Video Semantic Segmentation via Sparse Temporal Transformer

Jiangtong Li*
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
keep_moving-Lee@sjtu.edu.cn

Wentao Wang*
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
wwt117@sjtu.edu.cn

Junjie Chen
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
chen.bys@sjtu.edu.cn

Li Niu†
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
ustcnewly@sjtu.edu.cn

Jianlou Si
SenseTime Research,
SenseTime
sijianlou@sensetime.com

Chen Qian
SenseTime Research,
SenseTime
qianchen@sensetime.com

Liqing Zhang†
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
zhang-lq@cs.sjtu.edu.cn

ABSTRACT

Currently, video semantic segmentation mainly faces two challenges: 1) the demand of temporal consistency; 2) the balance between segmentation accuracy and inference efficiency. For the first challenge, existing methods usually use optical flow to capture the temporal relation in consecutive frames and maintain the temporal consistency, but the low inference speed by means of optical flow limits the real-time applications. For the second challenge, flow-based key frame warping is one mainstream solution. However, the unbalanced inference latency of flow-based key frame warping makes it unsatisfactory for real-time applications. Considering the segmentation accuracy and inference efficiency, we propose a novel Sparse Temporal Transformer (STT) to bridge temporal relation among video frames adaptively, which is also equipped with query selection and key selection. The key selection and query selection strategies are separately applied to filter out temporal and spatial redundancy in our temporal transformer. Specifically, our STT can reduce the time complexity of temporal transformer by a large margin without harming the segmentation accuracy and temporal consistency. Experiments on two benchmark datasets, Cityscapes and Camvid, demonstrate that our method achieves the state-of-the-art segmentation accuracy and temporal consistency with comparable inference speed.

CCS CONCEPTS

• **Computing methodologies** → **Video segmentation**.

*Both authors contributed equally to this research.
†Corresponding Authors.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475409

KEYWORDS

semantic segmentation, video semantic segmentation, transformer, temporal consistency, semi-supervised learning

ACM Reference Format:
Jiangtong Li, Wentao Wang, Junjie Chen, Li Niu, Jianlou Si, Chen Qian, and Liqing Zhang. 2021. Video Semantic Segmentation via Sparse Temporal Transformer. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475409

1 INTRODUCTION

Video semantic segmentation (VSS) aims to assign a semantic label to each pixel in video frames. As an important research topic for applications such as autonomous driving and robotics, it has attracted widespread attention in the research community [15, 20, 25, 28, 36, 37, 40]. Whereas, VSS is quite challenging due to two reasons: 1) since the consecutive annotation does not exist in current datasets, the models need to perform temporally consistent semantic segmentation in a semi-supervised manner. 2) for real-time applications, the models have to balance segmentation accuracy and inference efficiency.

For temporally consistent semantic segmentation, the method should be capable of aligning the prediction between consecutive frames. Therefore, one feasible solution is estimating frame-to-frame motion warping (*e.g.*, optical flow) to segment consecutive frames, like NetWarp [20] and GRFP [40]. However, the inference of optical flow is time-consuming, making these methods unsuitable for real-time applications. To utilize the optical flow more efficiently, ETC [37] adopted warped prediction loss to constrain the prediction of current frame during training and performed single-frame prediction during inference. Nevertheless, the usage of optical flow in ETC [37] is affected by estimation errors caused by occlusions, non-textured regions, and large motions, which might be harmful for the segmentation model. Besides, only counting on the current frame during inference also limits its temporal consistency.

To balance the segmentation accuracy and inference efficiency, existing methods can be divided into two groups. On the one hand, some methods employ large models towards the key frames, and

propagate to non-key frames using optical flows [50, 59, 61] or utilize small model to process the non-key frames [28]. However, such methods have two drawbacks: 1) optical flow models become more and more complicated, so the inference speed is compromised; 2) it requires different inference time for different frames, which causes unbalanced latency and limits its practical usage. On the other hand, some methods adopt knowledge distillation from large model towards small model [25, 37], which aims to improve the segmentation efficiency without increasing the computational cost. Nevertheless, since the model capacity of small model does not change, the improvement from the large model is also limited (+0.4 % mIoU in TDNet [25] and +0.47% mIoU in ETC [37]).

To achieve temporally consistent semantic segmentation without the favor of optical flow, the key is to align the corresponding semantic objects adaptively. Transformer [46], which has achieved great success in natural language processing (NLP) [13, 14] and computer vision (CV) [3, 4, 16], is capable of correlating the similar features with multi-head attention. Therefore, we propose to incorporate a temporal transformer into existing segmentation models as an adaptive module to capture the temporal relation among consecutive frames. According to the definition of query/key/value in transformer, the current frame is regarded as query frame and several previous frames are regarded as key/value frames. In detail, given a video clip with several frames, we first employ the encoder of image segmentation model (*e.g.*, PSPNet [56]) to encode each frame into feature maps. The feature map of current frame is treated as the query feature map and the feature maps of previous frames are treated as key/value feature maps. The output of temporal transformer, which shares the same shape as the query feature map, will be sent to the decoder of segmentation model. Note that the encoded features of previous frames can be reused in the inference steps for later frames, which will not introduce extra computation for encoding key frames and can also keep balanced latency. However, for a video containing several frames with high-resolution, the time complexity of the interaction between query feature map and key/value feature maps is extremely high and intolerable in the vanilla temporal transformer.

To make a trade-off between segmentation accuracy and inference efficiency, we further propose two selection strategies (*i.e.*, query selection and key selection) to reduce the time complexity of temporal transformer. Because key feature map and value feature map are identical in our temporal transformer, we only mention the operation to key feature map in the remainder of this section for brevity. In Figure 1 (a), we show two frames and their corresponding semantic labels, where the regions enclosed in orange (*resp.*, blue) boxes contain a single (*resp.* multiple) semantic object(s). Following Marin *et al.* [39], we divide the regions in the frames into simple regions and complex regions, where the complex regions usually contain multiple semantic categories and the simple regions only contain single semantic category. For complex regions, capturing the temporal relation would be helpful to improve the segmentation accuracy and temporal consistency. Instead, enriching the simple regions with temporal information may not bring much difference. Therefore, we propose Neighboring Similarity Matrix (NSM) combining cosine distance and Kullback–Leibler (KL) divergence to select the complex regions from the query frame, which is dubbed as query selection. Besides, in Figure 1 (b), we show five consecutive frames, where the first four frames are the key frames and the last frame is the query frame. For the first example, if we want to track the person or bike (enclosed in green box) in the key frame, it is a waste to search the whole region in key frames, whereas, enlarging the searching region (enclosed in red box) in a proper scale would be wiser. Therefore, for each query point in the query frame, we first select a small region in the nearest key frame as the key region. Then we gradually enlarge the radius of key region from the nearest key frame to the farthest key frame, which is dubbed as key selection. Based on the above two selection strategies, the time complexity of temporal transformer can be reduced from $O(Th^2w^2)$ to $O(Thw)$, in which h and w represent the height and width of the feature maps and T represents the number of key frames. We notice that TDNet [25] also applied attention module to capture the temporal relation, but TDNet can only focus on the nearest one or three frame(s), which may impede capturing temporal relation in a longer range. We integrate query selection and key selection into our temporal transformer, leading to a novel Sparse Temporal Transformer (STT) for video semantic segmentation, which can not only capture the temporal relation to maintain the temporal consistency but also balance the segmentation accuracy and inference efficiency in a good manner. It is worth noting that our STT is a plug-in module which can be incorporated into a wide range of semantic segmentation models.

We conduct extensive experiments on two benchmark datasets, Cityscapes [11] and Camvid [2], which show that our method achieves significant improvement for temporal consistency and segmentation accuracy. Further analyses on our query selection and key selection also show that our proposed selection strategies could locate the boundary region and align the semantic objects adaptively. Our contributions can be summarized as:

- We incorporate the transformer architecture into the image semantic segmentation model to capture temporal relation for video semantic segmentation task.
- We propose a novel Sparse Temporal Transformer (STT) module with key selection and query selection to balance the segmentation accuracy and inference efficiency in a good manner. Our proposed selection strategies can reduce the time complexity by a large margin without harming the segmentation accuracy and temporal consistency.
- Extensive experiments on two video semantic segmentation datasets, *i.e.* Cityscapes and Camvid, demonstrate the effectiveness of our method.

2 RELATED WORK

2.1 Image Semantic Segmentation

Image semantic segmentation [5, 21, 22, 29, 31, 32, 35, 41, 55] is the foundation of video semantic segmentation. The success of deep learning brought significant improvement to image semantic segmentation. Since Long *et al.* [38] first proposed a fully convolutional network (FCN) to segment images, deep convolution neural networks become the mainstream solution to semantic segmentation. Following FCN, recent researches proposed various schemes for efficient segmentation or high-accuracy segmentation. In [5–7, 54], dilated convolutions were used to enlarge receptive field. By means of stronger backbone networks like GoogleNets [45],

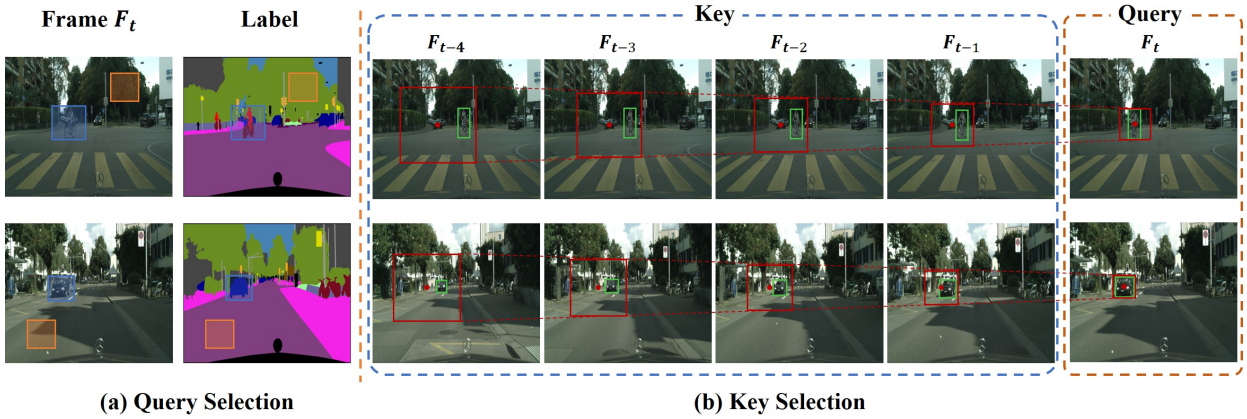

Figure 1: (a) Illustration of query selection. The simple (*resp.*, complex) regions enclosed in orange (*resp.*, blue) boxes contain a single semantic label (*resp.*, multiple semantic labels). We attempt to select the complex regions as queries. (b) Illustration of key selection. Given an object (enclosed in green boxes) in the query frame, to track this object in the key frames, we enlarge the searching region (enclosed in red boxes) from near frame to far frame, in which the searching regions form our selected key regions. Best viewed by zooming in.

ResNets [23], and DenseNets [26] or light-weight backbone networks like MobileNet [24] and BiSeNet [53], better segmentation performance or efficient segmentation can be achieved. To exploit multi-scale context, SegNet [1], Unet [43] and RefineNet [34] used auto-encoder architecture with skip connection to fuse low-level features with high-level features. PSPNet [56] and DeepLab [6] proposed PPM (Pyramid Pooling Module) and ASPP (Atrous Spatial Pyramid Pooling) to integrate multi-scale context for comprehensive scene understanding, respectively. HRNet [47] repeatedly aggregated features from four parallel branches with different resolutions. More recently, to aggregate global context, transformer based method SETR [57] was proposed to capture long range dependencies from image.

2.2 Video Semantic Segmentation

Unlike image semantic segmentation, video semantic segmentation (VSS) aims at labeling all frames in a video sequence which is sparsely labeled. The challenge for VSS is to keep a balance in exploiting the temporal information (accuracy) and reducing the computational cost (efficiency). Besides, maintaining temporal consistency in adjacent frames is also essential in VSS. To solve these challenges, existing methods can be mainly divided into two categories. The first category [15, 18, 20, 30, 36, 40] concentrates on improving the segmentation accuracy by mining extra information from neighboring unlabeled frames in the video sequence. NetWarp [20] and SVP [36] utilized optical flow to warp features from previous frame to current frame and combine both features to enhance the segmentation accuracy. GRFP [40] proposed a STGRU module which combined optical flow and gated recurrent units (GRU) to fuse spatial and temporal features. EFC [15] jointly trained a VSS and optical flow estimation network to make the two tasks promote each other.

The second category [19, 25, 28, 33, 37, 39, 44, 50, 61] focuses on efficient video segmentation by re-using the feature maps in the neighboring frames. Accel [28] further warped key frame features extracted from a large model, which are combined with shallow features from non-key frame to evaluate the final results. Different from key frame selection methods, Marin *et al.* [39] proposed adaptive key region selection method to promote the segmentation accuracy on small objects and semantic boundaries. TDNet [25] utilized shallow sub-network to encode high-level features from different frames and merged them with an attention module to realize lightweight computation. Inspired by knowledge distillation, ETC [37] distilled a compact segmentation model from a large model. Although these methods took a series of strategies to design an efficient model, there still exists costly overhead when calculating optical flow for flow-based methods and the promotion is limited for distill-based methods. In contrast to the above methods, we introduce a transformer-based method into VSS and propose an efficient transformer architecture to replace the optical flow, in which both efficiency and high accuracy can be achieved.

2.3 Transformer

Transformer, which is mainly based on self-attention mechanism [46], has brought astounding promotions to NLP [12–14, 52]. The breakthrough transformer-based models made in NLP has also attracted considerable interest from CV community. Recently, a number of researchers adapted transformer structure to extensive CV tasks (*e.g.*, object detection [3, 60], image classification [8, 16, 49], image generation and enhancement [4, 10, 42, 51], segmentation [48, 57]). Owing to strong representation capabilities and the reliance on few inductive biases, transformer-based vision networks have acquired remarkable performance and become a viable alternative to convolutional neural network (CNN). Although transformer models succeed in various tasks, their high requirements for memory and computing resources block them for real-time applications or time-consuming tasks (*e.g.*, video processing tasks). To adapt transformer to satisfy these requirements, SSTVOS [17] proposed grid attention

module and strided attention based on criss-cross attention [27] to relieve the computational burden on video object segmentation. Informer [58] leveraged a sparse self-attention mechanism to make efficient time-series forecasting. TransT [9] proposed an efficient attention-based feature fusion module in video object tracking which meets the real-time requirement. Nevertheless, there are no studies explore the usage of transformer for VSS applications (*e.g.*, automatic driving) which require real-time processing and high segmentation accuracy. In this paper, we proposed a STT method tailored for VSS. Our method can achieve state-of-the-art (SOTA) segmentation accuracy with low latency.

3 METHODOLOGY

In this section, we introduce our proposed Sparse Temporal Transformer (STT) for video semantic segmentation (VSS). In Section 3.1, we introduce the problem definition and the framework from a general point of view. In Section 3.2, we elaborate our proposed key selection strategy, query selection strategy, and temporal transformer model. In the remainder of this paper, we use regular letters to represent scalar and bold letters to represent vector, matrix and tensor.

3.1 Problem Definition and Framework Overview

In this paper, we focus on VSS, which aims to assign the semantic labels to each pixel of all frames. Apart from the segmentation accuracy, VSS also requires the segmentation results to be consistent between consecutive frames, which is also known as temporal consistency. Besides, VSS is also a semi-supervised task, where we only have sparse annotations (about 1 frame annotation every 30 frames) from current datasets, like Cityscapes [11] and Camvid [2]. Formally, given a video segmentation dataset $\mathcal{S} = \{(\mathbf{X}, \mathbf{y})\}$, where $\mathbf{X} \in \mathbb{R}^{(T+1)\times H\times W\times 3}$ is a piece of video with H, W, and $T+1$ being height, width, and temporal length respectively. The last frame in video $\mathbf{X}$ is the only frame that has annotated semantic labels $\mathbf{y} \in \mathbb{R}^{H\times W\times c}$, in which c is the total number of semantic categories. During testing, we need to predict the semantic labels for all frames in each video and evaluate the results from two aspects, *i.e.*, segmentation accuracy (mIoU) and temporal consistency [37].

An overall flowchart of our method is illustrated in Figure 2. The input of our method is a video clip containing $T+1$ frames, where the last frame ($\mathbf{F}_{T+1}$) is the target frame for semantic label prediction and the previous frames ($\mathbf{F}_1$ to $\mathbf{F}_T$) will be used by our temporal transformer to enhance the temporal relation. During training, the previous frames ($\mathbf{F}_1$ to $\mathbf{F}_T$) will provide semantic information for the last frame ($\mathbf{F}_{T+1}$) through our temporal transformer and the overall model will also be optimized by all these frames. In the inference stage, all the frames only need to be encoded once and the high-level features can be reused by our our temporal transformer, which prevents the redundant computation and guarantees that semantic segmentation can be done in parallel.

Specifically, our network consists of the following three parts, the shared encoder, the STT, and the segmentation decoder. The shared encoder and the segmentation decoder are the fundamental parts of an image segmentation model, which aim to capture the information within each frame and predict the semantic labels respectively. Specifically, we split the existing segmentation model (*e.g.*, PSPNet [56] or BiSeNet [53]) into encoder and decoder, and then we plug our STT between them, where the STT is proposed to capture temporal related information.

Considering that the time complexity of transformer architecture is extremely high, we design two efficient selection strategies towards the query frame ($\mathbf{F}_{T+1}$) and key frames ($\mathbf{F}_1$ to $\mathbf{F}_T$) of the temporal transformer to simplify the computation. These two selection strategies are capable of reducing the time complexity of our temporal transformer by a large margin without harming the performance of segmentation model (A detailed analysis will be found in Sec. 4.3.1 and Sec. 4.3.2). In the following section, we will detail our STT about the key selection, the query selection, and the temporal transformer.

3.2 Sparse Temporal Transformer

Before introducing the Sparse Temporal Transformer (STT), we first define the variables. The feature maps of previous frames are defined as key feature maps $\mathbf{K} \in \mathbb{R}^{T\times h\times w\times d_t}$ and the feature map of current frame is defined as query feature map $\mathbf{Q} \in \mathbb{R}^{h\times w\times d_t}$, where the T is the number of previous frames, h and w represents the height and width of feature maps, and d_t is the channel size.

3.2.1 Query Selection. Motivated by Marin *et al.* [39] which aims to enhance the representation of semantic boundaries region by content-adaptive downsampling, we also divide the feature map of current frame into simple regions and complex regions. The simple regions mean the regions with monotonous semantic labels, like the red boxes in Figure 1 (a); whereas, the complex regions usually contain multiple semantic objects or contain the boundaries between different semantic objects, like the blue boxes in Figure 1 (a). In experiments (details can be found in Sec. 3.2.1), we find that complex regions contribute more than simple regions to the segmentation accuracy and temporal consistency. To make a trade-off between accuracy and speed, we decide to enhance the segmentation results through temporal transformer while only focusing on relatively complex regions. As we stated previously, the simple regions usually contain unique semantic labels and the complex regions usually contain different semantic labels. To identify complex regions, we propose a novel metric named Neighboring Similarity Matrix (NSM) combining cosine distance and KL divergence to distinguish simple and complex regions in query feature map $\mathbf{Q}$.

Given a neighborhood radius r, a coordinate (u, v) and a query feature map $\mathbf{Q} \in \mathbb{R}^{h\times w\times d_t}$, we can get a feature point $\mathbf{q} = \mathbf{Q}_{[\mathbf{u}][\mathbf{v}]} \in \mathbb{R}^{1\times d_t}$ and its corresponding neighborhood $\mathbf{Q^n} = \mathbf{Q}_{[\mathbf{u-r:u+r}][\mathbf{v-r:v+r}]}$ based on the coordinate (u, v), then we flatten the neighborhood into a 2D feature matrix $\mathbf{Q^n} \in \mathbb{R}^{n_b\times d_t}$, where $n_b = (2r+1)^2$ is the total number of feature points in $\mathbf{Q^n}$. The NSM is defined as:

$$\mathbf{p}_{sim} = \text{SoftMax}(\mathbf{Q^n} \cdot \mathbf{q}^T), \tag{1}$$

$$\mathcal{D}_{KL} = KL(\mathbf{p}_u || \mathbf{p}_{sim}) = \sum_{i=1}^{n_b} p_{u[i]} \log \frac{p_{sim[i]}}{p_{u[i]}}, \tag{2}$$

$$\mathcal{D}_{cos} = \frac{1}{n_b} \sum_{i=1}^{n_b} (1 - \frac{\mathbf{Q^n_{[i]}} \cdot \mathbf{q}^T}{||\mathbf{Q^n_{[i]}}||_2 ||\mathbf{q}||_2}), \tag{3}$$

$$\mathcal{D}_{NSM} = \mathcal{D}_{KL} + \mathcal{D}_{cos}, \tag{4}$$

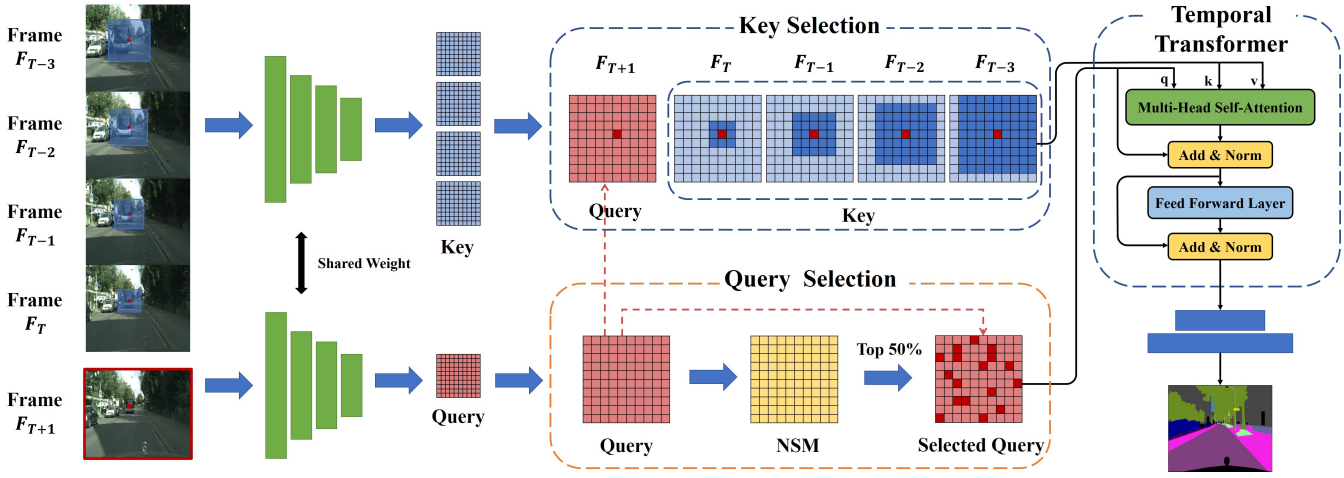

Figure 2: The flowchart of our Sparse Temporal Transformer (STT) method. For an input video clip, we first adopt an encoder to encode all the frames into their corresponding feature maps. Based on the feature map of current frame (query) and the feature maps of previous frames (key), the selected query region in the current frame and selected key regions in the previous frames are determined by our proposed query and key selection strategies. After that, we explore the temporal transformer to capture the temporal relation. Finally, the output of the temporal transformer is sent to a decoder to obtain the segmentation prediction. The NSM is short for Neighboring Similarity Matrix, which is used to identify complex regions.

where the $\mathbf{p}_u \in \mathbb{R}^{n_b \times 1}$ is a uniform distribution. The first term $\mathcal{D}_{KL}$ of NSM measures how close the similarity distribution $\mathbf{p}_{sim}$ is to the uniform distribution $\mathbf{p}_u$. When this term is large, the similarities between each feature point in $\mathbf{Q^n}$ and the feature point $\mathbf{q}$ are quite different and this feature point $\mathbf{q}$ is highly likely in the complex regions. When this term is close to zero, we can only know that the similarities between each feature point in $\mathbf{Q^n}$ and the feature point $\mathbf{q}$ are very close. But the similarities can be either very high or very low. Therefore, the second term $\mathcal{D}_{cos}$ of NSM measures whether the neighborhood $\mathbf{Q^n}$ is similar to the feature point $\mathbf{q}$.

Based on the definition of NSM, our proposed query selection can be summarized as follows: 1) given the query feature map $\mathbf{Q}$ and a neighborhood radius r, we first calculate the NSM for all features points in the query feature map $\mathbf{Q}$; 2) based on the calculated NSM, we select the top-50% feature points with largest $\mathcal{D}_{NSM}$ from the query feature map $\mathbf{Q}$, which constructs the selected query set $\tilde{\mathbf{Q}} \in \mathbb{R}^{\frac{hw}{2} \times 1 \times d_t}$.

3.2.2 Key Selection. Inspired by NetWarp [20] and EFC [15] which show that tracking the corresponding small regions in previous frames can bring much useful temporal information to the current frame, we propose to identify a small key region from every key feature map for each selected query point. As we can see in Figure 1 (b), if we want to track the person and bike in the key frames, it is unnecessary to search the whole frame of every key frames. Instead, a more reasonable way is to enlarge the searching regions gradually from near frame to far frame. Therefore, we design our key selection strategy following two rules: 1) the key frame farther from the current frame should have larger key region; 2) the size of key regions should vary within a proper range.

Based on the above two rules, we design a simple and efficient key selection strategy, which is decided by three hyper-parameters, *i.e.*, start size s, end size e, and expansion coefficient ϵ. Formally, the radius l_t of key region in t-th key feature map $\mathbf{K}_t$ is defined as:

$$l_t = \begin{cases} s + (T-t) * \epsilon & , \quad \text{if} \quad s + (T-t) * \epsilon < e; \\ e & , \quad \text{otherwise.} \end{cases} \tag{5}$$

For the t-th ($t \in [1, 2, .., T]$) key feature map $\mathbf{K}_t$ ($\mathbf{K}_1$ represents the farthest key feature map and $\mathbf{K}_T$ represents the nearest key feature map), the size of the key region will be $(2l_t + 1)^2$ and the center of the key region is decided by the coordinate of query point. Based on the key selection strategy, the key selection can be summarized as: 1) for each key feature map $\mathbf{K}_t$, we first calculate its corresponding radius $l_t \; \forall \; t \in [1, ..., T]$ of key region; 2) for each query point in the selected query set $\tilde{\mathbf{Q}}$, we identify its corresponding key region in each key feature map $\mathbf{K}_t$ based on the radius of key region l_t and center coordinate (u, v). 3) we aggregate all the key regions over T key frames for each query point to acquire the selected key set $\tilde{\mathbf{K}} \in \mathbb{R}^{\frac{hw}{2} \times n_k \times d_t}$, where the accumulated key region size $n_k = \sum_{t=1}^{T} (2l_t + 1)^2$ and $n_k < T(2e + 1)^2$.

After key selection, for each query point in the selected query set $\tilde{\mathbf{Q}}$, the size of key set is reduced from $O(Thw)$ to $O(Te^2)$, where $e^2 \ll hw$.

3.2.3 Temporal Transformer. The structure of the our temporal transformer encoder is shown in Figure 2, which has a multi-head attention layer and a feed-forward layer along with residual connection and layer normalization. For our temporal transformer, we adapt the transformer architecture as described by Vaswani *et al.* [46] with key and query as different features to fit into the requirement of VSS, *i.e.* enabling the current frame to capture the temporal relation from previous frames. Besides, in our setting, the key and value is the selected key set $\tilde{\mathbf{K}} \in \mathbb{R}^{\frac{hw}{2} \times n_k \times d_t}$ and the selected query set $\tilde{\mathbf{Q}} \in \mathbb{R}^{\frac{hw}{2} \times 1 \times d_t}$ respectively.

Method	Backbone	Cityscapes			Camvid		
		mIoU (%) ↑	TC (%) ↑	fps (frame/s) ↑	mIoU (%) ↑	TC (%) ↑	fps (frame/s) ↑
NetWarp [20]	ResNet101	80.6	-	0.3	67.1	-	2.8
DFF [61]	ResNet101	68.7	71.4	9.7	-	-	-
GRFP [40]	ResNet101	69.4	-	3.2	66.1	-	4.4
LVS [33]	ResNet101	76.8	-	5.9	-	-	-
Accel [28]	ResNet101/18	72.1	70.3	3.6	66.7	-	7.6
PSPNet18 [56]	ResNet18	75.5	68.5	10.8	71.0	-	24.4
PSPNet50 [56]	ResNet50	78.1	-	4.2	74.7	-	8.5
PSPNet101 [56]	ResNet101	79.4	69.7	2.1	77.6	77.1	4.1
TDNet-PSP18 [25]	ResNet18	76.8	70.4	11.8	72.6	73.2	25.2
TDNet-PSP50 [25]	ResNet50	79.9	71.1	5.6	76.0	77.4	11.1
ETC-PSP18 [37]	ResNet18	73.1	70.6	10.8	75.2	77.3	24.4
ETC-PSP101 [37]	ResNet101	79.5	71.7	2.1	79.4	78.6	4.1
STT-PSP18	ResNet18	**77.3**	**73.0**	11.5	**76.1**	**81.4**	24.7
STT-PSP101	ResNet101	**82.5**	**73.9**	2.2	**80.2**	**82.3**	4.2

Table 1: We compare our methods with previous High-Quality Methods on both Cityscapes and Camvid. ↑ represents higher value is better.

The multi-head attention is the core of our temporal transformer, which is a dense operator that allows each query feature point in selected query set to interact with its corresponding key features in selected key set. In VSS, we use the multi-head attention to capture long-range dependencies without recurrence, and it can be viewed intuitively as a cross-correlation operator that uses CNN features to capture the temporal relation within a period. Given the input selected key set ($\tilde{\mathbf{K}}$) and selected query set ($\tilde{\mathbf{Q}}$), which contains $\frac{hw}{2}$ query point and n_k key points associated with each query point, the multi-head attention can be formulated as:

$$\mathbf{H}_j = \text{SoftMax}(\frac{(\tilde{\mathbf{Q}}\mathbf{W}_j^Q)(\tilde{\mathbf{K}}\mathbf{W}_j^K)^T}{\sqrt{\hat{d}}}(\tilde{\mathbf{K}}\mathbf{W}_j^V)), \tag{6}$$

$$MH(\tilde{\mathbf{Q}}, \tilde{\mathbf{K}}) = [\mathbf{H}_1, ...\mathbf{H}_j, ..., \mathbf{H}_{\mathbf{n_h}}]\mathbf{W}^O, \tag{7}$$

where $\mathbf{W}_j^Q \in \mathbb{R}^{d_t \times \hat{d}}$, $\mathbf{W}_j^K \in \mathbb{R}^{d_t \times \hat{d}}$, $\mathbf{W}_j^V \in \mathbb{R}^{d_t \times \hat{d}}$ are projection matrices for j-th attention head with $\hat{d} = \frac{d_t}{n_h}$, $\mathbf{W}^O \in \mathbb{R}^{d_t \times d_t}$, $[\cdot, ..., \cdot]$ represents concatenation, and $MH()$ is short for multi-head attention. The temporal transformer encoder is then formulated as:

$$\mathbf{X} = LN(\tilde{\mathbf{Q}} + MH(\tilde{\mathbf{Q}}, \tilde{\mathbf{K}})), \tag{8}$$

$$FFN(\mathbf{X}) = \max(0, \mathbf{X}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{9}$$

$$TFE(\tilde{\mathbf{Q}}, \tilde{\mathbf{K}}) = LN(\mathbf{X} + FFN(\mathbf{X})), \tag{10}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_t \times \bar{d}}$, $\mathbf{b}_1 \in \mathbb{R}^{1 \times \bar{d}}$, $\mathbf{W}_2 \in \mathbb{R}^{\bar{d} \times d_t}$, $\mathbf{b}_2 \in \mathbb{R}^{1 \times d_t}$, $LN()$, $FFN()$ and $TFE()$ are short for layer normalization, feed-forward layer and transformer encoder.

For a vanilla temporal transformer (*i.e.*, without the query selection and key selection strategy), the computational complexity is $O(Th^2w^2)$, where all the key points in the key feature maps interact with all the query points in the query feature map. After incorporating our proposed query selection and key selection strategies into the temporal transformer, the time complexity of our STT is successfully reduced to $O(Thwe^2)$, where e is a constant in our experiment. To further reduce the computation of transformers, we also reduce the feature dimension of the key, value and query from d_t to $\frac{d_t}{4}$ with a multi-layer perceptron. Therefore, the final time complexity of our model is $O(Thw)$.

Method	Backbone	mIoU (%) ↑	TC (%)↑	fps (frame/s) ↑
DVSNet [50]	ResNet18	63.2	-	30.3
ICNet [55]	ResNet50	67.7	-	50.0
LadderNet [31]	DenseNet121	72.8	-	30.3
SwiftNet [41]	ResNet18	75.4	-	43.5
BiSeNet18 [53]	ResNet18	73.8	-	50.0
BiSeNet34 [53]	ResNet34	76.0	-	37.0
TDNet-BiSe18 [25]	ResNet18	75.0	70.2	47.6
TDNet-BiSe34 [25]	ResNet34	76.4	71.1	38.5
ETC-Mobi [37]	MobileNetV2	73.9	69.9	20.8
STT-BiSe18	ResNet18	**75.8**	**71.4**	44.2
STT-BiSe34	ResNet34	**77.3**	**72.0**	33.8

Table 2: We compare our method with previous High-Speed Methods on Cityscapes. ↑ represents higher value is better.

4 EXPERIMENT

4.1 Experiment Setup

4.1.1 Dataset. We evaluate our method and all the other baselines on two benchmark video semantic segmentation datasets: Cityscapes [11] and Camvid [2]. Detailed introduction of these two datasets can be found in supplementary.

4.1.2 Evaluation Metrics. To compare our proposed method with SOTA methods from both segmentation accuracy and temporal consistency, we adopt the same evaluation metrics on both datasets as ETC [37]. Specifically, for segmentation accuracy, we adopt the mean Intersection-over-Union (mIoU). for temporal consistency, we employ TC following ETC [37], which measures the consistency based on the mean flow warping error between all consecutive frames (more details about the TC metric can be found in ETC [37]).

4.1.3 Models and Baselines. We demonstrate the effectiveness of Sparse Temporal Transformer (STT) on different backbones. We select two SOTA image segmentation models for our experiments: PSPNet [56] and BiSeNet [53]. For the latter method, we compare

	SS (s)	ES (e)	EC (ϵ)	key size	mIoU (%)	TC (%)	fps (frame/s)
1	1	5	1	527	77.2	73.0	11.5
2	2	5	1	639	77.3	72.9	11.1
3	3	5	1	735	77.1	73.0	10.7
4	1	3	1	279	76.5	72.1	11.9
5	1	7	1	679	77.3	72.8	11.0
6	1	5	2	663	77.1	72.8	11.0
7	1	5	3	695	77.2	72.9	11.0
8	-	-	-	57344	75.1	69.9	0.2

Table 3: The effect of key selection strategy. The SS, ES, EC represent the start size s, the end size e, and the expansion coefficient ϵ, respectively. The key size means the accumulated key region size, which is derived from SS, ES and EC. - in the last row indicates that we use all the key features as the selected keys.

	NR (r)	TR	mIoU (%)	TC (%)	fps (frame/s)
1	1	50 %	76.1	71.2	11.5
2	3	50 %	77.1	72.8	11.5
3	5	50 %	77.3	73.0	11.5
4	7	50 %	77.2	72.9	11.5
5	9	50 %	76.8	72.1	11.5
6	5	0 %	75.3	68.7	13.6
7	5	25 %	76.7	72.4	12.6
8	5	75 %	77.3	72.7	10.5
9	5	100 %	77.2	72.9	9.4

Table 4: The effect of query selection strategy. NR and TR represent the neighborhood radius and selection top-ratio.

with the modified version (BiSeNet*), which is from TDNet [25] and claimed to have higher efficiency and better training convergence. For these two image segmentation backbones, we extend them by plugging our STT module between the encoder part and decoder part. We followed TDNet [25] to split the encoder part and decoder part.

4.2 Comparison with Existing Methods

To verify the effectiveness of our method, we compare it with the SOTA methods on Cityscapes [11] and Camvid [2]. Following TDNet[25], the compared methods can be divided into two groups, one group (High-Speed Methods) has strength in inference speed while the other group (High-Quality Methods) has better segmentation accuracy. We compare our method with two groups of methods to prove that our method can achieve high segmentation accuracy with comparable inference speed.

4.2.1 Comparison with High-Quality Methods. Considering high segmentation accuracy (mIoU) and temporal consistency (TC), we choose PSPNet [56] as our base model. We evaluate our STT method based on PSPNet18 and PSPNet101 respectively, and compare them with NetWarp [20], DFF [61], GRFP [40], LVS [33], Accel [28], PSPNet [56], TDNet [25], and ETC [37]. The results tested on Cityscapes validation set and Camvid testset are all summarized in Table 1. It can be seen that all our STT based models outperform other High-Quality Methods on segmentation accuracy metrics (mIoU and TC). The TC of TDNet-PSP18 and TDNet-PSP50 is measured based on their released code and trained parameters. The TC and mIoU of other baselines are directly copied from ETC [37] or TDNet [25]. Higher performance will be selected when the experiment results of baselines are reported differently in these two paper. Among them, our STT-PSP101 model obtains the highest segmentation accuracy (mIoU and TC), in which mIoU is 82.5% and TC is 73.9%. Note that although NetWarp has the second highest mIoU, it has the lowest fps which is far lower than other methods. Benefiting from the usage of temporal transformer, our method can capture the temporal relation among consecutive frames effectively without the optical flow estimation error existing in flow-based methods.

4.2.2 Comparison with High-Speed Methods. Concentrating on model efficiency (mIoU and fps), we choose BiSeNet [56] as our base model. We evaluate our STT method based on BiSeNet18 and BiSeNet34 respectively, and compare them with DVSNet [50], ICNet [55], LadderNet [31], SwiftNet [41], BiSeNet [53], TDNet [25], and ETC [37]. The results tested on Cityscapes validation set are all summarized in Table 2. The TC of TDNet-BiSe18 and TDNet-BiSe34 is measured based on their released code and trained parameters. The TC and mIoU of other baselines are directly copied from ETC [37] or TDNet [25]. It can be seen our STT method has an advantage over other High-Speed Methods on mIoU while also has a comparable inference speed. Our STT-BiSe34 model obtains best performance over other methods, which mIoU is 77.3% and fps is 33.8 frame/s. We realize a better balance on segmentation accuracy and inference speed compared to other High-Speed Methods. Based on key selection and query selection, we effectively reduce the time complexity in transformer from $O(Th^2w^2)$ to $O(Thw)$ which brings great promotion to our model on inference time.

4.3 Ablation Study

In this section, we study the effect of key selection and query selection. All the experiments in this section are conducted on the Cityscapes [11] dataset with the STT-PSP18. More ablation studies about the effect of the number of key frames can be found in supplementary.

4.3.1 Effect of key selection. In this section, we study the effect of different key selection strategy, where the start size, end size and the expansion coefficient represent the minimal searching region for the nearest frame, the maximum searching region for the farthest frame and how fast we enlarge the searching region, respectively. Table 3 shows the effect of these three hyper-parameters. From row 1, 2 and 3, we can find that starting from a larger searching region does not improve the segmentation accuracy or temporal consistency significantly, however, the inference speed gets slower. From row 1, 4 and 5, we can find that 5 is a better choice for the maximum search regions, since smaller maximum search regions (row 4) may filter out some semantic objects and larger choice (row 5) would harm the inference speed without much improvement. From row 1, 6 and 7, we find that expansion coefficient does not affect the segmentation accuracy or temporal consistency significantly. For the concern of speed, we choose 1 as the expansion coefficient in our experiment. Besides, comparing the first row and the last row, we find that all evaluation metrics drop without our key selection

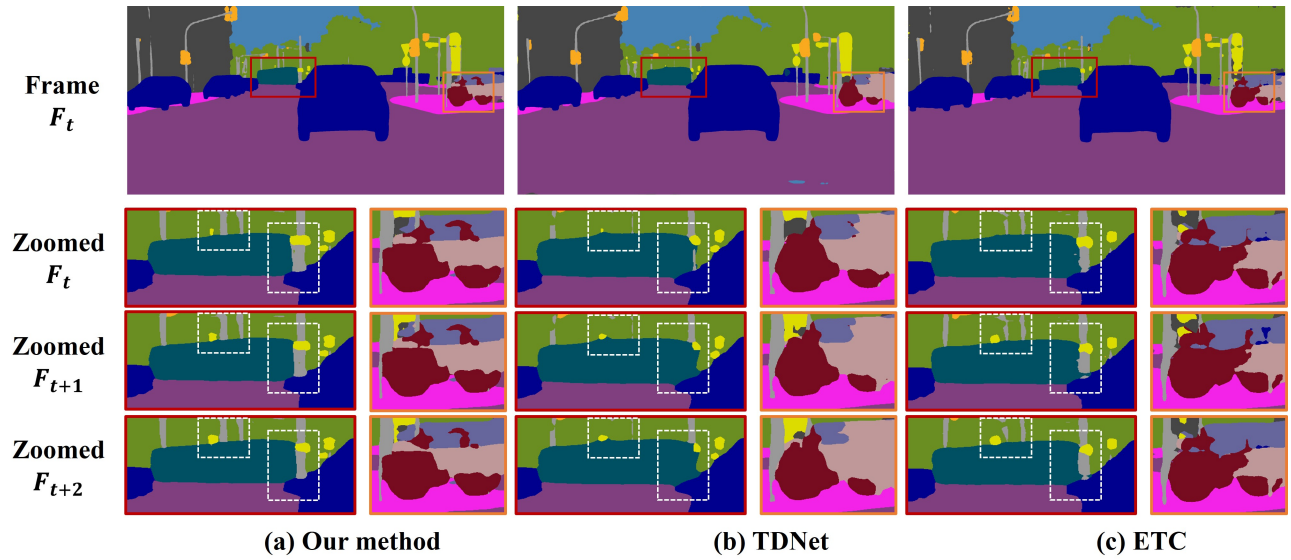

Figure 3: The segmentation results of our method compared with two baseline methods, *i.e.*, TDNet [25] and ETC [37]. On the top part, we show full-size segmentation results of frame F_t. For better visualization, we zoom the region in the red and orange box across 3 frames in the bottom part of figure. In the red boxes, our model is able to generate more consistent semantic label to the edge of the streetlight and moving trolleybus. In the orange boxes, our model is able to generate more fine-grained and stable results (*e.g.*, bicycle) across frames compared with the baseline methods.

strategy. For the performance (mIoU and TC) drop, we suspect that the huge key size prevents the model from getting converged within 80k iterations. Besides, the huge key size also introduces lots of noise from dissimilar regions. For the speed (fps) drop, the key size in the last row is one hundred times larger than that in the first row, which will surely decrease the inference speed significantly.

4.3.2 Effect of query selection. In this section, we study the effect of different query selection strategies. The neighborhood radius (NR) represents the neighborhood size to be considered when selecting the query features. The selection top-ratio (TR) represents the area ratio in each frame which should be considered as complex region. The results are summarized in Table 4. From row 1-5, we find that choosing the neighboring radius in a proper region (*e.g.*, 3-7) will not affect the query selection largely. However, if the neighborhood radius is too small (1) or too large (9), complex region and simple region may also be confused, which will weaken the representation of complex regions. From row 1, 6, 7 and 8, we can find that the more features we selected from the query feature map $\mathbf{Q}$, the better segmentation accuracy and temporal consistency we will get. Whereas, when the selection top-ratio is beyond 50%, we find that the improvement in segmentation accuracy and temporal consistency is quite limited. To balance the performance and inference speed, we choose top-50% query feature from the query feature map ($\mathbf{Q}$) as the selected query set ($\tilde{\mathbf{Q}}$).

4.4 Case Study

In this section, we show the segmentation results of our method. More cases about the visualization of attention map between selected query and key and the query selection is in supplementary.

We select two representative baselines (TDNet [25] and ETC [37]) for qualitative comparisons. Following TDNet [25] and ETC [37], we employ all three segmentation models based on PSPNet18 [56] for visualization in Fig 3. It can be seen our proposed method can assign more consistent labels to the moving objects and generate more accurate segmentation results. More visualization results can be found in the supplementary.

5 CONCLUSION

In this paper, we have studied the video semantic segmentation from a new viewpoint, *i.e.*, capturing the temporal relation across frames by temporal transformer, where the current frame and previous frames are taken as the query and key. To balance the segmentation performance and inference speed, we have proposed two feature selection strategies, *i.e.*, query selection and key selection to reduce the time complexity of our temporal transformer significantly. Comprehensive experiments on two benchmark datasets have demonstrated that our method remarkably outperforms the SOTA approaches with comparable inference speed.

ACKNOWLEDGMENTS

The work is supported by the National Key R&D Program of China (2018AAA0100704) and is partially sponsored by National Natural Science Foundation of China (Grant No.61902247) and the Shanghai Science and Technology RD Program of China (20511100300). This work is also sponsored by Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). This work is also supported by SenseTime Collaborative Research Grant.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39, 12 (2017), 2481–2495.

[2] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. 2008. Segmentation and recognition using structure from motion point clouds. In *ECCV 2008*.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *ECCV 2020*.

[4] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. 2020. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364* (2020).

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

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40, 4 (2017), 834–848.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV 2018*.

[8] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *ICML 2020*.

[9] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. 2021. Transformer Tracking. *arXiv preprint arXiv:2103.15436* (2021).

[10] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509* (2019).

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR 2016*.

[12] Gonçalo M Correia, Vlad Niculae, and André FT Martins. 2019. Adaptively sparse transformers. In *EMNLP 2019*.

[13] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL 2019*.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL 2018*.

[15] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. 2020. Every frame counts: joint learning of video segmentation and optical flow. In *AAAI 2020*.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR 2021*.

[17] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. 2021. SSTVOS: Sparse Spatiotemporal Transformers for Video Object Segmentation. In *CVPR 2021*.

[18] Mohsen Fayyaz, Mohammad Hajizadeh Saffar, Mohammad Sabokrou, Mahmood Fathy, Fay Huang, and Reinhard Klette. 2016. STFCN: spatio-temporal fully convolutional neural network for semantic segmentation of street scenes. In *ACCV 2016*.

[19] Junyi Feng, Songyuan Li, Xi Li, Fei Wu, Qi Tian, Ming-Hsuan Yang, and Haibin Ling. 2020. TapLab: A Fast Framework for Semantic Video Segmentation Tapping into Compressed-Domain Knowledge. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[20] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. 2017. Semantic video cnns through representation warping. In *ICCV 2017*.

[21] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. 2020. Context-aware Feature Generation for Zero-shot Semantic Segmentation. In *ACM MM 2020*. 1921–1929.

[22] Hao He, Xiangtai Li, Kuiyuan Yang, Guangliang Cheng, Jianping Shi, Yunhai Tong, Zhengjun Zha, and Lubin Weng. 2021. BoundarySqueeze: Image Segmentation as Boundary Squeezing. *arXiv preprint arXiv:2105.11668* (2021).

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR 2016*.

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. In *CVPR 2017*.

[25] Ping Hu, Fabian Caba, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. 2020. Temporally distributed networks for fast video semantic segmentation. In *CVPR 2020*.

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *CVPR 2017*.

[27] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. 2019. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*.

[28] Samvit Jain, Xin Wang, and Joseph E Gonzalez. 2019. Accel: A corrective fusion network for efficient semantic segmentation on video. In *CVPR 2019*.

[29] Wei Ji, Xi Li, Fei Wu, Zhijie Pan, and Yueting Zhuang. 2019. Human-centric clothing segmentation via deformable semantic locality-preserving network. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 12 (2019), 4837–4848.

[30] Xiaojie Jin, Xin Li, Huaxin Xiao, Xiaohui Shen, Zhe Lin, Jimei Yang, Yunpeng Chen, Jian Dong, Luoqi Liu, Zequn Jie, et al. 2017. Video scene parsing with predictive feature learning. In *ICCV 2017*.

[31] Ivan Kreso, Sinisa Segvic, and Josip Krapac. 2017. Ladder-style densenets for semantic segmentation of large natural images. In *ICCV Workshops 2017*. 238–245.

[32] Xiangtai Li, Hao He, Xia Li, Duo Li, Guangliang Cheng, Jianping Shi, Lubin Weng, Yunhai Tong, and Zhouchen Lin. 2021. PointFlow: Flowing Semantics Through Points for Aerial Image Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4217–4226.

[33] Yule Li, Jianping Shi, and Dahua Lin. 2018. Low-latency video semantic segmentation. In *CVPR 2018*.

[34] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. 2017. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR 2017*.

[35] Lizhao Liu, Junyi Cao, Minqian Liu, Yong Guo, Qi Chen, and Mingkui Tan. 2020. Dynamic Extension Nets for Few-shot Semantic Segmentation. In *ACM MM 2020*. 1441–1449.

[36] Si Liu, Changhu Wang, Ruihe Qian, Han Yu, Renda Bao, and Yao Sun. 2017. Surveillance video parsing with single frame supervision. In *CVPR 2017*.

[37] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. 2020. Efficient Semantic Video Segmentation with Per-frame Inference. In *ECCV 2020*.

[38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR 2015*.

[39] Dmitrii Marin, Zijian He, Peter Vajda, Priyam Chatterjee, Sam Tsai, Fei Yang, and Yuri Boykov. 2019. Efficient segmentation: Learning downsampling near semantic boundaries. In *ICCV 2019*.

[40] David Nilsson and Cristian Sminchisescu. 2018. Semantic video segmentation by gated recurrent flow propagation. In *CVPR 2018*.

[41] Marin Orsic, Ivan Kreso, Petra Bevandic, and Sinisa Segvic. 2019. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *CVPR 2019*.

[42] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *ICML 2018*.

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI 2015*.

[44] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. 2016. Clockwork convnets for video semantic segmentation. In *ECCV 2016*.

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR 2015*.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS 2017*.

[47] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. 2020. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020).

[48] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. 2021. End-to-End Video Instance Segmentation with Transformers. In *CVPR 2021*.

[49] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. 2020. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677* (2020).

[50] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. 2018. Dynamic video segmentation network. In *CVPR 2018*.

[51] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. 2020. Learning texture transformer network for image super-resolution. In *CVPR 2020*.

[52] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS 2019*.

[53] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV 2018*.

[54] Fisher Yu and Vladlen Koltun. 2016. Multi-scale context aggregation by dilated convolutions. In *ICLR 2016*.

[55] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. 2018. Icnet for real-time semantic segmentation on high-resolution images. In *ECCV 2018*.
[56] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *CVPR 2017*.
[57] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. 2021. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In *CVPR 2021*.
[58] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *AAAI 2021* (2021).
[59] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. 2018. Towards high performance video object detection. In *CVPR 2018*.
[60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR 2021*.
[61] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Deep feature flow for video recognition. In *CVPR 2017*.