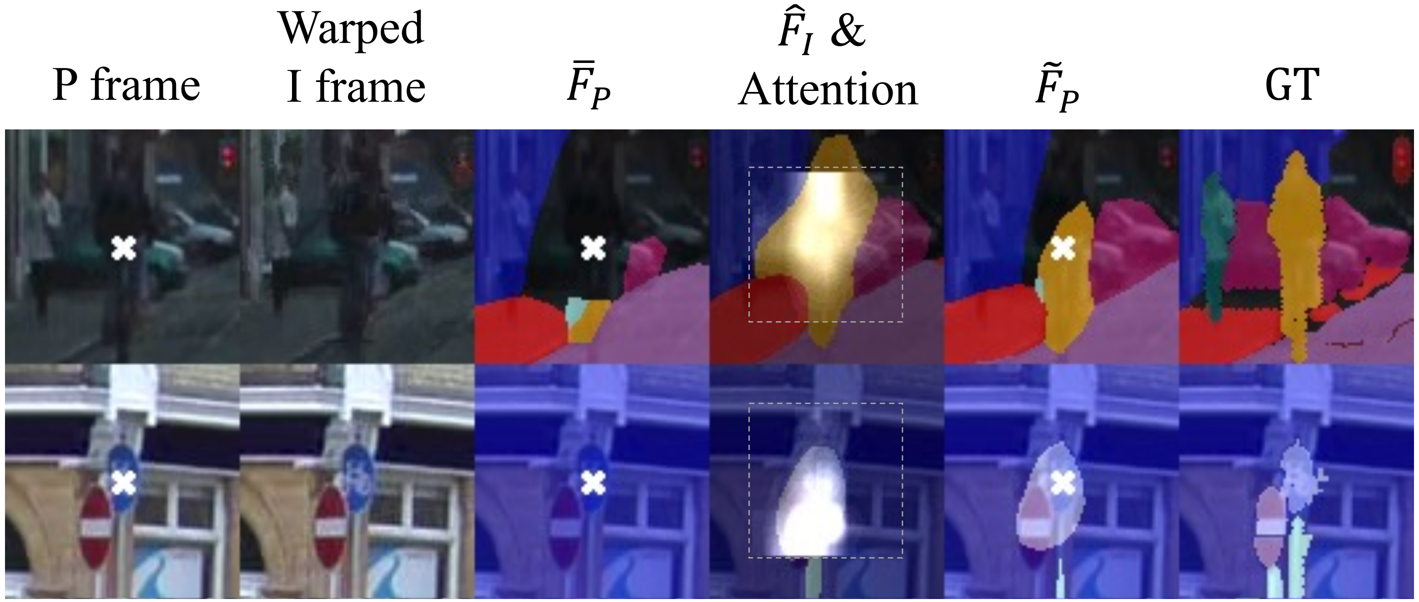
Experiments

- About the local attention mechanism.
  1. It corrects the wrong features in $\bar{F}_P$.



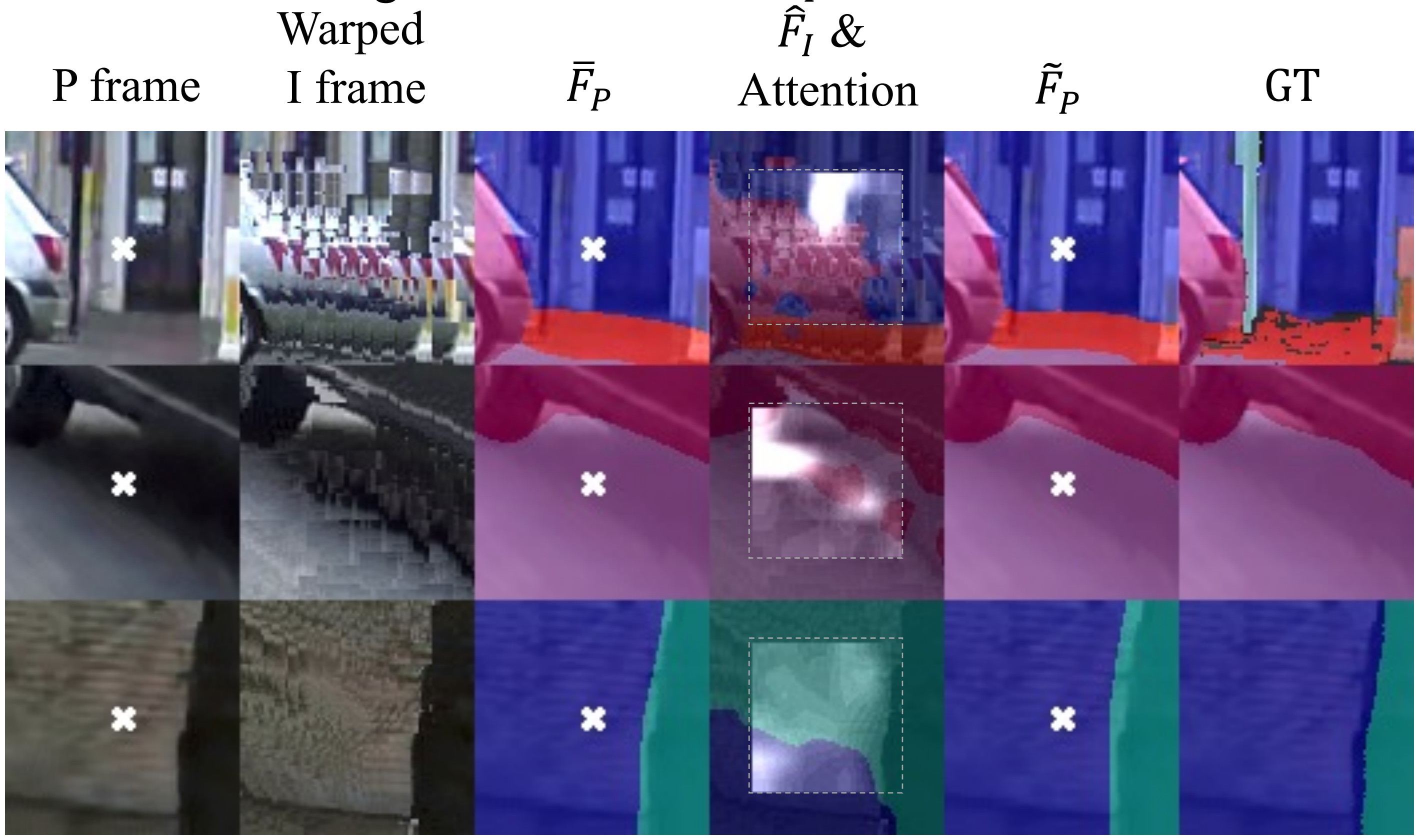
Experiments

- About the local attention mechanism.
  1. It corrects the wrong features in $\bar{F}_P$.
  2. It complements the missing features in $\bar{F}_P$.



Experiments

- About the local attention mechanism.
  1. It corrects the wrong features in $\bar{F}_P$.
  2. It complements the missing features in $\bar{F}_P$.
  3. It resists the misleading features from $\hat{F}_I$.



Future Work

- More adaptive adjustment with more resolution levels.
- Experiments with more segmentation backbones.
- Apply the similar idea to other video-related applications.
    - Object tracking, instance segmentation, etc.
    - Utilize the existing information inside the compressed videos.
- …

Thanks

Paper

arXiv

Code

GitHub