Motion-state Alignment for Video Semantic Segmentation

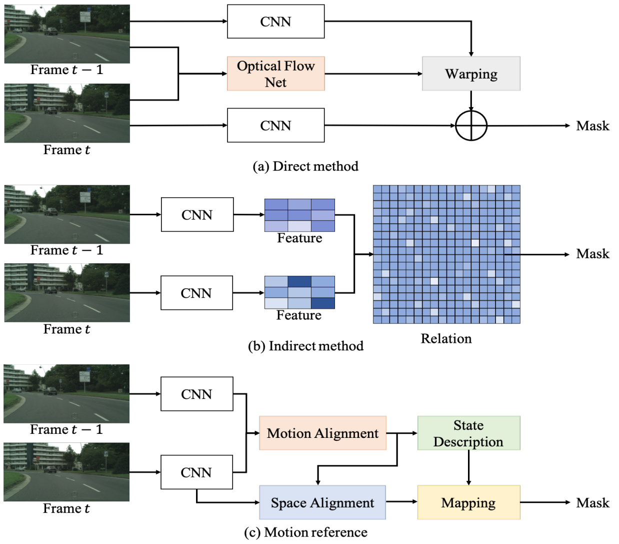
Jinming Su*, Ruihong Yin*, Shuaibin Zhang and Junfeng Luo
Meituan
{sujinming, yinruihong, zhangshuaibin02, luojunfeng}@meituan.com

Abstract

In recent years, video semantic segmentation has made great progress with advanced deep neural networks. However, there still exist two main challenges i.e., *information inconsistency and computation cost. To deal with the two difficulties, we propose a novel motion-state alignment framework for video semantic segmentation to keep both motion and state consistency. In the framework, we first construct a motion alignment branch armed with an efficient decoupled transformer to capture dynamic semantics, guaranteeing region-level temporal consistency. Then, a state alignment branch composed of a stage transformer is designed to enrich feature spaces for the current frame to extract static semantics and achieve pixel-level state consistency. Next, by a semantic assignment mechanism, the region descriptor of each semantic category is gained from dynamic semantics and linked with pixel descriptors from static semantics. Benefiting from the alignment of these two kinds of effective information, the proposed method picks up dynamic and static semantics in a targeted way, so that video semantic regions are consistently segmented to obtain precise locations with low computational complexity. Extensive experiments on Cityscapes and CamVid datasets show that the proposed approach outperforms state-of-the-art methods and validates the effectiveness of the motion-state alignment framework.*

1. Introduction

Semantic segmentation [5, 13, 25], as a dense prediction task, assigns every pixel with the category. Based on image-based semantic segmentation, video semantic segmentation [23, 39, 40] introduces temporal information for each frame, which is very challenging and has attracted lots of attention. Recently, video semantic segmentation has made good progress and been widely applied to autonomous driving [15], video surveillance [19], and other fields.



(a) Direct method

(b) Indirect method

(c) Motion reference

Figure 1. Comparison of different methods for video semantic segmentation. (a) Direct methods explicitly learn the pixel movement depending on the pretrained optical flow network, which leads to information inconsistency. (b) Indirect methods implicitly model the relationship between all pixels with the attention mechanism, which usually results in high computation costs. (c) The proposed motion-state alignment framework decouples the task into motion alignment, state alignment, and semantic assignment, which ensures consistent and efficient learning of semantics.

For video semantic segmentation, the key is how to utilize temporal information to improve the accuracy and consistency of segmentation results across frames. Towards this end, various methods based on deep learning [18, 28, 31] have been proposed to conduct video semantic segmentation recently. From the perspective of feature representation and use, these existing methods can be summarized into two main categories, *i.e.*, direct methods and indirect methods. Direct methods (as shown in Fig. 1 (a)) [12, 39, 40] usually use a separately pretrained optical flow network [11, 17, 29, 34] to propagate information between adjacent frames without fine-tune on semantic segmentation dataset, owing to the lack of optical flow annotations. However, there is a domain shift between the datasets for flow and datasets for segmentation, so this method leads to

[*] Equal contribution.

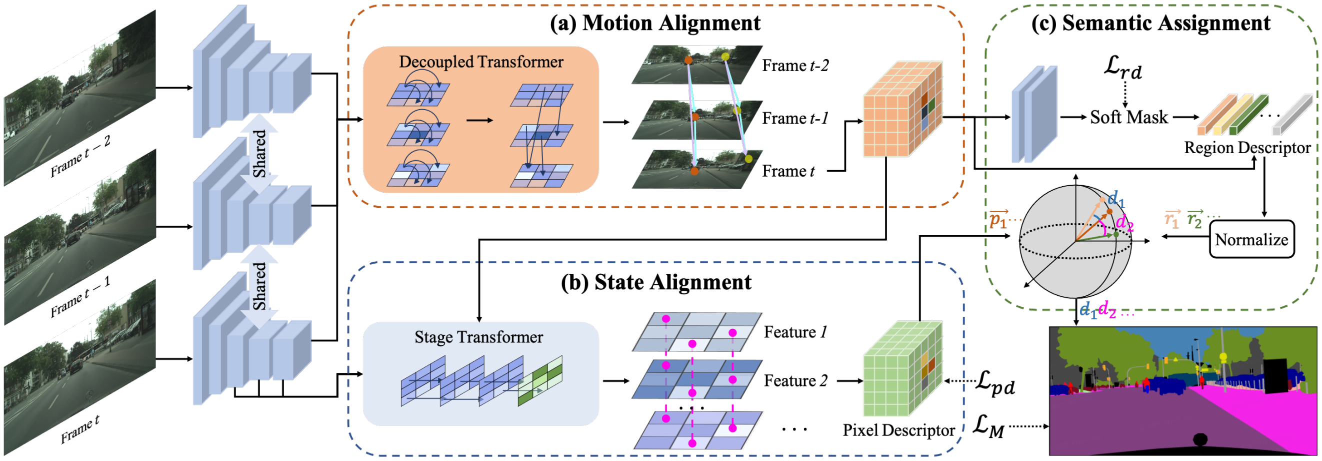


Figure 2. Framework of the proposed motion-state alignment framework. We first extract the independent features for each frame in one video with the shared feature extractor. Then, the dynamic semantics and static semantics about the current frame are aligned via motion and state alignment branches to produce region descriptors and pixel descriptors. Finally, each pixel descriptor is linked to the corresponding region descriptor in a partitioned way to obtain the final segmentation result.

warp inaccuracy and information inconsistency. Additionally, optical flow estimation has difficulties in handling occlusion and targets moving out of the frame, which is harmful to segmentation quality. The errors of optical flow network will propagate to other frames, which is harmful to the segmentation performance. Additionally, optical flow estimate has difficulties in handling occlusion. To tackle this dilemma, indirect methods (Fig. 1 (b)) [15, 23, 31, 33] propose to calculate a relationship matrix by attention-based mechanism to implicitly align features across multiple frames. Different from direct methods, indirect methods can implicitly extract temporal features. However, these methods calculate the similarity between all pixels across frames, which results in high computation cost and low inference speed. This limits the application on real-world applications, like autonomous driving. Therefore, video semantic segmentation still faces two main difficulties, *i.e.* information inconsistency and computation cost. Information inconsistency is to model temporal consistency between adjacent frames. In video semantic segmentation, applying segmentation network to separate frame will generate inconsistent results, but the relation between frames can increase information and decrease uncertainty during training and inference. Therefore, instead of using one single frame, encoding temporal relation between frames plays an important role in improving performance. Another difficulty is that deep and complex methods based on attention mechanism [30, 33] have high computation cost, although they have succeed in achieving high accuracy. It is hard to extend these outstanding segmentation methods to real-world applications, like autonomous driving. Therefore, methods with low computation cost are very non-trivial in real-world scenes.

To address the problem of information inconsistency and high computation cost, we put forward a novel motion-state alignment framework for video semantic segmentation as shown in Fig. 2. We divide the inter-frame information in one video into dynamic and static features, and construct corresponding motion and state alignment branches, which respond for maintaining the consistency between dynamic and static semantics respectively. In the motion alignment branch, an efficient spatial-temporal decoupled Transformer is established to align motion features across adjacent frames, which can maintain the motion consistency between temporal frames with low computations. With a stage Transformer, the state alignment branch is constructed to strengthen static features and maintain state consistency. Additionally, a semantic assignment mechanism is proposed to fuse both dynamic and static semantics with semantic partition. In the end, all pixels are partitioned into different regions with semantic categories as the final segmentation results. In the proposed framework, we construct different modules to align dynamic and static information respectively, which can effectively keep the information consistent across adjacent frames. Experimental results on two challenging video segmentation datasets have shown the effectiveness and efficiency of the proposed framework.

Our contributions are as follows: (1) We propose a novel motion-state alignment framework for video semantic segmentation to address information inconsistency and high computation costs from the new motion-state perspective. (2) Motion alignment mechanism is constructed to align information between temporal sequences and keep motion consistency. State alignment mechanism is built to conduct pixel-to-pixel alignment between different features and keep static consistency. (3) Our method achieves high accuracy on two challenging datasets: 81.5% mIoU on Cityscapes and 78.8% on CamVid, which is superior to the state-of-the-art methods in video semantic segmentation. In addition, our lightweight model with ResNet18 can reach 24.39 FPS on the Cityscapes dataset with competitive mIoU 75.8%.

2. Related Work

In this section, we review the related works that aim to deal with the challenges of video semantic segmentation.

2.1. Direct Methods

In order to realize temporal consistency, direct methods [12,39,40] adopt an optical flow network to predict per-pixel correspondence between the current frame and other frames. NetWarp [12] computes the flow between adjacent frames and warps intermediate features. VPLR [40] leverages an optical flow network and a spatio-temporal transformer recurrent layer to propagate information between frames. EFC [9] jointly trains segmentation network and optical flow network. It takes a lot of time to calculate optical flow between every two frames. To reduce computation cost and speed up the inference, some methods adopt the idea of keyframes and share the feature maps of sparse keyframes. For example, DFF [39] only runs the expensive feature extraction network on keyframes and then warps the feature maps to non-key frames. Accel [18] propagates high-detail features on a reference branch by means of optical flow estimation. Although direct methods are able to capture temporal information successfully, they still cannot be constrained by the disadvantages of optical flow networks.

2.2. Indirect Methods

Indirect methods [15, 23, 31] are another kind of method to gain temporal consistency. They can avoid the use of optical flow networks. TMANet [31] explores long-range temporal dependencies by means of the self-attention mechanism, which computes a heavy relation matrix between each pixel of neighboring frames. ETC [23] introduces temporal loss and temporal consistency knowledge distillation methods to improve the temporal consistency. TDNet [15] divides the complex feature network into shallow sub-networks for each frame and performs an attention propagation module to achieve temporal consistency. In indirect methods, although the dimension of feature maps is small, they still compute the high-dimension relation matrix between pixels. Besides, there is a much closer relationship in pixels of consecutive temporal or spatial context.

2.3. Video Transformer

In video understanding, like video super-resolution and video classification, it is a new trend to excavate long-sequence dependencies by ways of Transformer architecture [6, 10, 24, 30]. TimeSformer [3] adopts Transformer over the space-time volume on the action classification task. ViViT [1] develops several factorized Transformer blocks on spatial and temporal dimensions. VSR-Transformer [35] uses a spatial-temporal attention layer to exploit spatial-temporal information in video super-resolution. Transformer is very efficient to capture spatial and temporal relations. However, at present, no method keeps eye on applying Transformer to video semantic segmentation.

3. The Proposed Approach

In the section, we first introduce the novel motion-state perspective, and then explain each part of the framework.

3.1. Motion-state Perspective

To deal with the task of video semantic segmentation, we rethink this task from a novel motion-state perspective, in which semantic information in the video is divided into two parts, namely dynamic semantics and static semantics. Dynamic semantics refers to semantic information that can be easily extracted from motion, which is usually beneficial to semantic region location. Static semantics refers to semantic information that can be easily extracted at each state point during motion and contain more details of the semantic area. In this perspective, information consistency consists of motion consistency and state consistency.

Based on the motion-state perspective, we propose a novel motion-state alignment framework for video semantic segmentation (named as **MSAF**) as depicted in Fig. 2, to address the difficulties (*i.e.*, information inconsistency and computation cost). In this framework, we first extract independent features for each frame of one video. Then, through the motion alignment mechanism and state alignment mechanism, the dynamic semantics and static semantics about the current frame are aligned to guarantee information consistency. Next, pixel descriptors from the state alignment are linked to the corresponding region descriptors from the motion alignment in a semantic partition strategy. Thus, all pixels can be assigned to different regions with precise semantic categories and obtain the final segmentation results. In this way, our proposed method can learn both dynamic and static semantics, thus the video semantic regions are consistently segmented to obtain precise locations with low computational complexity. Details of the proposed approach are described as follows.

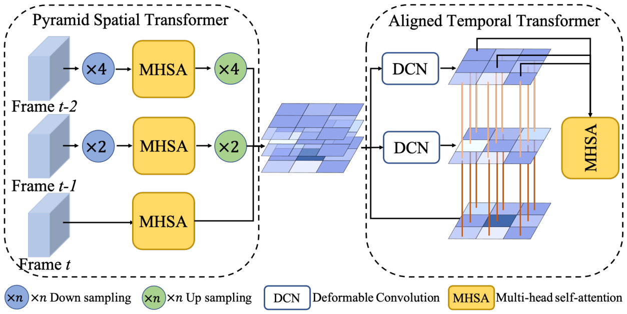

Pyramid Spatial Transformer
Aligned Temporal Transformer
×4
MHSA
×4
Frame t-2
×2
MHSA
×2
Frame t-1
MHSA
Frame t
DCN
DCN
MHSA
×n Down sampling
×n Up sampling
DCN Deformable Convolution
MHSA Multi-head self-attention

Figure 3. Structure of decoupled Transformer.

3.2. Feature Extractor

To extract features for each frame of one video, we take ResNet [14] as the feature extractor, which removes the last two layers (*i.e.*, classification and global average pooling layers) for the pixel-level prediction task. Feature extractor has five stages for feature encoding, named as $\mathcal{F}_s(\pi_s)$ with parameters π_s, where $s = 1, 2, \dots, 5$ represent the sth stage of the feature extractor. For convenience, we use $\mathcal{F}_s$ to represent $\mathcal{F}_s(\pi_s)$. To obtain larger feature maps, the strides of all convolutional layers in the residual modules $\mathcal{F}_4$ and $\mathcal{F}_5$ are set to 1. To further enlarge the receptive fields of high-level features, we set the dilation rates [36] to 2 and 4 for convolution layers in $\mathcal{F}_4$ and $\mathcal{F}_5$, respectively. Therefore, after the feature extractor, for a $H \times W$ input image, we can get a feature map with the size of $\frac{H}{8} \times \frac{W}{8}$ by the feature extractor. Note that we use the same shared feature extractor to capture features for every frame.

3.3. Motion Alignment

In one video, there are two important benefits of frame information. Firstly, information across adjacent frames can increase perception and decrease uncertainty. Secondly, the information can make it easy to capture the motion regions, which is beneficial to semantic region location.

In order to deal with the information inconsistency in many existing methods, we propose a motion alignment mechanism to capture the motion information in the video and maintain the motion consistency, as shown in Fig. 2. Specifically, in order to model the motion consistency between frames, with reference to optical flow that models a learnable "flow" to match pixels between frames, we propose an efficient decoupled Transformer to learn the "flow" as depicted in Fig. 3.

Decoupled Transformer. Transformer [30] is beneficial to long-term and global dependence, which is the main advantage compared with the ordinary convolutional operation. It is a promising choice to model the long-distance relationship in both time and space dimensions by utilizing Transformer, but it has the disadvantage in a lot of computation. Specifically, we represent the extracted features as $\mathcal{F}_5^t$ with the size of $\mathcal{C}_{\mathcal{F}_5} \times H_{\mathcal{F}_5} \times W_{\mathcal{F}_5}$, where $t, \mathcal{C}_{\mathcal{F}_5}, H_{\mathcal{F}_5}$ and $W_{\mathcal{F}_5}$ means the frame index, number of the channel, width and height of the feature map. In this paper, we empirically choose three frames with the feature $\mathcal{F}_5^{t-2}$, $\mathcal{F}_5^{t-1}$ and $\mathcal{F}_5^t$ as an example. Given a naive explanation, ignoring the effect of scaled dot-product and multi-head on reducing the amount of computation, Transformer can model the relationship between all pixels in a feature map. Thus, when Transformer is utilized to model spatio-temporal information, the computation cost of Transformer is as follows:

$$C(\text{Transformer}) = 3n \times n \times n = 3n^3, \tag{1}$$

where $n = \mathcal{C}_{\mathcal{F}_5} \times H_{\mathcal{F}_5} \times W_{\mathcal{F}_5}$ means the number of pixels in a feature map. Considering frames $t-2$, $t-1$, and t, the computation cost that constructs the relation between one pixel with all other pixels in spatio-temporal dimension is $n \times n$. Therefore, the total computation cost between frame t and $t-1$ as well as $t-2$ is n^3. Therefore, the computation cost of vanilla Transformer is $3n^3$. Usually, this computation cost is tolerable, unless the size of feature graph is relatively large.

To reduce the computation cost, we consider decoupling the spatio-temporal relationship and propose a decoupled Transformer (D-Transformer) as shown in Fig. 3. D-transformer consists of two modules *i.e.*, pyramid spatial transformer (PST) and aligned temporal transformer (ATT).

In PST, features captured by the feature extractor for each frame are as the input. Next, features are downsampled to reduce the amount of computation. The distant frames in the temporal dimension provide more abstract information with a larger sampling rate (×4 in frame $t-2$), while the near frames provide more specific information (×2 in frame $t-1$). Then, features after sampling are fed to the multi-head self-attention module (MHSA) [30] to model global dependency in the spatial dimension. After that, the corresponding upsampling operation is adopted to ensure the precise alignment in spatial dimensions. PST block is formally defined as:

$$\begin{aligned}
\mathcal{D}_{\text{PST}}^{t-2} &= \text{US}^{t-2}(\text{MHSA}_{spa}^{t-2}(\text{DS}^{t-2}(\mathcal{F}_5^{t-2}))) \\
\mathcal{D}_{\text{PST}}^{t-1} &= \text{US}^{t-1}(\text{MHSA}_{spa}^{t-1}(\text{DS}^{t-1}(\mathcal{F}_5^{t-1}))), \\
\mathcal{D}_{\text{PST}}^{t} &= \text{MHSA}_{spa}^{t}(\mathcal{F}_5^{t})
\end{aligned} \tag{2}$$

where US and DS mean up-sampling and down-sampling operation. MHSA_{spa} means multi-head self-attention module in the spatial dimension with the following formulation:

$$\begin{aligned}
X' &= \text{Attention}(\text{LN}(X)) + X \\
\text{MHSA}_{spa} &= \text{MLP}(\text{LN}(X')) + X',
\end{aligned} \tag{3}$$

where X is the input tensor, LN is layer normalization layer [2], MLP is multi-layer perceptron, and Attention is the self-attention module referring to [30] for details. In this way, spatial features of neighboring frames are extracted in a pyramid way, and the computation cost is as follows (note we ignore the reduction of computation caused by downsampling for fair):

$$C(\text{PST}) = 3 \times n^2 = 3n^2. \tag{4}$$

In ATT, three feature maps of different frames after PST are used as input. Due to motion, a pixel at the current frame may not match the pixel at the same location of adjacent frames. Therefore, it is necessary to first align the spatial information pixel by pixel in the temporal dimension. For this purpose, we introduce deformable convolutional layers [8] in adjacent frames to extract the offset of each pixel caused by motion. The detail is as follows:

$$Y(p) = \sum_{k=1}^{K \times K} w_k \cdot X(p + p_k + \Delta p_k), \tag{5}$$

where K defines the number of points that may correspond to the pixel at this position in the adjacent frame, $X(p)$ is the pixel at location p of input feature X, and $Y(p)$ represents the pixel at location p of input feature Y. p_k means the location range of $K \times K$ centered on the current position, w_k is the weight value and Δp_k is the learned offset. In this way, each pixel can learn the position of the associated pixel in adjacent frames. Here we set $K = 3$ to perceive multiple positions and ensure the information alignment between frames. Then, each pixel at the current frame and its associated pixels in adjacent frames are passed together to MHSA to model the temporal long-term dependency in the spatial dimension. The ATT block is formally defined as:

$$\mathcal{D}_{\text{ATT}} = \text{MHSA}_{tem}(\text{DCN}^{t-2}(\mathcal{D}^{t-2}_{\text{PST}}), \text{DCN}^{t-1}(\mathcal{D}^{t-1}_{\text{PST}}), \mathcal{D}^{t}_{\text{PST}}), \tag{6}$$

where DCN is the deformable convolutional operation and MHSA$_{tem}$ means multi-head self-attention in the temporal dimension. DCN is only used in neighbouring two frames to align a pixel in the current frame with pixel in neighbouring frames. In this way, we can get the computation cost of ATT as follows (note the computation cost of DCN can be regarded as $K^2 n$):

$$C(\text{ATT}) = 2 \times 3n + 2 \times 9n = 24n. \tag{7}$$

Therefore, the whole computation cost of D-transfomer is:

$$\begin{aligned} C(\text{D-Transformer}) &= C(\text{PST}) + C(\text{ATT}) \\ &= 3n^2 + 24n \end{aligned}. \tag{8}$$

Comparing Equation (1) with (8), we can clearly see that the value of C(Transformer) - C(D-Transformer) increases monotonously and linearly in the range $[\frac{1+\sqrt{33}}{2}, +\infty)$, and $\lim_{n\to+\infty} \frac{C(\text{Transformer})}{C(\text{D-Transformer})} = n$. Obviously, $n \gg \frac{1+\sqrt{33}}{2}$, which means that the computation cost of D-Transformer is much smaller than original Transformer and D-Transformer is much more efficient.

Therefore, D-Transformer can construct the spatio-temporal relationship between video frames, and implicitly ensure the alignment of motion information between frames, which is beneficial to ensure motion consistency.

For convenience, motion alignment branch is denoted as $\mathcal{M}(\pi_{\mathcal{M}})$ with parameters $\pi_{\mathcal{M}}$. The output of $\mathcal{M}$ including three features *i.e.*, $\mathcal{M}^{t-2}$, $\mathcal{M}^{t-1}$ and $\mathcal{M}^{t}$ that all have the same shape with $\mathcal{F}^t_5$, which captures the dynamic information of one video. In order to obtain semantics, $\mathcal{M}$ is supervised by the ground-truth semantic mask, and the supervision signal will be mentioned later. Therefore, $\mathcal{M}$ carries rich dynamic semantics. In this manner, motion alignment efficiently maintains motion consistency.

3.4. State Alignment

Except for dynamic information, static information also plays an important role in video semantic segmentation. The static semantics can be easily extracted from a frame described as a motion state at a certain moment, which is usually more conducive to restoring the details of semantic areas. In order to ensure the expressive ability of state information in feature space, we align the state information of different feature sources and propose a state alignment mechanism as depicted in Fig. 2.

Stage Transformer. As neural network goes deeper, low-level features contain more details of images, and high-level features have more semantic information. To contain both detail and semantic information, in the state alignment branch, we collect features at different levels of the current frame (regarded as state features) and feed them to the proposed stage Transformer (S-Transformer), which can gain detailed information and model the relationship of corresponding pixels between features of different levels in the stage dimension. The S-Transformer block is formally defined as:

$$\mathcal{S}_{TR} = \text{MHSA}_{sta}(\mathcal{F}^t_3, \mathcal{F}^t_4, \mathcal{F}^t_5, \mathcal{M}^t), \tag{9}$$

where MHSA$_{sta}$ is the multi-head self-attention module in the stage dimension. Moreover, to enhance the features of current frames, we also add the features $\mathcal{M}^t$ of current frames from motion alignment to S-Transformer. In this way, static information from different stage and dynamic information of current frame are aligned together and modeled for the state consistency.

Pixel Descriptor. We define the state alignment branch as $\mathcal{S}(\pi_{\mathcal{S}})$ with parameters $\pi_{\mathcal{S}}$, The output of $\mathcal{S}$ then goes through two convolutional layers to gain pixel descriptor. For convenience, we name the pixel descriptor as $\mathcal{P}$. To obtain the semantic information, $\mathcal{P}$ is supervised by the ground-truth mask of semantic segmentation (represented as G) by minimizing the loss:

$$\mathcal{L}_{pd} = CE(\mathcal{P}, G), \tag{10}$$

where $CE(\cdot,\cdot)$ is the cross-entropy loss function with the following formulation:

$$CE(P, G) = -\sum_{i=1}^{H\times W}\sum_{j=1}^{C} G_{i,j}\log P_{i,j}, \tag{11}$$

where $P_{i,j}$ and $G_{i,j}$ represents the predicted and real probabilities that the ith ($i \in \mathbb{R}^{H\times W}$) pixel belongs to jth ($1 \le j \le C$) class, respectively. Under the supervision, we regarded each pixel $pix \in \mathbb{R}^{\mathcal{C}_{\mathcal{F}_5}\times H_{\mathcal{F}_5}\times W_{\mathcal{F}_5}}$ in $\mathcal{P}$ as the pixel descriptor, which represents the static semantics and state details of each pixel.

3.5. Semantic Assignment

After obtaining the motion features $\mathcal{M}^t$ and state features $\mathcal{P}$, we propose the semantic assignment mechanism to produce the final segmentation results as shown in Fig. 2.

Region Descriptor. From motion information, it is easy to extract region-level information, which is beneficial to semantic region location. Therefore, we introduce the region descriptors to characterize the region-level features.

To generate region descriptors, we first add two convolutional layers after $\mathcal{M}^t$ to obtain dynamic information. After the two convolutional layers, the supervision from the ground-truth semantic mask G is imposed to produce the soft mask $\mathcal{M}^{t'}$ by minimizing the loss:

$$\mathcal{L}_{rd} = CE(\mathcal{M}^{t'}, G). \tag{12}$$

Then, we weight $\mathcal{M}$ with each logit (feature map before softmax) of $\mathcal{M}^t$, and average the result by global pooling in the spatial dimension to obtain region descriptors $\mathcal{R} \in \mathbb{R}^{Cls\times 1\times 1}$ with Cls is the number of semantic categories. Each region descriptor $r \in \mathcal{R}$ represents the region of one semantic category.

Semantic Partition. With region descriptors and pixel descriptors with similar semantics, we propose the semantic partition strategy to link pixel descriptors to corresponding region descriptors as shown in Fig. 2.

In order to judge the association between pixels and regions, we use the minimum cosine distance as the evaluation metric. Specifically, we compute all the distance between pixel descriptor $pix_i \in \mathcal{P}, i = 1, 2, \ldots, H_{\mathcal{F}_5} \times W_{\mathcal{F}_5}$ with each region descriptor $r_j \in \mathcal{R}, j = 1, 2, \ldots, Cls$, and then the index with the minimum distance is taken as the semantic category. This processing is formulated as:

$$\begin{aligned} CLASS_{pix_i} &= \text{argmin}_j(d_1, d_2, \ldots, d_j, \ldots, d_{Cls}) \\ d_j &= 1 - \cos(pix_i, r_j) \\ \cos(pix_i, r_j) &= \frac{pix_i \cdot r_j}{|pix_i||r_j|} \end{aligned}, \tag{13}$$

where $|\cdot|$ means the magnitude of a vector. In Equation (13), $pix_i \cdot r_j$ is dot product of pix_i and r_j. For each d_j, the ignoring magnitude operation about $|pix_i|$ doesn't affect the loss value because of the same pix_i, and we add normalization operation to r_j for each region descriptors. In this way, Equation (13) can be reduced to a matrix multiplication (*i.e.*, the matrix multiplication of $\mathcal{P}$ and $\mathcal{R}$) followed by an argmin operation in channel dimension. After this, we obtain the final results M, which is expected to approximate the ground-truth masks M by minimizing the loss:

$$\mathcal{L}_M = CE(M, G), \tag{14}$$

For convenience, semantic assignment branch is denoted as $\mathcal{A}(\pi_{\mathcal{A}})$ with parameters $\pi_{\mathcal{A}}$.

By taking the losses of Eqs.(10), (12), and (14), the overall learning objective can be formulated as follows:

$$\min_{\mathbb{P}} \mathcal{L}_{pd} + \mathcal{L}_{rd} + \mathcal{L}_M, \tag{15}$$

where $\mathbb{P}$ is the set of $\{\pi_s\}_{s=1}^5$, $\pi_{\mathcal{M}}$, $\pi_{\mathcal{S}}$, and $\pi_{\mathcal{A}}$.

4. Experiments and Results

4.1. Experimental Setup

Datasets. To verify the effectiveness of our proposed method, we evaluate results on two public video semantic segmentation datasets, *i.e.*, Cityscapes [7] and CamVid [4] datasets. The Cityscapes dataset consists of a large set of video frames from 50 European cities. The dataset provides 5,000 fine annotations (*i.e.*, 2,975 for training, 500 for validation, 1,525 for testing) and 20,000 coarse annotations. In Cityscapes, there are 19 semantic categories, and the size of images is 1024x2048. In our experiments, we only use 5,000 finely-annotated images for training and inference. In addition, CamVid has 4 videos with 11-class annotations, which are divided into 367/101/233 for training/validation/testing respectively. The image resolution in CamVid is 640x960.

Evaluation Metrics. Our method is evaluated on the accuracy, efficiency, and temporal consistency. We apply mean Interaction over Union(mIoU) to report the accuracy. mIoU averages the IoU value in all valid semantic categories of the dataset. Frames per second (FPS) is used to present the efficiency of methods. And temporal consistency(TC) [20] is adopted to evaluate the temporal stability in video tasks, which calculates the mean flow warping error between two neighboring frames.

Training and Inference. We verify the robustness of our methods on a series of backbones, *i.e.*, ResNet18 [14], ResNet50, ResNet101, PSPNet18 [38], PSPNet50, PSPNet101, HRNet [32]. During training, our method is optimized by AdamW [26] with a batch size of 2 per GPU. We apply a "poly" learning rate policy, and the initial learning

Table 1. Performance of state-of-the-art models on Cityscapes. “-V” (“-T”) means on the validation (test) dataset, and a single “-” indicates that data cannot be obtained. Note that all our results are in single-scale testing, and all our models are only trained on the training dataset (fine). The best result are in **red**.

Models	Year	Backbone	mIoU-V	mIoU-T	FPS	TC
PSPNet18	2017	PSPNet18	75.5	-	10.99	68.5
Accel	2019	ResNet18	-	72.1	2.27	70.3
ETC	2020	PSPNet18	-	73.1	9.5	70.6
TDNet	2020	PSPNet18	76.8	-	11.76	-
STT	2021	ResNet18	77.3	-	11.5	73.0
PC	2022	HRNet-w18	76.4	-	-	71.2
Ours	-	ResNet18$_{512}$	75.8	-	24.39	72.2
Ours	-	ResNet18$_{768}$	77.8	-	13.54	73.1
Ours	-	ResNet18	78.2	78.3	8.76	73.8
Ours	-	PSPNet18	**78.3**	**78.5**	7.91	73.9
PSPNet50	2017	PSPNet50	78.1	-	4.2	-
Accel	2019	ResNet50	-	74.2	1.49	-
TDNet	2020	PSPNet50	79.9	79.4	5.62	-
TMANet	2021	ResNet50	80.3	-	2.14	-
Ours	-	ResNet50	80.4	**79.8**	2.57	73.9
Ours	-	PSPNet50	**80.7**	-	2.37	74.1
PSPNet101	2017	ResNet101	79.7	78.4	2.78	69.7
DFF	2017	ResNet101	68.7	-	5.59	71.4
PEARL	2017	ResNet101	76.5	76.5	-	-
NetWarp	2017	PSPNet101	80.6	-	0.33	-
LVS	2018	ResNet101	76.8	-	2.63	-
GRFP	2018	PSPNet101	80.2	-	-	-
Accel	2019	ResNet101	-	75.5	1.15	-
ETC	2020	PSPNet101	79.5	-	-	-
LMA	2021	PSPNet101	78.5	-	1.32	-
PC	2022	ResNet101	76.3	-	-	72.4
Ours	-	ResNet101	81.3	80.0	1.96	74.8
Ours	-	PSPNet101	81.5	81.1	1.82	75.2
Ours	-	HRNetW48	**82.1**	**81.5**	2.02	75.4

rate is 0.00006. A linear warm-up strategy is used in the first 1,500 iterations. We train the model for 80,000 iterations in total. The images are randomly cropped by 769x769 on Cityscapes and 640x640 on CamVid. Data augmentation, like flip and resize, is adopted in our method.

4.2. Comparisons with state-of-the-art methods

Our method is compared with state-of-the-art methods including PEARL [19], NetWarp [12], LMA [28], DFF [39], GRFP [27], TMANet [31], DFF [39], LVS [22], MC [16], Accel [18], ETC [23], TDNet [15], EFC [9], STT [21] and PC [37].

In Table 1, we compare our method with recent state-of-the-art methods on the Cityscapes dataset. Our method shows great improvement in accuracy on multiple backbones, which shows the robustness of our method. Our method based on ResNet18 has made great achievement with mIoU 78.5% on the test set, which outperforms other lightweight methods on both val and test sets. Additionally, our ResNet18-based method with input size 512x1024 reaches 24.39 FPS, which can achieve real-time inference. With backbone ResNet50 and PSPNet50, our method also shows great accuracy and speed. In addition, on complex backbones, our method achieves a mIoU of 80.0%, 81.1%, 81.5% with backbone of ResNet101, HRNet48, PSPNet101 respectively. Our method with a complex backbone also has advantages in inference time.

Table 2. Performance of state-of-the-art models on CamVid. “-V” means on validation dataset and “-T” means on test dataset. The best result are in **red**.

Models	Year	Backbone	mIoU-V	mIoU-T	FPS
MC	2018	SegNet	-	63.7	0.5
Accel	2019	ResNet18	-	66.7	5.88
TDNet	2020	PSPNet18	-	72.6	25.00
STT	2021	PSPNet18	76.1	-	24.7
PC	2022	HRNetW18	73.2	-	-
Ours	-	ResNet18	**76.6**	**75.7**	25.13
NetWarp	2017	Dilation	-	67.1	2.53
GRFP	2018	Dilation8	-	66.1	4.35
Accel	2019	ResNet50	-	67.7	4.18
TDNet	2020	PSPNet50	-	76.0	11.11
EFC	2020	Dilation8	-	67.4	-
TMANet	2021	ResNet50	76.5	-	7.72
Ours	-	ResNet50	**77.4**	**77.2**	9.26
Accel	2019	ResNet101	-	69.3	3.13
PC	2022	ResNet101	75.2	-	-
Ours	-	ResNet101	77.6	77.5	6.54
Ours	-	PSPNet101	**77.9**	78.0	5.99
Ours	-	HRNetW48	77.7	**78.8**	6.14

As listed in Table 2, we also validate the robustness of our method on the CamVid dataset. We can see that our method with different backbones (*i.e.*, ResNet18, ResNet50, ResNet101, PSPNet101) has achieved significant improvement on both accuracy and efficiency. In particular, compared with Accel in different backbones, our method has significantly improved and runs 2-4 times faster.

In Table 1, we also compare Temporal Consistency (TC) with state-of-the-art methods. The TC of our MSAF is higher than other methods, which verifies that our MSAF can achieve alignment between frames implicitly without the help of optical flow algorithms.

In Fig. 4, we compare the qualitative results with the state-of-the-art method. We can see that our proposed method can segment the images much more precisely in some complex scenes. These show that our MSAF can extract better semantics information for the segmentation task.

4.3. Ablation Analysis

In Table 3, we carry out ablation experiments on the Cityscapes validation set with backbone ResNet50 to present the effectiveness of each part in our method.

To investigate the effectiveness of the motion alignment, we replace the motion alignment with a vanilla Transformer

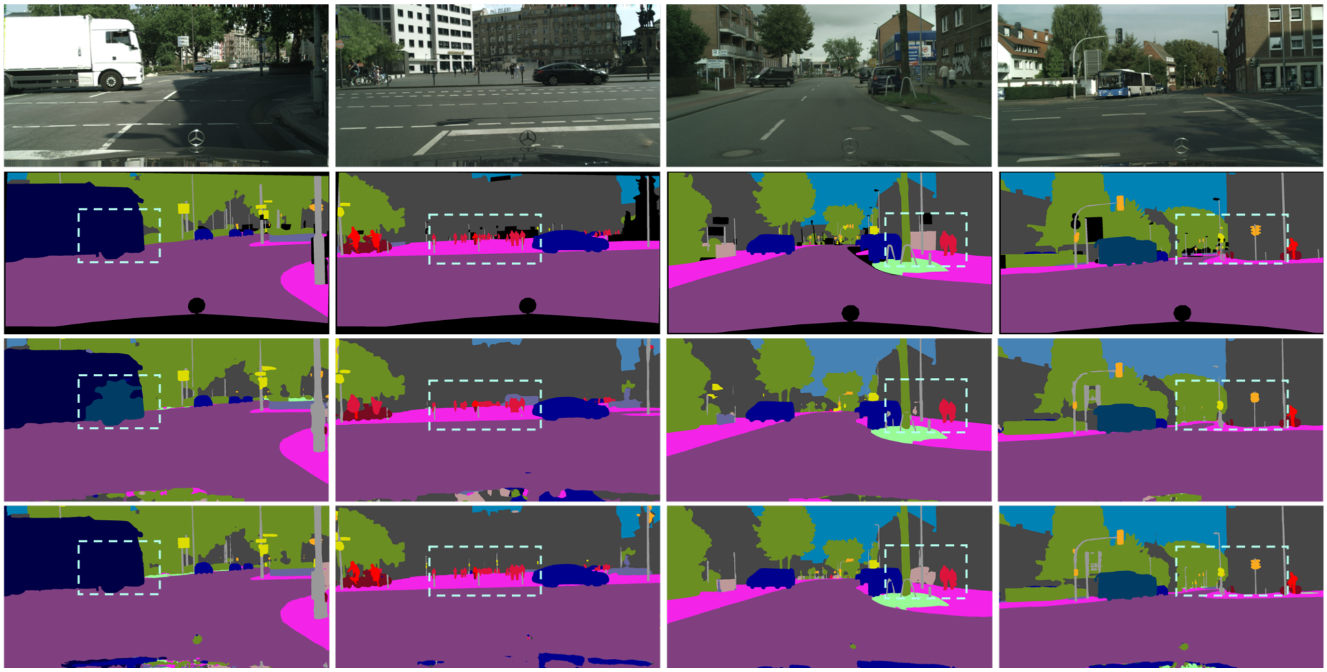


Figure 4. Qualitative results on Cityscapes validation set. From top to bottom: original images, the ground truth, LMA, the segmentation results of our ResNet101 based model.

Table 3. Performance of different settings of the proposed method (ResNet-50 based) on Cityscapes validation dataset.

	MonA	StaA	SemA	mIoU
Baseline				74.5
Baseline + TR				77.5
Motion only	✓			79.1
State only		✓		76.5
w/o Motion		✓	✓	77.3
w/o State	✓		✓	79.7
w/o Semantics	✓	✓		79.8
w/o DCN	✓	✓	✓	79.8
MSAF	✓	✓	✓	**80.4**

(named baseline+TR in Table 3), which has more parameters than the whole framework MSAF. Although baseline+TR has shown an increase of 3.0% compared with baseline (74.5%), our method only with motion alignment (Motion only) achieves better accuracy (79.1%). This explains that our improvement is not due to the use of the Transformer module, but attributed to the design of motion alignment with decoupled transformer for video semantic segmentation. To provide evidence for this further, we remove the motion alignment (w/o Motion) in our MSAF. In this way, there is no temporal information in “w/o Motion”. Compared with w/o Motion, MSAF gains an improvement by mIoU 3.1%, which shows that motion alignment is helpful to guarantee the temporal consistency in video semantic segmentation. By the way, DCN brings a gain of 0.6%.

We also conduct an experiment with state alignment only (State only) and without state alignment (w/o State) to validate the effectiveness of state alignment. The feature maps from motion alignment and backbone are concatenated and forwarded to semantic alignment. In Table 3, we can see state alignment can improve the accuracy from 79.7% to 80.4%. This also shows that static information is important for video semantic segmentation.

To validate the effectiveness of semantic assignment, we remove the semantic assignment (w/o Semantics) and average the masks of motion alignment and state alignment as the final result. We find that the method with semantic assignment achieves 0.6% improvement.

5. Conclusion

In this paper, we rethink consistency in video semantic segmentation from the perspective of motion and state semantics, and propose a motion-state alignment framework to achieve both dynamic and static consistency. In this framework, we build a motion alignment branch to encode temporal relations with dynamic semantics. State alignment branch is proposed to enhance features at the pixel level to obtain rich static semantics. Finally, semantic assignment links pixel descriptors from state alignment with region descriptors from motion alignment. Experiments on challenging datasets show that our approach outperforms recent state-of-the-art methods in both accuracy and efficiency.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 3

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 3

[4] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pages 44–57. Springer, 2008. 6

[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 transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[6] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*, 1(2):3, 2021. 3

[7] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 6

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 5

[9] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: joint learning of video segmentation and optical flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10713–10720, 2020. 3, 7

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[11] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1

[12] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video cnns through representation warping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4453–4462, 2017. 1, 3, 7

[13] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multiscale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3562–3572, 2019. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 6

[15] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8827, 2020. 1, 2, 3, 7

[16] Po-Yu Huang, Wan-Ting Hsu, Chun-Yueh Chiu, Ting-Fan Wu, and Min Sun. Efficient uncertainty estimation for semantic segmentation in videos. In *Proceedings of the European Conference on Computer Vision*, pages 520–535, 2018. 7

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 1

[18] Samvit Jain, Xin Wang, and Joseph E Gonzalez. Accel: A corrective fusion network for efficient semantic segmentation on video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8866–8875, 2019. 1, 3, 7

[19] Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, et al. Video scene parsing with predictive feature learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5580–5588, 2017. 1, 7

[20] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision*, pages 170–185, 2018. 6

[21] Jiangtong Li, Wentao Wang, Junjie Chen, Li Niu, Jianlou Si, Chen Qian, and Liqing Zhang. Video semantic segmentation via sparse temporal transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 59–68, 2021. 7

[22] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5997–6005, 2018. 7

[23] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *European Conference on Computer Vision*, pages 352–368. Springer, 2020. 1, 2, 3, 7

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 3

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[27] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6819–6828, 2018. 7

[28] Matthieu Paul, Martin Danelljan, Luc Van Gool, and Radu Timofte. Local memory attention for fast video semantic segmentation. *arXiv preprint arXiv:2101.01715*, 2021. 1, 7

[29] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1164–1172, 2015. 1

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 3, 4

[31] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. *arXiv preprint arXiv:2102.08643*, 2021. 1, 2, 3, 7

[32] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 6

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[34] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*, pages 1385–1392, 2013. 1

[35] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5791–5800, 2020. 3

[36] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016. 4

[37] Yizhe Zhang, Shubhankar Borse, Hong Cai, Ying Wang, Ning Bi, Xiaoyun Jiang, and Fatih Porikli. Perceptual consistency in video segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2564–2573, 2022. 7

[38] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 6

[39] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358, 2017. 1, 3, 7

[40] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019. 1, 3