AuxAdapt: Stable and Efficient Test-Time Adaptation for Temporally Consistent Video Semantic Segmentation

Yizhe Zhang [*†]
yizhe.zhang.cs@gmail.com

Shubhankar Borse [‡§]
sborse@qti.qualcomm.com

Hong Cai [‡§]
hongcai@qti.qualcomm.com

Fatih Porikli [‡]
fporikli@qti.qualcomm.com

Abstract

In video segmentation, generating temporally consistent results across frames is as important as achieving frame-wise accuracy. This paper presents an efficient, intuitive, and unsupervised online adaptation method, AuxAdapt, for improving the temporal consistency of most neural network models. It does not require optical flow and only takes one pass of the video. Since inconsistency mainly arises from the model's uncertainty in its output, we propose an adaptation scheme where the model learns from its own segmentation decisions as it streams a video, which allows producing more confident and temporally consistent labeling for similarly-looking pixels across frames. For stability and efficiency, we leverage a small auxiliary segmentation network (AuxNet) to assist with this adaptation. More specifically, AuxNet readjusts the decision of the original segmentation network (Main-Net) by adding its own estimations to that of MainNet. At every frame, only AuxNet is updated via back-propagation while keeping MainNet fixed. We extensively evaluate our test-time adaptation approach on standard video benchmarks, including Cityscapes, CamVid, and KITTI. The results demonstrate that our approach provides label-wise accurate, temporally consistent, and computationally efficient adaptation.

1. Introduction

Recent years have witnessed remarkable progress in image-based semantic segmentation. With the rising popularity and pervasiveness of videos, there is now an increasing focus on video segmentation as a necessary functionality for higher-level computer vision tasks. While it is possible to treat video segmentation as an image segmentation problem and apply image-based models to each frame independently, such segmentations usually lack consistency in time. In other words, image pixels across consecutive video frames that belong to the same semantic class and share similar visual appearances can be labeled differently, resulting in artifacts such as flickering of segmentation. See examples in the 1st and 3rd columns of Fig. 1.

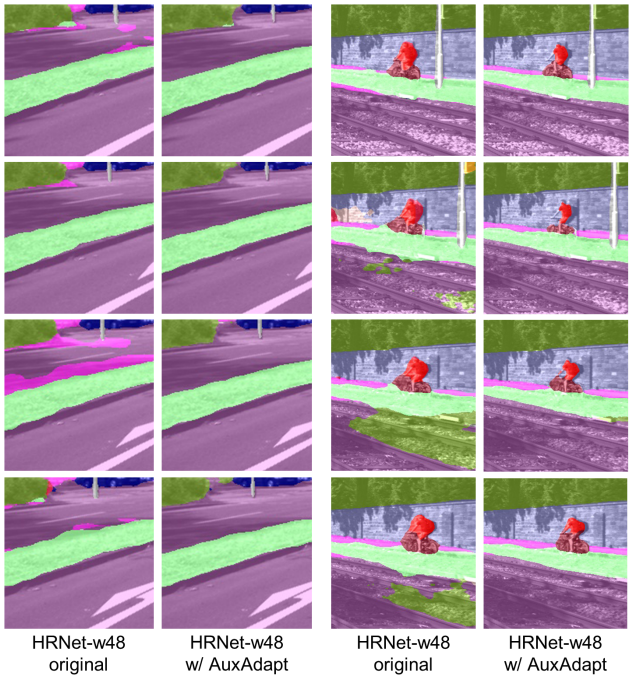

Figure 1: Segmentations of two pairs of consecutive video frames. Columns 1 and 3 show the results of the state-of-the-art HRNet-w48 model [40]. Columns 2 and 4 are obtained by applying our efficient AuxAdapt to HRNet-48. As visible, AuxAdapt improves the temporal consistency significantly.

There have been several attempts to bring temporal consistency in video semantic segmentation. Most methods utilize optical flow to impose consistency as it establishes pixel correspondence across frames. For instance, [9, 11] propose a joint training of segmentation and optical flow, showing that the network can provide more consistent segmentation by learning both tasks together. Other methods incorporate pretrained optical flow as an additional channel of information to the network [31] for regularization during training [27]. While optical flow is commonly employed, there are drawbacks. First, inferring accurate optical flow maps is challenging; thus, such a dependency limits the performance of segmentation, often causing additional is-

[*]Work done at Qualcomm AI Research.
[†]Nanjing University of Science and Technology, Nanjing, China
[‡]Qualcomm AI Research, an initiative of Qualcomm Technologies, Inc.
[§]Equal contribution.

sues. Moreover, existing joint optical flow and segmentation methods need accurately annotated datasets. Besides, except for [27], all these methods require the optical flow during inference, which creates computational overheads.

Some previous approaches incorporate test-time model adaptation to enhance temporal consistency [4, 24, 30, 42]. Nevertheless, most of these methods require dense correspondence through motion estimation or patch matching [3]. A few works that do not use explicit optical flow computation, e.g., [12, 17], still depend on feature conformity across multiple frames. In their pioneering work, Lei et al. [25] have proposed Deep Video Prior (DVP), which can do away with the optical flow computation by leveraging how a deep neural network learns. Given a test video and the processed frames, DVP trains a network from scratch to generate these processed frames and employs early stopping to prevent the network from overfitting to temporally-inconsistent patterns. While DVP can provide adaptation, it demands training the original network model for more than 20 epochs on each test video, resulting in a drastic computational cost.

Our paper proposes an efficient unsupervised online (test-time) adaptation method to promote temporal consistency of video semantic segmentation. Our approach applies to any existing segmentation models. On a high level, we enable the network to efficiently learn from its own decisions without a dependency on optical flow or other temporal features either in training or test. The network adapts based on the (hard) labeling decisions it makes for the pixels of the current frame while it sequentially segments video frames. Our observation is that consecutive frames share similar visual content; therefore, by reinforcing the network to bolster on its previous responses, it is possible to obtain consistent responses for similar regions in future frames. Our intuition here is that we do not need to update the entire network to accomplish such a reinforcement. Instead, we incorporate a tiny auxiliary network (AuxNet) to steer and assist with the adaptation, as illustrated in Fig. 2 (right). We refer to our method as **AuxAdapt**.

Since this AuxNet can be easily trained with only one backward pass in back-propagation on a single frame, our method is very efficient. In comparison, existing adaptation methods such as DVP require training on a large set of frames for significantly more epochs. AuxNet retains a small architecture and works in lower spatial resolution and lower frame rate. It is trained for the same semantic segmentation task as the main segmentation network (MainNet), to which it provides test-time adaptation. During inference for each frame, MainNet is frozen, and only AuxNet is updated as the integrated model streams through the video. The final segmentation is determined by the "aggregated" outputs of MainNet and AuxNet. In this way, the segmentation model adapts continuously to a given video with only a fraction of the computational cost of DVP since only AuxNet gets updated instead of the entire MainNet.

Furthermore, our proposed AuxAdapt maintains the segmentation accuracy while improving temporal consistency online. As MainNet is kept intact and contributes to the aggregated output, we prevent catastrophic forgetting and limit the variation of accuracy. Although AuxNet employs a smaller architecture and usually does not provide as high of a standalone accuracy as MainNet, training on the aggregated outputs allows it to distill knowledge from MainNet during the adaptation. As we shall see in the experiments, even when AuxNet's accuracy is a few percentage points lower than that of MainNet (in terms of mIoU), the overall segmentation accuracy is maintained. In some cases, the overall accuracy even improves slightly due to the ensemble effect of the two networks. In contrast, the state-of-the-art DVP trains a new network for a specific test video, and their model does not provide accurate segmentation for other test data without expensive retraining. Moreover, DVP relies on early stopping to a trade-off between temporal consistency and accuracy on the test video, but does not provide a clear criterion for early stopping in test time.

We summarize our main contributions as follows:

- We present an unsupervised online adaptation method, AuxAdapt, to boost temporal consistency of video semantic segmentation. Our approach does not require optical flow (or cross-frame features). It can be appended to any segmentation network.
- We introduce a novel tiny auxiliary network (AuxNet) that accompanies the selected main segmentation network (MainNet). The final segmentation is obtained by fusing these two networks' outputs. Only AuxNet is updated at each frame with only one epoch during test time, which significantly reduces the computational load. AuxNet can take advantage of temporal and spatial sub-sampling for further acceleration.
- We leverage on a simple change-detection-based adaptive momentum when performing the online adaptation, which adjusts the momentum coefficient based on the difference of two consecutive frames. We show that this provides a good balance between temporal consistency and segmentation accuracy.
- We conduct extensive experiments on Cityscapes, Camvid, and KITTI with comprehensive ablation studies that demonstrate the efficacy of our approach.

2. Related Work

Video Semantic Segmentation: Since the introduction of fully convolutional networks (FCNs) [28], various network architectures have been developed to improve the segmentation accuracy as well as efficiency, e.g., U-Net [34], PSPNet [43], DeepLab [7], and HRNet [40], to count a few. Applying an image-based semantic segmentation network for

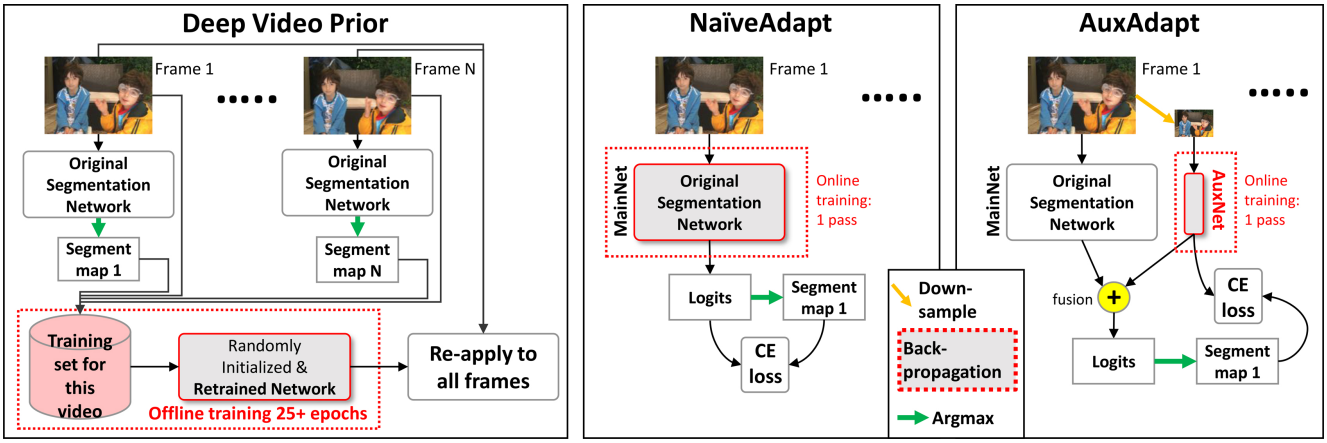

Figure 2: **Left:** Deep Video Prior (DVP) [25] first applies the original network to all video frames. The outputs, together with the corresponding inputs, are collected to form a training set. A new network is then trained based on this set using at least 25 epochs. Finally, the retrained network is applied to the same video to obtain the final outputs. This means DVP applies twice inference over the entire set of frames and a full-scale, computationally expensive retraining. **Middle:** NaiveAdapt: the main segmentation network is being updated when processing through the video frames. **Right:** AuxAdapt uses a tiny auxiliary network on lower spatial resolution together with the main segmentation network. It only updates AuxNet. This has several advantages (see Introduction). Note that AuxNet estimates are fused with MainNet *before* cross-entropy loss with softmax (CE) [1]. Logits is a $H \times W \times K$ tensor.

videos in a per-frame fashion is a common choice in practice due to its simplicity. Although recent state-of-the-art segmentation networks (e.g., HRNet) can achieve impressive results on individual images, they often induce temporally inconsistent results when applied to video frames [27].

Temporal Consistency: There are numerous studies on improving the temporal consistency of video processing tasks such as colorization, dehazing, and segmentation [24, 25, 30]. More related to our paper are those that study temporal consistency of video semantic segmentation. In the existing literature, researchers have proposed incorporating different information during training to improve segmentation temporal consistency, e.g., optical flow [19, 31], 3D structure [13, 22], and/or utilizing more complex models, e.g., Recurrent Neural Network (RNN) [33, 36], Conditional Random Field (CRF) [23], non-local attention [17]. However, these methods either require accurate pixel-correspondence across frames, which is not always available or reliable, or utilize more complicated models (e.g., attention), which introduce significant computational overhead. Recently, Liu et al. [27] have proposed incorporating optical flow only during training and utilizing distillation to derive smaller networks, making inference more efficient. However, their method still requires video data for training and is constrained by the estimated optical flow's reliability. Some other works exploit the temporal nature of videos to improve accuracy [14, 26, 32, 39, 6, 18] and not for improving consistency.

Test-Time Adaptation: Past studies have also looked at test-time model updates, e.g., Tent [38] for domain adaptation. Online adaptation could be another route for improving temporal consistency. Some works utilize optical flow during the test for adaptation, which requires multi-frame processing and considerable computations [4, 30, 42]. Recently, Lei et al. [25] proposed DVP for single-video test-time adaptation, which has been shown to work well for low-level vision tasks, e.g., colorization. However, DVP requires significant model update effort during test time. Some other works assume that the ground-truth annotations are available for the first frame and consider the adaptation as a few-shot learning problem [29, 37, 41], which is a different setup than what we analyze in our paper.

3. Proposed Method

When a semantic segmentation network is uncertain about its estimates (i.e., the top two classes having numerically similar responses), small input variations could result in contrasting segmentation decisions due to the hard-decision rule (i.e., argmax) applied at the end of the network. An example of this issue can be seen in Fig. 1 (1st and 3rd columns), where similar road pixels are assigned to different labels across time. Such output uncertainty constitutes a major cause of temporal inconsistency in video segmentation.

This section describes the proposed AuxAdapt method that improves temporal consistency by enabling the network to learn from its own hard decisions. Note that soft decisions do not generate a loss and thus are not useful in an unsupervised setting. Online adaptation encourages the network to be more confident in its outputs, leading to a more consistent semantic segmentation across frames.

AuxAdapt employs an auxiliary network (AuxNet) to aid the adaptation. AuxNet is tiny; thus, it has a small computation overhead. Not updating the given main network (MainNet) improves the overall stability and prevents catastrophic forgetting. It also helps to maintain accuracy. Adapting only the AuxNet branch makes the process more flexible, allowing it to apply most segmentation architectures. In addition, AuxNet can utilize intermittent (conditional) updates and spatial sampling to reduce computations further. We also bring in an adaptive momentum scheme to decide how much of the previous information to use for adaptation, based on the difference between two consecutive frames.

3.1. Learning from Network's Own Decisions

Consider a pretrained semantic segmentation network f^{main}, which takes as input an RGB image, $x \in [0, 1]^{H\times W\times 3}$, and generates a response/prediction map, $y^{\text{main}} \in \mathbb{R}^{H\times W\times K}$, for a K-class semantic segmentation task, where H and W are the height and width of the input image, respectively. Given a spatial location (i, j) in the image, $y^{\text{main}}(i, j, k)$ is a score that indicates how likely this pixel belongs to class k. For the segmentation decision, an argmax operation is applied pixel-wise to the last dimension of y^{main}, such that the most probable class is assigned to each pixel. We denote this hard (discrete) decision as $y^{\text{seg}} \in \{1, 2, \ldots, K\}^{H\times W}$. To obtain semantic segmentation for a sequence of video frames $X = \{x_1, \ldots, x_T\}$, f^{main} can be applied to generate $Y = \{y_1^{\text{seg}}, \ldots, y_T^{\text{seg}}\}$, where T is the number of frames in the video.

While applying f^{main} to X can generate video semantic segmentation, the resulting output Y is usually temporally inconsistent, as shown in Fig. 1 (1st and 3rd columns). This is mainly due to the network's uncertainty in its own output, as aforementioned. To improve the temporal consistency, it is necessary to reduce the network's uncertainty. We address this by training the network on its own hard decisions, which reinforces its own belief. This allows the network to generate more confident predictions for image regions that are visually similar to what it has seen before. Our method works in a one-pass fashion, only requiring the network to go through the video once. The adaptation based on the current frame will immediately lead to more temporally consistent segmentation on the next frame, as consecutive frames share similar visual content. Note that existing test-time adaptation methods, e.g., DVP [25], Tent [38], do not take advantage of the sequential property (i.e., consecutive frames are visually similar) and need to re-apply the updated network to the full video again. DVP also requires training at least 25 epochs for adapting to a video.

One possible, straightforward way to implement such an adaptation scheme is to compute the loss between y_t^{main} and y_t^{seg} and update the network on this loss, at each time t. We refer to this adaptation scheme as **NaiveAdapt**, which is illustrated in Fig. 2 (middle). This NaiveAdapt approach, however, has some drawbacks. First, performing backward pass on the entire network is computationally expensive.[1] Also, adapting the segmentation network for a long video can degrade its general segmentation accuracy. Moreover, updating the network can be tricky for architectures that require information from previous frames, such as TDNet [17] and those with recurrent modules.

Algorithm 1: AuxAdapt

Input: $x_1, x_2, \ldots, x_T$;
Output: $y_1^{\text{seg}}, y_2^{\text{seg}}, \ldots, y_T^{\text{seg}}$;
Load trained MainNet f^{main}, which will be frozen;
Load trained AuxNet f^{aux} as f_1^{aux};
Initialize $t = 1$;
while $t \leq T$ **do**
  $y_t^{\text{main}} = f^{\text{main}}(x_t)$,[2] $y_t^{\text{aux}} = f_t^{\text{aux}}(x_t)$;
  $y_t^{\text{seg}}(i, j) = \underset{k}{\text{argmax}}\, y_t^{\text{main}}(i, j, k) + y_t^{\text{aux}}(i, j, k), \forall(i, j)$;
  Compute loss: $\mathcal{L}(y_t^{\text{aux}}, y_t^{\text{seg}})$ using Eq. (2);
  Update f_t^{aux} using Eq. (1), which gives f_{t+1}^{aux};
  $t \leftarrow t + 1$;
end

3.2. Adaptation Using Auxiliary Network

In order to overcome the disadvantages of NaiveAdapt, we propose **AuxAdapt**, which employs a separate auxiliary network (AuxNet), f^{aux}, to work with the main segmentation network (MainNet), f^{main}, during the adaptation process. AuxNet is a separately-trained small-sized segmentation network. When streaming the video, at each time t, MainNet and AuxNet produce their respective prediction maps for the current frame, y_t^{main} and y_t^{aux}. An argmax operation is then applied to the summation of these two maps to obtain the discrete semantic segmentation decision, y_t^{seg}. Unlike NaiveAdapt, in AuxAdapt, only the AuxNet is updated based on y_t^{seg} while MainNet is kept frozen. AuxNet is updated using gradient descent as follows:

$$\begin{aligned} \Delta\theta_t^{\text{aux}} &= \beta\Delta\theta_{t-1}^{\text{aux}} + \alpha\nabla_{\theta^{\text{aux}}}\mathcal{L}(y_t^{\text{aux}}, y_t^{\text{seg}}), \\ \theta_t^{\text{aux}} &= \theta_{t-1}^{\text{aux}} + \Delta\theta_t^{\text{aux}}, \end{aligned} \tag{1}$$

where θ^{aux} denotes the parameters of AuxNet, α is the learning rate, β is a momentum coefficient that controls the contribution of past gradients. $\mathcal{L}$ is the loss function:

$$\mathcal{L}(y_t^{\text{aux}}, y_t^{\text{seg}}) = \sum_{i=1}^{H}\sum_{j=1}^{W} \frac{\mathcal{L}_{\text{CE}}(y_t^{\text{aux}}(i, j), y_t^{\text{seg}}(i, j))}{HW}, \tag{2}$$

[1] An alternative is to update the last layer(s) of the network, however, this does not provide meaningful improvements, as we shall see in the experiments in Sec. 4.3.

[2] While our algorithm description assumes that MainNet operates on each individual frame, AuxAdapt is compatible with those that utilize information from multiple frames (e.g., [17]): $y_t^{\text{main}} = f^{\text{main}}(x_t, x_{t-1}, ..., x_1)$, as we shall see in Sec. 4.3.

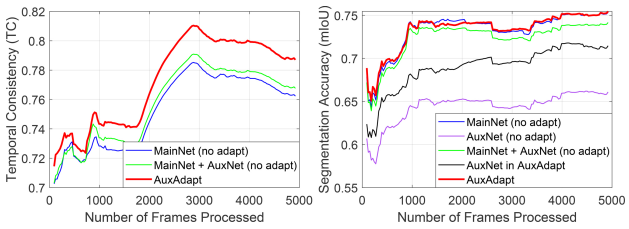

Figure 3: Adaptation dynamics of AuxAdapt on a CamVid test video. AuxAdapt improves temporal consistency (left) and maintains overall accuracy (right) in online adaptation.

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss (with softmax) [1].

Since the updates are performed as the network consumes each frame of the video, the batch size is 1. As such, the mean and standard deviation of batch normalization layers are fixed during this process.

During this process, although MainNet is frozen, the adaptation is enabled by combining the output of the adaptable AuxNet and the output of the MainNet to generate the final discrete segmentation decision for each frame. Then, AuxNet learns from the discrete final decision to reduce the uncertainty in the overall prediction. Our AuxAdapt algorithm is summarized in Algorithm 1.

AuxAdapt offers several advantages. Efficiency-wise, by adopting a small AuxNet, the computation of its forward and backward passes will be much smaller than that of updating the entire MainNet. In addition, we can easily apply AuxAdapt to any semantic segmentation network, as we only need to run forward pass on MainNet and avoid involving the possibly intricate training procedures of MainNet.

Furthermore, AuxAdapt can maintain the segmentation accuracy both on the given test video and on other general test images, thanks to our integrated MainNet-AuxNet model during the adaptation. More specifically, as MainNet is unchanged, we fundamentally prevent catastrophic forgetting. In contrast, updating MainNet itself (i.e., NaiveAdapt) can degrade segmentation accuracy, as we shall see in Sec. 4.3. In addition, combining MainNet's and AuxNet's outputs for the overall segmentation has a number of benefits. First, the variation of accuracy is limited due to MainNet's contribution to the overall output. Learning from the aggregated decisions allows AuxNet to distill knowledge from MainNet, which encourages AuxNet to match MainNet's performance. Lastly, the ensemble effect of the two networks also helps maintain (sometimes even improve) the segmentation accuracy.

Figure 3 shows the adaptation dynamics for a sample test video. Fig. 3 (left) shows that AuxAdapt (red) considerably improves the temporal consistency of the original MainNet (blue) while streaming the video. During this process, it can be seen in Fig. 3 (right) that AuxAdapt (red) closely matches the accuracy of the original MainNet (blue). With AuxAdapt, AuxNet improves its accuracy by learning from MainNet during the process (black) as compared to the non-adapted AuxNet (purple).

Remark 1. *While it is desirable for AuxNet to be very efficient, AuxNet should still provide a reasonable standalone segmentation accuracy. This can be achieved by utilizing a state-of-the-art efficient model, e.g., HRNet-w18 [40]. We note that it is not necessary for AuxNet to be as accurate as MainNet. In our experiments, we test various MainNet-AuxNet combinations where the AuxNet is usually a few percentage points less accurate than MainNet, but AuxAdapt is able to maintain the overall segmentation accuracy.*

3.3. Reducing Computation Further

In addition to utilizing an AuxNet, it is possible to further improve the efficiency of the adaptation process, by exploiting the redundancies in the temporal and spatial domains.

Intermittent Adaptation: The consecutive frames in a video are usually highly similar, especially for a video with a large frame rate. As such, we do not need to update the network for every single frame and can instead perform the model update every several frames. As we shall show in our experiments, performing such intermittent adaptation further reduces computation while still providing a considerable improvement on temporal consistency.

Confidence-Based Spatial Sampling: When the network is already highly confident in its prediction on a pixel, the loss (i.e., $\mathcal{L}_{\text{CE}}$ for pixel (i, j)) for this spatial location will be very small. Such loss terms will not provide meaningful contributions to updating the model and will instead incur unnecessary computation. As such, we can remove such redundant spatial locations from the overall loss computation by setting a confidence threshold. For each pixel, when the combined segmentation confidence (the highest softmax score among the K classes) is above the given threshold, the corresponding loss will not be included in the model update.

3.4. Motion-Based Adaptive Momentum

How the network adapts should also depend on how fast the video scene evolves. For instance, given a slowly-moving scene, earlier frames can still be relevant and visually similar to the current frames. On the other hand, for a fast-changing scene, older frames can quickly become irrelevant. Momentum provides a mechanism for the network to control how much it should utilize previous information via the discounted gradients associated with earlier frames. A larger momentum (i.e., a larger β in Eq. (1)) allows the network to retain more past information and thus can benefit temporal consistency. This, however, might reduce segmentation accuracy as the network's update is mixed with the decisions on the earlier, less related frames. On the other hand, a smaller momentum makes the network focus more on the current frame and better preserves the original segmentation accuracy but provides less temporal consistency improvement.

To achieve a good balance, we propose using a motion-based adaptive scheme for the momentum coefficient. More specifically, we set $\beta = 1 - \frac{1}{HW}\|x_t - x_{t-1}\|_1$, where $\|\cdot\|_1$ denotes L_1 norm. This sets the strength of the momentum conditioned on how fast the frames change.

4. Experiments

In this section (and also in Supplementary File), we present a comprehensive performance analysis on large benchmark datasets, compare with baselines and the current state of the art, assess the cross-dataset adaptation ability of our method, and evaluate our method under intermittent adaptation and spatial sampling, as well as our adaptive momentum scheme. Finally, we conduct in-depth ablation studies on various aspects of our method.

4.1. Experiment Setup

Datasets: We run extensive performance evaluations of our method on Cityscapes (CS) [10], Camvid [5], and KITTI [15]. For CS, we use the validation set of 500 videos, each with 30 frames of size 1024×2048. For CamVid, we use the test set, which contains two long video sequences with a frame size of 720×960. To ensure a fair comparison with the latest video semantic segmentation method [27], we use their setup for evaluating temporal consistency.

We use KITTI for cross-dataset adaptation experiments, i.e., the networks trained on CS are used for test-time adaptation on KITTI. To accommodate the class number and class label assignment difference between CS and KITTI, we map the 7 overlapping classes from KITTI to CS and treat all the other classes from both datasets as a separate class named "others". We utilize raw KITTI data for the test-time adaptation task, containing 60 videos, capped at 100 frames each. The frame size is 384×1280. For evaluating segmentation accuracy, we use the 200 images that have ground-truth annotations, which are sampled across videos.

Networks: We use several state-of-the-art models as the main segmentation network (MainNet), including HRNet [40], DeepLabV3+ [8] with a ResNet-101 backbone [16], ETC [27] with a PSPNet-18 backbone [43], and TDNet [17] with a PSPNet-18 backbone. For HRNet, we use two versions: HRNet-w48-s4 and HRNet-w18-s4.[3] As for AuxNet, we use HRNet-w18-s8 and HRNet-w16-s8, which are light-weight models. The MainNet-AuxNet combinations are summarized in Table 1.[4]

Hyperparameters: We use a learning rate of 0.0001 for all our experiments, a typical value for model fine-tuning.

[3]The number following "w" indicates the channel multiplier and the number following "s" indicates the up-sampling ratio towards the output.

[4]When no trained AuxNets are available, we can create an AuxNet using a low-resolution copy of MainNet. This provides temporal consistency improvement comparable to that of a separately-trained AuxNet, but can incur slightly more computation as its architecture is dictated by MainNet. Detailed results can be found in the Supplementary File.

Test Set	MainNet	AuxNet
CS, KITTI	HRNet-w48-s4 [40]	HRNet-w18-s8
CS	HRNet-w18-s4 [40]	HRNet-w16-s8
CS	DeepLabV3+ [8] (RN-101)	HRNet-w18-s8
CS	ETC [27] (PSP-18)	HRNet-w18-s8
CS	TD4 [17] (PSP-18)	HRNet-w16-s8
CamVid	HRNet-w18-s4 [40]	HRNet-w16-s8
CamVid	PSPNet-101 [43]	HRNet-w18-s8
CamVid	WideResNet-38 [45]	HRNet-w18-s8

Table 1: MainNet-AuxNet combinations used in our main results shown in Tables 2, 3, and 4.

Method	TC	mIoU	GMAC/$_F$
CC [35]	71.2	67.7	-
DFF [44]	71.4	68.7	-
Accel [21]	70.3	72.1	-
HRNet-w18-s8 [40]	71.9	72.6	19
ETC [27] (PSP-101)	71.7	79.5	1731
HRNet-w48-s4 [40]	72.1	81.0	750
w/ DVP [25]	77.4	59.0	56924
w/ Tent [38]	29.8	62.4	3000
w/ AuxAdapt (ours)	75.8	81.0	808
HRNet-w18-s4 [40]	70.5	76.2	78
w/ DVP [25]	76.9	73.6	5898
w/ Tent [38]	55.3	61.3	310
w/ AuxAdapt (ours)	75.3	76.6	128
DeepLabV3+ [8]	71.0	76.2	633
w/ DVP [25]	72.2	76.2	48108
w/ Tent [38]	25.2	59.7	2532
w/ AuxAdapt (ours)	75.1	76.6	691
ETC [27] (PSP-18)	70.6	73.1	463
w/ DVP [25]	76.2	71.5	35188
w/ Tent [38]	57.1	58.6	1852
w/ AuxAdapt (ours)	76.2	75.1	514
TD4 [17] (PSP-18)	71.6	76.8	239
w/ AuxAdapt (ours)	74.5	77.1	290

Table 2: Performance evaluation on Cityscapes validation set.

4.2. Evaluation Metrics

To evaluate segmentation accuracy, we use the standard mean Intersection-over-Union (mIoU). For temporal consistency (TC) evaluations, we use FlowNet2 [20] to compute optical flow between two adjacent frames and warp the segmentation at frame t to frame $t-1$. We then compare the warped and actual segmentations for each frame ($t < T$) using mIoU. The overall mIoU then serves as the TC metric. Note that this is the **same** TC metric used in [27]. We evaluate the computation efficiency of a method based on the MAC count in the forward and backward passes per frame. Notice, the backward pass costs twice as many MACs as the forward pass [2].

4.3. Results

We compare our proposed approach with various state-of-the-art (SOTA) methods, including 1) non-adaptive methods that train and test on single frames [8, 21, 35, 40, 43, 44, 45], 2) non-adaptive methods that utilize multi-frame information (e.g., optical flow, temporal attention) [17, 27], and 3) adaptation methods [25, 38], as well as baseline adaptation methods that update MainNet without using AuxNet. Finally, we present results using our proposed motion-based adaptive momentum.

Evaluation on Cityscapes & Camvid: On Cityscapes, as summarized in Table 2, AuxAdapt significantly improves the temporal consistency of the state-of-the-art models while preserving (and in some cases, improving) the segmentation accuracy. For instance, for MainNet of HRNet-w48-s4, AuxAdapt improves the TC from 72.1 to 75.8, requiring only 7% additional computation. Furthermore, our approach can also improve models that have already utilized multi-frame information, e.g., ETC [27] that utilizes optical flow during training, TDNet [17] that uses temporal attention, as shown in the last two blocks of Table 2.

We see that the SOTA test-time adaptation methods do not provide reliable performance and cost much more computation. For instance, DVP [25] incurs an unfavorable trade-off between temporal consistency and accuracy. It trains, from scratch, a new network to mimic the original network and requires early stopping to prevent overfitting to temporally inconsistent patterns. However, stopping too early leads to a large performance gap between the original and new models. As such, when DVP delivers a TC similar to ours, it considerably reduces segmentation accuracy, and when it attains the original accuracy, its TC improvement is minimal. Moreover, as DVP requires training a new network for 25 epochs (default setting), its computation is prohibitively high for test time (46–70$\times$ higher than AuxAdapt). As for Tent [38], it significantly degrades both TC and accuracy. Tent uses entropy minimization to update the batch normalization (BN) layers of the original network, which severely limits the adaptation capability. Furthermore, to update the BN layers, it is required to propagate through the entire network.

On CamVid, as shown in Table 3, AuxAdapt consistently improves temporal consistency and maintains the segmentation accuracy for a longer test video. In contrast, DVP degrades in both TC and accuracy, as the default 25 epochs are not sufficient for the larger test data, and there is no guideline on how to increase the training time w.r.t. the video length properly. Similarly, Tent does not perform well as the much longer test video makes it challenging for training. Note that CamVid videos are much longer (6000+ frames).

Cross-Dataset Adaptation: In practice, the test data can have different characteristics than the training data, e.g., scenes and camera settings. Here, we evaluate AuxAdapt for such a scenario by adapting a Cityscapes-trained model to KITTI data. Note that the segmentation accuracy is evaluated on KITTI based on the 200 images (from different videos!) with ground-truth annotations. Such an evaluation will assess whether the adapted model maintains a general segmentation capability beyond the one specific test video.

Method	TC	mIoU	GMAC/$_F$
DFF [44]	78.0	66.0	–
Accel [21]	76.2	66.7	–
HRNet-w18-s4 [40]	75.8	73.2	26
w/ DVP [25]	56.6	71.4	1946
w/ Tent [38]	59.1	29.2	102
w/ AuxAdapt (ours)	79.1	73.2	42
PSPNet-101 [43]	76.7	76.2	691
w/ DVP [25]	51.3	72.5	52546
w/ Tent [38]	20.6	28.3	2765
w/ AuxAdapt (ours)	79.5	76.4	711
WideResNet-38 [45]	78.1	80.6	1920
w/ AuxAdapt (ours)	79.4	80.8	1995

Table 3: Performance evaluation on Camvid test set.

Method	TC	mIoU	GMAC/$_F$
HRNet-w48-s4 [40]	57.4	65.9	176
DVP [25]	64.1	27.2	13361
w/ AuxAdapt (ours)	63.5	65.8	189

Table 4: Cross-dataset adaptation from Cityscapes to KITTI.

The results are summarized in Table 4. First, it can be seen that the performance of MainNet (without adaptation) drops considerably as compared to that on Cityscapes, due to the difference between KITTI and Cityscapes. By applying AuxAdapt, the TC significantly improves from 57.4 to 63.5. We also compare with DVP. While DVP improves temporal consistency, it incurs significantly heavier computation, and the obtained model obtained cannot provide acceptable segmentation accuracy on images outside the given test video. In other words, the DVP-trained model cannot be applied to a new test video without incurring another expensive training session. On the other hand, AuxAdapt can readily provide accurate segmentation on new test data, without the need of further adapting.

Comparing with Baseline Adaptation Schemes: We compare AuxAdapt with the NaiveAdapt baseline on Cityscapes and KITTI. For NaiveAdapt, we consider two options: 1) updating only the last part,[5] and 2) updating all the layers of MainNet. As shown in Table 5, only updating the last part of MainNet does not result in meaningful TC gain, while updating the entire MainNet produces less or similar TC gain but incurs much higher computation. Note that for cross-dataset adaptation on KITTI, updating the entire MainNet consid-

[5]The last part of HRNet consists of two conv. layers and a BN layer.

Method	TC	mIoU	GMAC/$_F$
Cityscapes			
HRNet-w48-s4	72.1	81.0	750
w/ NaiveAdapt (Last Part)	72.3	80.5	889
w/ NaiveAdapt (All Layers)	76.2	80.9	2249
+ AuxNet (HRNet-w18-s8)	73.3	80.9	768
w/ AuxAdapt (ours)	75.8	81.0	808
HRNet-w18-s4	70.5	76.2	78
w/ NaiveAdapt (Last Part)	70.7	76.2	109
w/ NaiveAdapt (All Layers)	74.1	76.6	233
+ AuxNet (HRNet-w16-s8)	72.9	76.5	95
w/ AuxAdapt (ours)	75.3	76.6	128
KITTI			
HRNet-w48-s4	57.4	65.9	176
w/ NaiveAdapt (Last Part)	57.3	65.7	208
w/ NaiveAdapt (All Layers)	62.3	62.5	527
+ AuxNet (HRNet-w18-s8)	58.8	65.2	180
w/ AuxAdapt (ours)	63.5	65.8	189

Table 5: Comparison with baseline adaptation schemes, including two options of NaiveAdapt which updates the last part and all layers of MainNet, respectively (without AuxNet). We also report the performance of the integrated MainNet-AuxNet model, without model update (4th row of each block).

erably reduces accuracy due to overfitting to the given test video. As for the case of adding MainNet's and AuxNet's outputs without adaptation, TC increases slightly but is considerably lower than AuxAdapt, indicating that adaptation is key to improving TC.

Intermittent Adaptation: As shown in Table 6, under intermittent adaptation, AuxAdapt still considerably improves TC and maintains the original accuracy, using even lower computation. Note that on KITTI, as the update happens less frequently, TC improvement drops slightly faster as compared to on Cityscapes. This is because the networks are more uncertain on KITTI as they are trained on Cityscapes and the scenes in KITTI evolve more quickly. In spite of these challenging factors, AuxAdapt is still able to considerably improve TC with intermittent adaptation.

Confidence-Based Spatial Sampling: As shown in Table 7, by using confidence-based sampling, we significantly reduce the number of pixel locations included in the loss computation, which further enhances efficiency. Meanwhile, we achieve similar TC improvements. On KITTI, we notice a minor drop in mIoU when applying the spatial sampling. This is because the Cityscapes-trained networks are more uncertain on KITTI and excluding the confident pixels can limit AuxNet's learning from MainNet.

Motion-Based Adaptive Momentum: In Figure 4, we show that our motion-adaptive momentum automatically finds a good balance between TC and accuracy across datasets, as compared to using a fixed momentum.

Update Frequency	TC	mIoU	GMAC/$_F$
Cityscapes			
No Adaptation	72.1	81.0	750
Every Frame	75.8	81.0	808
Every 2 Frames	75.4	81.0	789
Every 5 Frames	74.8	81.0	777
Every 10 Frames	74.7	80.9	773
KITTI			
No Adaptation	57.4	65.9	176
Every Frame	63.5	65.8	189
Every 2 Frames	62.3	66.2	185
Every 5 Frames	60.6	66.3	182
Every 10 Frames	59.9	66.1	181

Table 6: Intermittent Adaptation. MainNet is HRNet-w48-s4 and AuxNet is HRNet-w18-s8.

Selection Criterion	Cityscapes			KITTI		
	TC	mIoU	%	TC	mIoU	%
None	72.1	81.0	0	57.4	65.9	0
All	75.8	81.0	100	63.5	65.8	100
Conf.<0.9	76.6	81.0	15.0	63.8	64.1	18.0
Conf.<0.8	76.5	81.0	11.4	64.0	63.8	13.9

Table 7: Confidence-based spatial sampling. MainNet is HRNet-w48-s4 and AuxNet is HRNet-w18-s8. The 3rd column of each dataset block shows the average percentage of pixel locations included in the model updates. Note that "None" is MainNet only (without adaptation) and "All" is AuxAdapt without sub-sampling.

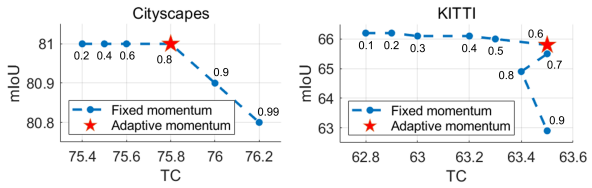

Figure 4: TC and mIoU by using our motion-adaptive momentum coefficient, as compared to fixed momentum coefficients.

5. Conclusions

In this paper, we proposed a novel, unsupervised online adaptation method, AuxAdapt, for improving temporal consistency of video semantic segmentation in test time. AuxAdapt does not require optical flow (or cross-frame features) and can be applied to any segmentation network. By employing a small auxiliary network to work with the original segmentation network, AuxAdapt considerably boosts the temporal consistency of state-of-the-art segmentation models while using a tiny fraction of the computation required by existing adaptation methods. Furthermore, AuxAdapt exploits temporal and spatial sub-sampling for further acceleration, and utilizes an adaptive scheme to automatically set the momentum for adaptation.

References

[1] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.

[2] http://www.cs.toronto.edu/~rgrosse/courses/csc321_2017/slides/lec6.pdf.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24, 2009.

[4] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister. Blind video temporal consistency. *ACM Transactions on Graphics*, 34(6):1–9, 2015.

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

[6] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *European Conference on Computer Vision*, pages 695–714. Springer, 2020.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 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, 2017.

[8] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, 2018.

[9] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H.n Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] M. Ding, Z. Wang, B. Zhou, J. Shi, Z. Lu, and P. Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[12] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, R. Klette, and F. Huang. Stfcn: Spatio-temporal FCN for semantic video segmentation. *arXiv preprint arXiv:1608.05971*, 2016.

[13] G. Floros and B. Leibe. Joint 2D-3D temporally consistent semantic segmentation of street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[14] R. Gadde, V. Jampani, and P. V. Gehler. Semantic video cnns through representation warping. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[15] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[17] P. Hu, F. Caba, O. Wang, Z. Lin, S. Sclaroff, and F. Perazzi. Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[18] Ping Hu, Federico Perazzi, Fabian Caba Heilbron, Oliver Wang, Zhe Lin, Kate Saenko, and Stan Sclaroff. Real-time semantic segmentation with fast attention. *IEEE Robotics and Automation Letters*, 6(1):263–270, 2020.

[19] J. Hur and S. Roth. Joint optical flow and temporally consistent semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2016.

[20] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[21] S. Jain, X. Wang, and J. E. Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[22] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3D reconstruction from monocular video. In *Proceedings of the European Conference on Computer Vision*, 2014.

[23] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[24] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang. Learning blind video temporal consistency. In *Proceedings of the European Conference on Computer Vision*, 2018.

[25] C. Lei, Y. Xing, and Q. Chen. Blind video temporal consistency via deep video prior. In *Advances in Neural Information Processing Systems*, 2020.

[26] J. Li, Y. Zhao, X. He, X. Zhu, and J. Liu. Dynamic warping network for semantic video segmentation. *Complexity*, 2021.

[27] Y. Liu, C. Shen, C. Yu, and J. Wang. Efficient semantic video segmentation with per-frame inference. In *Proceedings of the European Conference on Computer Vision*, 2020.

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[29] T. Meinhardt and L. Leal-Taixé. Make one-shot video object segmentation efficient again. *Advances in Neural Information Processing Systems*, 2020.

[30] O. Miksik, D. Munoz, J. A. Bagnell, and M. Hebert. Efficient temporal consistency for streaming video scene analysis. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.

[31] D. Nilsson and C. Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[32] A. Pasad, A. Gordon, T.-Y. Lin, and A. Angelova. Improving semantic segmentation through spatio-temporal consistency learned from videos. *arXiv preprint arXiv:2004.05324*, 2020.

[33] M. Rebol and P. Knöbelreiter. Frame-to-frame consistent semantic segmentation. *arXiv preprint arXiv:2008.00948*, 2020.

[34] O. Ronneberger, P. Fischer, and T. Brox. U-Net: convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[35] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell. Clockwork convnets for video semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2016.

[36] R. Sibechi, O. Booij, N. Baka, and P. Bloem. Exploiting temporality for semi-supervised video segmentation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.

[37] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017.

[38] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[39] H. Wang, W. Wang, and J. Liu. Temporal memory attention for video semantic segmentation. *arXiv preprint arXiv:2102.08643*, 2021.

[40] J. Wang, K. Sun, T. Cheng, B.i Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[41] H. Xiao, B. Kang, Y. Liu, M. Zhang, and J. Feng. Online meta adaptation for fast video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(5):1205–1217, 2019.

[42] C.-H. Yao, C.-Y. Chang, and S.-Y. Chien. Occlusion-aware video temporal consistency. In *Proceedings of the ACM International Conference on Multimedia*, 2017.

[43] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[44] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[45] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.