Dual Correlation Network for Efficient Video Semantic Segmentation

Shumin An, Qingmin Liao*, Zongqing Lu, *Member, IEEE* and Jing-Hao Xue, *Senior Member, IEEE*

***Abstract*—Video data bring a big challenge to semantic segmentation due to the large volume of data and strong inter-frame redundancy. In this paper, we propose a dual local and global correlation network tailored for efficient video semantic segmentation. It consists of three modules: 1) a local attention based module, which measures correlation and achieves feature aggregation in a local region between key frame and non-key frame; 2) a consistent constraint module, which considers long-range correlation among pixels from a global view for promoting intra-frame semantic consistency of non-key frame; and 3) a key frame decision module, which selects key frames adaptively based on the ability of feature transferring. Extensive experiments on the Cityscapes and Camvid video datasets demonstrate that our proposed method could reduce inference time significantly while maintaining high accuracy. The implementation is available at https://github.com/An01168/DCNVSS.**

***Index Terms*—Video semantic segmentation, local attention, consistent constraint, key frame selection**

I. Introduction

SEMANTIC segmentation is a fundamental task of computer vision. With the aim of predicting labels for each pixel in an image, semantic segmentation could facilitate the perception and understanding of scene. It has been widely used in a variety of applications, e.g. autonomous driving [1], [2], urban management [3], robot vision [4].

With the proposal of FCN [5], semantic segmentation can be achieved by an end-to-end deep convolutional neural network. Many subsequent approaches [6]–[13] have been proposed to make further improvements. Although these methods achieve significant performance on image data, they have limited performance on video data, without considering the temporal information among frames.

Video signal is easy to be obtained in daily life, such as from autonomous driving and road monitoring. Methods designed for video processing have drawn increasing attention recently. However, semantic segmentation of video data remains a challenging problem. Compared with image data, video produces a much larger volume of data, e.g. 30 frames per second in Cityscapes [1]. The large volume of data would bring big computing and storage burden to systems.

To increase efficiency, there have been several methods [14], [15], [17], [19] specially designed for video semantic segmentation. The common way is to accelerate segmentation speed via reducing computing redundancy, e.g. DFF [17], LVSS [15], DAVSS [14]. They utilize the similar characteristic of consecutive frames in video, getting the feature map of current frame by propagating features from previous frames. Although these methods could increase efficiency, they need to estimate optical flow [24], [25] or add an extra network module to achieve the propagating process, which is time-consuming. Clockwork [19] adopts the high-level feature map of previous frame into current frame straightly, which further reduces inference time, but this makes the segmentation accuracy unsatisfactory due to the difference between these two frames. Moreover, the above methods only consider the inter-frame propagating, but ignore the correlation among pixels in the same frame, which reduces the accuracy of semantic segmentation.

Therefore, in this paper, to address big redundancy of video

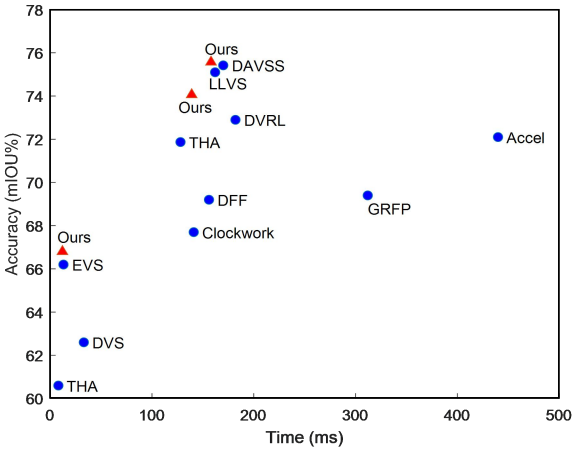


Fig. 1: Accuracy and inference time of various video semantic segmentation methods on the Cityscapes val dataset. Methods include DAVSS [14], LLVS [15], Accel [16], DFF [17], GRFP [18], Clockwork [19], THA [20], EVS [21], DVS [22], DVRL [23] and Ours. Red triangles denote our methods and blue dots denote other methods. In ascending order of inference time, our methods are performed on three conditions: adopting fixed time schedule on lightweight basic model, adopting fixed time schedule and KDM schedule separately on large basic model; while THA are performed on two conditions: adopting fixed time schedule on lightweight basic model and large basic model separately. Our methods achieve the highest accuracy while maintaining competitive inference speed on both lightweight and large basic models.

Corresponding author: Qingmin Liao
S. An, Q. Liao, Z. Lu are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China, and also with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China
J.-H. Xue is with the Department of Statistical Science, University College London, London WC1E 6BT, UK

data and the lack of considering intra-frame correlation, we propose a dual local and global correlation network by leveraging two complementary modules: a *local attention based module*, which propagates high-level feature of previous frame in a local and non-parametric way, for increasing efficiency; and a *consistent constraint* module, which considers long-range correlation among pixels from a global view, for promoting intra-frame semantic consistency and increasing accuracy.

The local attention based module selectively aggregates high-level semantic information from previous frame to current frame. The aggregating weight is determined by the similarity between two positions in low-level features of two frames. It does not rely on any flow network or extra network module, and thus can produce a highly efficient propagating process. The consistent constraint module is motivated by knowledge distillation [26]–[29], and the goal is to promote semantic consistency of the feature of non-key frame, via obtaining global context information from the feature that is produced from same frame but in a key-frame segmentation framework. It could increase the accuracy of non-key frame effectively, without increasing any inference loads.

In addition, the scene is changing in the video, where objects sometimes move violently, sometimes smoothly. Hence it is reasonable to adaptively change the frequency of key frame selection based on the situation of scene, e.g., selecting key frames frequently when scene changes dramatically, and infrequently when scene changes regularly. Therefore, different from the common way to select key frames in a fixed interval in [14], [16], [17], [20], this paper proposes an adaptive key frame selection strategy, which takes the propagating ability of the proposed dual correlation network as criterion. We define potential prediction error (PPE) to measure this propagating ability from key frame to current frame. The higher the PPE is, the worse the propagating performance is, and the more possibility the current frame is regarded as key frame. With the proposed key frame selection strategy, we are able to utilize computational resources more efficiency, or further improve the segmentation performance given a fixed number of key frames.

Fig. 1 shows several comparisons of recent efficient video semantic segmentation methods on accuracy and inference time. We can see that our method achieves the highest accuracy while maintaining competitive inference speed.

In summary, our main contributions are as follows:

- We propose a tailored approach to efficient video semantic segmentation by leveraging two complementary modules for considering both efficiency and accuracy: a *local attention based module* and a *consistent constraint module*.
- The local attention based module gets the feature of current frame via propagating the high-level feature from previous frame in a local and non-parametric way, for high segmentation efficiency. The consistent constraint module considers the long-range correlation among pixels within current frame, for promoting semantic consistency and increasing accuracy, without increasing any inference burden.
- In addition, we also propose a novel adaptive key frame selection strategy based on the ability of feature transferring, which is capable of saving computational resources and further improving the segmentation performance.

In the rest of this paper, Section II provides a preview of recent research in image and video semantic segmentation. Section III details the proposed method, which includes local attention based module, consistent constraint module, and key frame selection strategy. Section IV presents experiments and analysis. Section V makes a summary of this paper.

II. Related Work

A. Image Semantic Segmentation

Most existing works on semantic segmentation are at the image level, and they can be regarded as the basis of video-based tasks. FCN [5] proposes an end-to-end framework utilizing deep neural network for image semantic segmentation, and achieves remarkable performance. Many subsequent approaches have been proposed based on FCN, for improving accuracy or efficiency. In [6], [7], [11], [30], [31], muti-scale features are obtained to enlarge receptive field via dilated convolutions or skip connection. In [32], the structural information of objects is extracted based on self-attention mechanism to increase the segmentation accuracy. Strong backbones, e.g. Resnet [33], GoogleNet [34] and DenseNet [35], are exploited to get high segmentation accuracy. To gain context information, ReSeg [36] and DAG-RNN [37] make use of recurrent neural network; [9], [10], [38], [39] utilize attention mechanism; SETR [40] and SegFormer [41] adopt transformer. They are able to model relationship among pixels and get good segmentation performance. By means of lightweight models, e.g. SFANet [42], LEDNet [43] and BiseNet [44], high efficiency could be achieved. However, these methods do not consider the temporal information among frames, which is crucial for video-based tasks, and have a limited performance on video semantic segmentation. Therefore, it is meaningful to develop methods tailored for video semantic segmentation.

B. Video Semantic Segmentation

Video semantic segmentation can be regarded as an extension of image semantic segmentation. Unlike image data, video data have the characteristics of large amount, strong inter-frame redundancy and sparse annotations. Based on how to exploit temporal information, existing video segmentation methods could be mainly divided into two groups. One group is to improve accuracy via obtaining extra information from adjacent frames. For example, [18], [45]–[47] adopt optical flow to warp the features of neighboring unlabelled frames to current frame, and fuse multi-frame features via RNN or temporal constraint. [48]–[50] utilize LSTM or 3D convolution to model temporal information and gain spatio-temporal feature. [51], [52] build video prediction models to predict the labels of future frame, and those labels can be regarded as new samples or supplementary information to help training. TMANet [53] and LMANet [54] use attention mechanism to read relevant semantic information from previous frames, for making features more representative. SSVOS [55] uses a

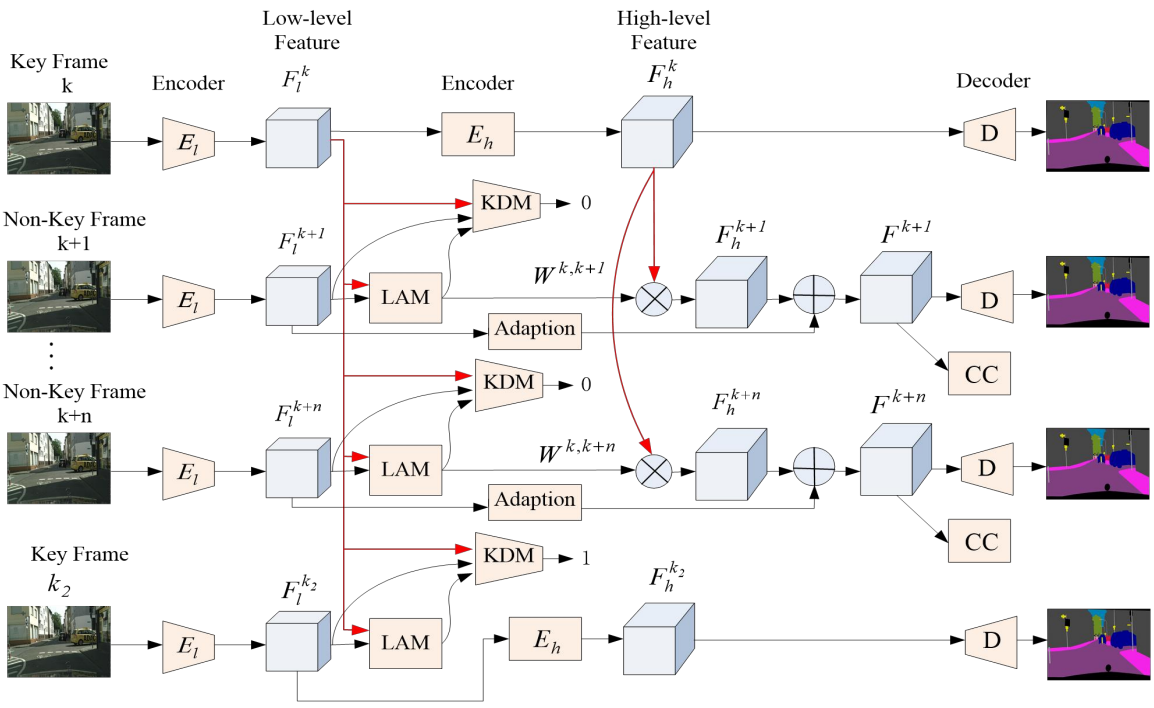


Fig. 2: The diagram of proposed framework. The time index of the first key frame is k, its low-level feature F_l^k and high-level feature F_h^k are extracted from lower part of the encoder E_l and higher part of the encoder E_h. For the later time index $k+n$, $(n>0)$, the low-level feature F_l^{k+n} is firstly extracted. Based on both F_l^{k+n} and F_l^k, the local weight coefficient $W^{k,k+n}$ is calculated via local attention based module (LAM). Then based on F_l^{k+n}, F_l^k and $W^{k,k+n}$, the key frame decision module (KDM) will decide if the $k+n$ frame should be regarded as a new key frame. If the decision is yes, the high-level feature F_h^{k+n} is obtained from higher part of the CNN E_h; otherwise, F_h^{k+n} is calculated by propagating from F_h^k with $W^{k,k+n}$, and the overall feature F^{k+n} is derived by combining F_h^{k+n} with the adaption of low-level feature F_l^{k+n} and the supervision of consistent constraint module (CC). Finally, a decoder D is applied on the high-level feature, which is obtained in either way, to get segmentation results.

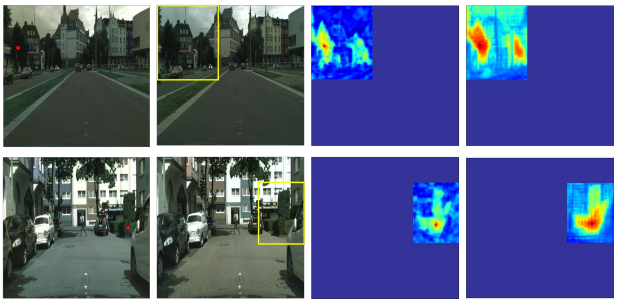
scribble attention module and a CRF based regularized loss to make features get more accurate context information and improve segmentation accuracy under scribble-level supervision condition. SPR [56] proposes a bidirectional propagation process, and builds forward and backward dictionaries to model inter-frame relationship. However, these methods need to build extra modules on the top of high-level features to combine temporal information, which is low in efficiency and hard to satisfy the real-time requirements of practical application.

The other group is to improve efficiency via reducing computing redundancy. Clockwork [19] adopts high-level feature of previous frame into current frame straightly, which reduces inference time significantly, but the misalignment of these two frames makes the segmentation accuracy unsatisfactory. ETC [57] considers temporal consistency in the training stage, and segments each frame independently in the inference stage. It is able to increase the accuracy, but cannot accelerate the inference speed. DFF [17], DVSNet [22] and DAVSS [14] warp the feature of previous frame to current frame with the aid of optical flow. However, it is time consuming to estimate the optical flow, and the evaluated error would cause bad effect on the warping process. Instead of using optical flow, LLVS [15] builds a weight prediction network for propagating feature, which achieves good performance on large basic model, but has less advantage on lightweight basic model since the time used for weight prediction is non-negligible. THA [20] performs feature propagation via holistic attention, and achieves very fast inference speed, e.g. 131 fps on the Cityscapes dataset. However, it only considers feature transforming on one-to-one same position between key frame and non-key frame, which is unreasonable as the objects keep on moving. Hence, the accuracy of THA is unsatisfactory.

Different from the one-to-one pattern, this paper considers the correlation between one position in non-key frame and the local area with the center at the corresponding position in key-frame, by means of local attention mechanism. Our proposal is able to achieve high efficiency while keeping good accuracy on both large and lightweight basic networks.

III. Proposed Method

In this section, we first give an overview of the whole framework, and then elaborate three modules respectively.



(a) image1 (b) image2 (c) low-level (d) high-level

Fig. 3: The local correlation map of low-level and high-level features on two adjacent frames.

A. Framework Overview

In this paper, we adopt the common paradigm in video processing, propagating feature from key frame to non-key frame by considering inter-frame similarity. For key frame, the segmentation result is obtained from main model, which is the same as that in image segmentation, with high accuracy but slow inference speed. For non-key frame, the high-level feature is propagated from key frame, which costs much less computing time than producing it from scratch.

There are two crucial problems in this framework: how to set key frames and how to propagate high-level features from key frames to non-key frames. Since low-level features in the network are relatively less costing to extract, and they also contain rich semantic information to reflect the characteristics of frames, the proposed method takes advantage of them to solve the two problems. Specifically, the key frame decision module (KDM) estimates potential prediction errors based on low-level features, and then uses them as a measure to select key frames. The local attention based module (LAM) calculates local corresponding relations on low-level features, and then uses them as weight coefficients to propagate high-level features from key frames to non-key frames. Moreover, the consistent constraint module (CC) is applied to further promote the semantic consistency.

Our proposed framework is diagramed in Fig. 2, in which a convolutional neural network is applied to extract features. The encoder is split into two parts: lower part E_l and higher part E_h. In the inference stage, given a video, to initialize the whole segmentation process, the first frame is set as a key frame, the index of which is k, and the segmentation result is obtained from the main model, in which both low-level feature F_l^k and high-level feature F_h^k are calculated. For the subsequent frame as current frame, the index of which is $k+n$, $(n > 0)$, the segmentation process is conducted adaptively. The low-level feature F_l^{k+n} is firstly extracted from E_l. Then the local weight coefficient $W^{k,k+n}$ is calculated via LAM with the input of both F_l^{k+n} and F_l^k. The next step is to determine whether to set the $k+n$ frame as a new key frame. Two low-level features F_l^k, F_l^{k+n} and corresponding local weight coefficient $W^{k,k+n}$ are input into KDM, to reach the judgement, depending on the feature propagating ability. If the output of KDM is 0, the $k+n$ frame is not selected as a key frame, the high-level feature F_h^{k+n} is propagated from the previous key frame, and also supervised by CC for promoting semantic consistency. In addition, to make model more robust against scene changes, same as in LLVS [15], both F_l^{k+n} and F_h^{k+n} are used to predict final results, where F_l^{k+n} is adapted by means of three convolution layers and then concatenated with F_h^{k+n} together. A simple decoder is applied to get the final segmentation result. If the output of KDM is 1, the $k+n$ frame is selected as a new key frame, and gets result from lower part encoder, higher part encoder and decoder section. The subsequent frames are compared and propagated with this new key frame.

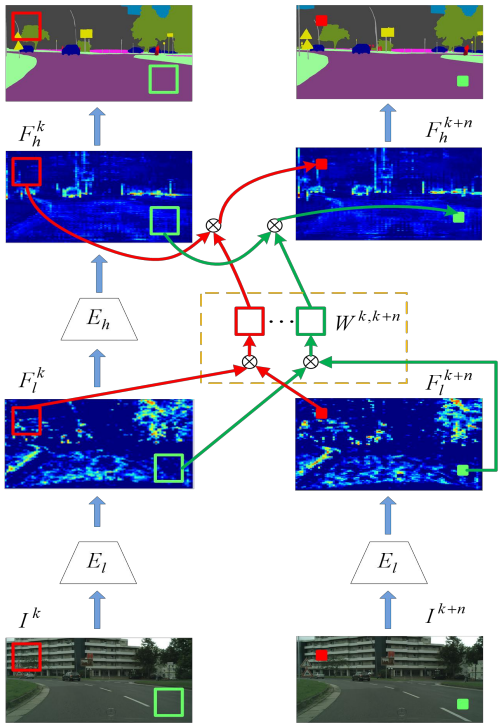


Fig. 4: The diagram of propagating features from key frame to non-key frame. Let I^k denote key frame, I^{k+n} denote non-key frame. The low-level features F_l^k, F_l^{k+n} are extracted firstly, then based on these two features, local similarity calculation is applied to obtain weight $W^{k,k+n}$. Using $W^{k,k+n}$, the high-level feature of non-key frame F_h^{k+n} is calculated via linear combinations of corresponding local neighbors on the high-level feature of key frame F_h^k. Finally, F_h^{k+n} is used to predict segmentation results.

As the cost of producing low-level feature and propagating high-level feature is much lower than producing high-level feature from scratch, our proposed method can decrease computation and inference time largely. Furthermore, the appropriate key frame selection is able to increase overall accuracy.

B. Local Attention Based Module

1) Motivation: Since the difference is minor between consecutive frames in a video, we can find the corresponding

relations in a local neighbouring area between key frame and non-key frame, where the similarity of features can be utilized as weight for propagation. To obtain weights, one intuitive method is to calculate local similarity based on high-level features directly, e.g. DMVOS [58]. However, this way defeats the purpose of reducing computation by omitting the generation of high-level features. In Fig. 3, we choose two adjacent frames and present their local correlation map between red dot position of image1 and corresponding local neighbouring area in yellow rectangle of image2 on low-level (Fig. 3(c)) and high-level (Fig. 3(d)) features separately. It could be seen that they have similar feature correlation in a local scope on different levels, for the reason that pixels belong to same category tend to have similar feature on both low-level and high-level aspect in the same network. Motivated by this observation, also considering low-level features are much less costly to obtain, this paper adopts low-level features to calculate the weights for propagating high-level features from key frames to non-key frames.

2) Feature Propagation: This paper utilizes local attention mechanism [59] to conduct feature propagation. The detailed process is illustrated in Fig. 4.

Assume the key frame is I^k and non-key frame is I^{k+n}. Their low-level features F_l^k, F_l^{k+n} are obtained from lower part of encoder E_l, the size is $N_2 \times H_2 \times W_2$, where H_2, W_2, N_2 represent height, width, number of channels respectively. Then the weight $W^{k,k+n}$ is obtained via local similarity calculation with the input of both F_l^k and F_l^{k+n}. The high-level feature of key frame F_h^k with size $N \times H \times W$ is calculated from higher part of encoder E_h. With F_h^k and $W^{k,k+n}$, the high-level feature of non-key frame can be obtained as

$$F_h^{k+n}(i,j) = \sum_{x=-r}^{r} \sum_{y=-r}^{r} W_{i,j}^{k,k+n}(x,y) F_h^k(i-x, j-y) \quad (1)$$

where i,j denote the coordinates of a pixel in the feature map, $i \in \{1,\ldots,W\}$, $j \in \{1,\ldots,H\}$, and R is length and width of the local area, $r = (R-1)/2$. Same weights are used in different channels. After getting F_h^{k+n}, a simple decoder is applied to obtain the segmentation result of non-key frame. The cross entropy loss with ground-truth label is utilized to supervise the training of model.

3) Local Similarity Calculation: In this section, we elaborate the way to calculate weight $W^{k,k+n}$ corresponding to the part in the orange dotted box in Fig. 4.

As illustrated in Fig. 5, two low-level features F_l^k, F_l^{k+n} get bilinear interpolation firstly to make their width and height the same as those of the high-level features, then reduce the number of channels to $1/4$ of the original number via 1×1 convolution. The size of new features F_2^k, F_2^{k+n} is $C \times H \times W$. For each position (i,j) of F_2^{k+n}, where $i \in \{1,\ldots,W\}$, $j \in \{1,\ldots,H\}$, with size $1 \times 1 \times C$, there is a local area in F_2^k that could be referred to, where the center is (i,j) and the length and width is R. The weight could be calculated as

$$W_{i,j}^{k,k+n}(x,y) = F_2^{k+n}(i,j) F_2^k(i-x, j-y) \quad (2)$$

where x, y represent the displacement relative to the central position (i,j), and $x,y \in \{-(R-1)/2,\ldots,(R-1)/2\}$. The

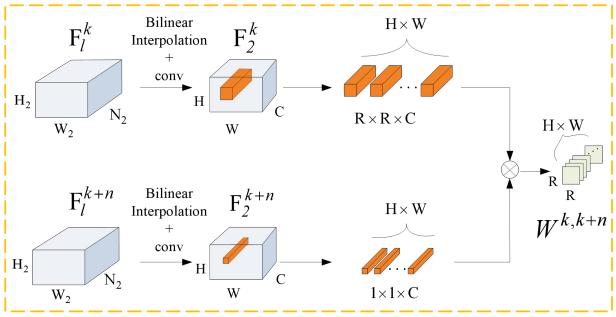


Fig. 5: The diagram of local similarity calculation.

computational complexity of local similarity is $O(CHW \times R^2)$, where $R << H, W$.

C. Consistent Constraint Module

1) Motivation: Previous video semantic segmentation methods only attend to building dependency between key frame and non-key frame, but ignore the correlation of pixels within non-key frame, which is crucial for promoting semantic consistency and increasing accuracy. For results obtained from only key frame feature propagation, as shown in the first and third row in Fig. 6, we visualize the map of correlation between one randomly selected position, marked by red dot in input image, and all other positions in the same image on high-level feature. Without the consideration of context information within non-key frame, the intra-class similarity and inter-class difference of features are indistinct, e.g. the distinction between the upper part of the car and the building is weak (shown in the first row of Fig. 6(b)); and the line between road and sidewalk is blurred (shown in the third row of Fig. 6(b)). Therefore, the phenomenon of inconsistent segmentation results inside the object is easy to occur.

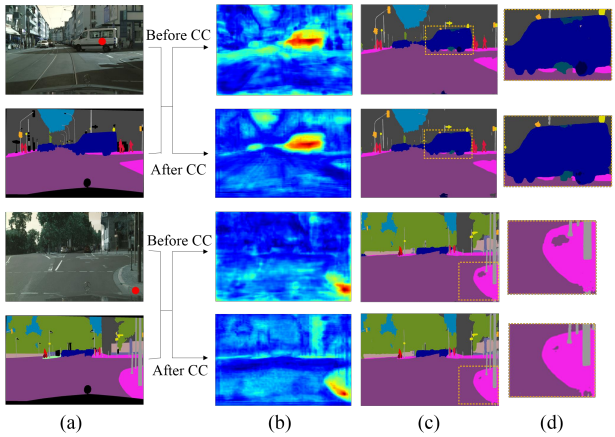


Fig. 6: The effectiveness of consistent constraint (CC): (a) two input images and their ground-truth segmentation: for each input image, a point is selected randomly and marked by red dot; (b) the visualization of correlation map of the given point; (c) segmentation results; (d) zoomed segmentation results.

Motivated by knowledge distillation methods [26]–[28], [60], [61], which are able to increase the accuracy of multi tasks via transferring the specific knowledge from each single task or increase the accuracy of small student network via transferring context information from large teacher model, a consistent constraint module is proposed to enrich the long-range dependency among pixels within non-key frame by distilling it from the feature obtained from key frame segmentation. It could promote semantic consistency of non-key frame, and do not add any computing burden at the same time. Moreover, with the aim of transferring context knowledge, instead of aligning feature maps directly [61], we use the pair-wise similarity among pixels as knowledge.

2) Consistent Constraint (CC): Assume the current frame I^{k+n} is judged as non-key frame, and the high-level feature via propagating from previous key frame is A. If I^{k+n} is processed as key frame segmentation, the corresponding high-level feature is B. Compared with feature A, feature B contains more context information due to the better segmentation performance it achieves. Therefore, feature B can be regarded as teacher, to transfer context knowledge to feature A.

Following [26]–[28], we use pair-wise correlation map to represent global context information. By assuming feature A is with size $N \times H \times W$, the correlation map G_A could be calculated as

$$g_{ij}^{A} = \frac{1}{H \times W} \cdot \frac{a_i a_j}{\|a_i\|_2 \|a_j\|_2} \tag{3}$$

where g_{ij}^{A} is the element of G_A, and $i, j \in \{1, \ldots, H \times W\}$. a_i, a_j are the feature vectors in A, which are located in positions i, j respectively, with size $N \times 1$. Cosine similarity is used to measure the similarity of two positions in feature. Let G_B represent the correlation map of feature B. The L_2 loss is adopted to formulate the consistent constraint loss as

$$L_{CC} = \|G_A - G_B\|_2^2 \tag{4}$$

It is noteworthy that the consistent constraint loss is only applied in the training stage, and it does not increase any computation in the inference stage.

3) Total Loss: The total loss comprises the cross entropy loss L_{ce} with ground-truth labels and the consistent constraint loss:

$$L = L_{ce} + \alpha L_{CC} \tag{5}$$

where α is a hyper-parameter to balance the effect of these two losses. We set α as 20 in this paper, to make the ranges of two loss values comparable.

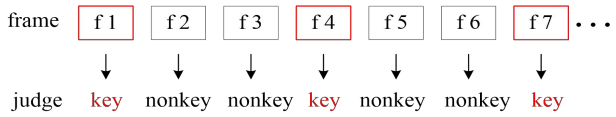
4) Visualization of Results with and without CC: To verify the effectiveness of CC module, we present the segmentation results and correlation map of high-level features with and without the CC. As shown in Fig. 6, after CC is applied, the intra-class similarity and inter-class difference of features are enhanced, e.g. features in the upper part of the car are more similar to the features in the center part, and distinguish significantly from the building behind (shown in Fig. 6(b) of the first image); the semantic consistency inside the sidewalk is enhanced, and the boundary between road and sidewalk gets clearer (shown in Fig. 6(b) of the second image). The long-range dependencies enriched by CC help model make more robust decisions, as shown in the segmentation results in Fig. 6(c-d), where the number of misclassified pixels inside objects has been reduced.

D. Key Frame Selection Strategy

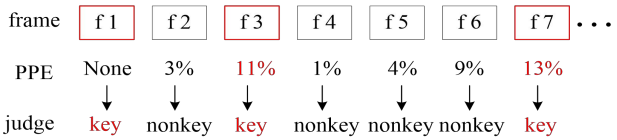
1) Motivation: Different from the common way to select key frames in a fixed time interval, LLVS [15] proposes a key frame selection strategy according to the situation of scene changing, which utilizes the ratio of different labels between previous key frame and current frame as a criterion to judge whether current frame should be regarded as key frame. Nevertheless, there exists such a situation: Although the ratio of different labels between key frame and current frame is large, the objects are moving regularly, and no new objects appear. It is more reasonable to regard the current frame as non-key frame in this situation, other than key frame as defined by LLVS [15].

Motivated by this observation, this paper proposes a novel adaptive key frame selection strategy, which takes the propagating ability of proposed dual correlation network as criterion. If the scene changes a lot, beyond the propagating ability of proposed method, the current frame is set as new key frame. Otherwise, if the scene changing only has a mild adverse effect on the performance of propagating process, the current frame is set as non-key frame.

2) Key Frame Decision Module (KDM): The common way of key frame selection is to select key frames in a fixed time interval, as shown in Fig. 7(a), it selects one key frame in every three frames. Different from the fixed time interval selecting strategy, this paper selects key frames adaptively, which defines potential prediction error (PPE), and uses it to select key frames. As shown in Fig. 7(b), to initialize the entire segmentation process, the first frame of a video is set as key frame, then the PPE between this key frame and the current frame is calculated. If PPE is smaller than a threshold, the current frame is set as non-key frame. Otherwise, the current frame is set as a new key frame, and subsequent frames calculate PPE with this new key frame.



(a) Fixed time interval key frame selection, the time period is 3



(b) Adaptive key frame selection, the threshold of PPE is 10%

Fig. 7: The processes of two key frame selection strategies.

The key to the propagating ability of proposed model is the local weights learned from low-level features. Since the upper

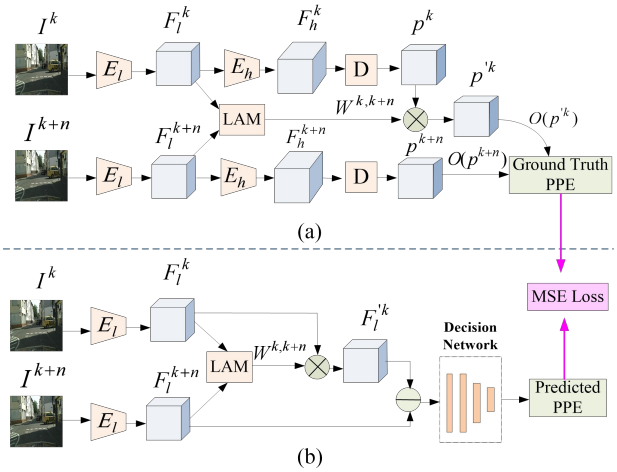


Fig. 8: The diagram of predicting PPE and the network training: (a) The process to calculate PPE as defined. Key frame I^k and current frame I^{k+n} are input into lower part of encoder E_l, higher part of encoder E_h, decoder D to get the probability outputs p^k and p^{k+n}. Based on p^k and the local weight $W^{k,k+n}$ (obtained from LAM), the propagated probability output $p^{'k}$ is calculated. Then the label outputs $O(p^{'k})$, $O(p^{k+n})$ are obtained to calculate the ratio of different labels, which is the ground truth PPE. (b) The network of predicting PPE. Low-level features F_l^k, F_l^{k+n} are extracted from E_l, and corresponding local weight $W^{k,k+n}$ is obtained from LAM. Based on F_l^k and $W^{k,k+n}$, the propagated low-level feature $F_l^{'k}$ is calculated. With the input of $F_l^{'k} - F_l^{k+n}$, the decision network will predict the value of PPE.

bound of propagating ability is when the result produced from local weight propagating is the same as the result produced from key frame segmentation, we define the difference of these two segmentation results as a criterion to measure the ability. The local weight represents the degree of local similarity between key frame and current frame, and we use it with the probability output to calculate difference.

Assume the probability outputs of key frame I^k and current frame I^{k+n} are p^k and p^{k+n} respectively via key frame segmentation, as shown in Fig. 8(a). The size of p^k and p^{k+n} is $M \times H \times W$, where M is the number of categories in labels. The local weight between two frames is $W^{k,k+n}$ with size $R^2 \times H \times W$. It is worth noting that the probability output is selected before interpolation and softmax operation in decoder. The probability output which is propagated from key frame via local weight can be written as

$$p^{'k}(i,j) = \sum_{x=-r}^{r} \sum_{y=-r}^{r} W_{i,j}^{k,k+n}(x,y) p^k(i-x, j-y) \quad (6)$$

where $i \in \{1, \ldots, W\}$, $j \in \{1, \ldots, H\}$, $r = (R-1)/2$.

The potential prediction error (PPE) is calculated as

$$PPE = \frac{D(O(p^{'k}), O(p^{k+n}))}{Q} \quad (7)$$

where $O(\cdot)$ represents the final label output of each probability output, $D(\cdot)$ is a function that counts the number of pixels which have different predicting labels, Q represents the total number of pixels in final label output. The larger the PPE, the bigger the difference between two segmentation results, and the weaker the propagating ability.

However, the probability output of current frame cannot be obtained in inference stage, a regression model is needed to be built for predicting PPE. To obtain PPE without calculating probability output, similar to our LAM, we utilize low-level features to predict it. In detail, as shown in Fig. 8(b), given key frame I^k and current frame I^{k+n}, their low-level features F_l^k, F_l^{k+n} and corresponding local weight $W^{k,k+n}$ are calculated. The low-level feature of current frame via propagating from key frame is

$$F_l^{'k}(i,j) = \sum_{x=-r}^{r} \sum_{y=-r}^{r} W_{i,j}^{k,k+n}(x,y) F_l^k(i-x, j-y) \quad (8)$$

where $i \in \{1, \ldots, W\}$, $j \in \{1, \ldots, H\}$, $r = (R-1)/2$.

Considering the positive correlation between $F_l^{'k} - F_l^{k+n}$ and PPE, we build a simple decision network to predict PPE with the input of $F_l^{'k} - F_l^{k+n}$. It consists of one convolutional layer, one average pooling layer and two full-connected layers. MSE loss is adopted to supervise the training of decision network with the ground truth PPE, which is defined in Eq.(7). Note that the ground truth PPE is only used for training, and it is not accessible in inference stage. The low-level features F_l^k, F_l^{k+n} need to get bilinear interpolation at first to make their width and height the same as those of the high-level feature, then go on propagating by using Eq.(8).

IV. Experiments

In this section, we start by an introduction to video datasets and implementation details, and then perform ablation studies to validate each module of the proposed method. Finally, we present segmentation performance on two public datasets, and compare our method with recent approaches.

A. Datasets

1) Cityscapes: The Cityscapes dataset is an urban street scene dataset, which contains 2975, 500, and 1525 snippets for training, validation and testing, respectively. The dataset is sparsely annotated. Each snippet has 30 frames of images, and only the 20th frame has dense pixel annotations, with 19 classes for evaluation.

2) Camvid: The Camvid dataset is a driving scene dataset. It contains three videos, with scenes at daytime and dusk. It annotates 701 images in detail, and they are divided into 367 images for training, 100 images for validation, and 233 images for testing. The number of semantic classes is 11.

B. Implementation Details

To validate that our proposed method can perform well in both cumbersome model and lightweight model, this paper chooses two public models as basic models: PSPNet101 [7] and Bisenet18 [44].

1) Network Structure: For PSPNet101, conv4_3 is choosed to be a split point between low-level feature and high-level feature. For Bisenet18, conv3_1 is choosed as the split point. In general, the higher layer the split point chooses, the better accuracy the segmentation results achieve, but then the more time is needed to generate low-level feature and perform propagating. To balance the accuracy and efficiency, we choose suitable split points for each model after several experiments.

For LAM, if the width or height of low-level feature is different from that of the high-level feature, the low-level feature gets bilinear interpolation at first to make its width and height the same as those of the high-level feature. In the local similarity calculation, to reduce computation, two low-level features of key frame and non-key frame reduce the number of channels to $1/4$ of the original number via a convolution layer with 1×1 kernels. Then the local weight is calculated by using Eq.(2). In the feature propagation, the high-level feature of key frame reduces the number of channels to $1/8$ of the original number firstly via a convolutional layer with 3×3 kernels. Then the feature is propagated by using Eq.(1). In addition, to make the model more robust against scene changes, the low-level feature of non-key frame is also used for final prediction. It adjusts the number of channels to the number of propagated high-level feature via a convolutional layer with 3×3 kernels, and then is fed to two other convolution layers with the same number of channels and 3×3 kernels. Finally, the adapted low-level feature is concatenated with the propagated high-level feature to predict segmentation results. The Cross Entropy Loss and the proposed Consistent Constraint Loss are adopted to train this module.

The KDM takes two low-level features of key frame and non-key frame and the corresponding local weight as input, and the output is potential prediction error. Same as in LAM, the two low-level features need to get bilinear interpolation firstly to align their width and height with those of the high-level feature, which subsequently reduce the number of channels to $1/4$ of the original number via a convolution layer with 1×1 kernels. To further reduce computation, we adopt the encoder-decoder mode for low-level feature propagation. In detail, following the above operation, another convolutional layer with 1×1 kernels is applied on the low-level feature of key frame to reduce the number of channels from $1/4$ to $1/16$ of the original number. Then this feature is propagated by using Eq.(8) with the corresponding local weight, and then recovered the number of channels to $1/4$ of the original number via a convolutional layer with 1×1 kernels. The propagated low-level feature of key frame is subtracted from the low-level feature of non-key frame, and then put into a decision network to get the value of potential prediction error, which contains one convolutional layer with 3×3 kernels, one average pooling layer and two full-connected layers. The MSE Loss is adopted to train this module.

2) Training: The basic networks PSPNet101 and Bisenet18 are firstly trained with image data and ground truth labels. As the dataset is sparsely annotated, we choose a pair of images each time to train LAM and KDM modules. The second image of each pair is with ground truth labels, while the first image is a preceding frame that is randomly selected from [1,10] interval. The first image of each pair is regarded as key frame, and the second image is regarded as current frame. The next step is to train the LAM module, in which the segmentation results of current frame and the corresponding ground truth labels are utilized to supervise the training of model. In the training stage, the parameters of feature extraction are kept unchanged, and we fine-tune the parameters of LAM. The last step is to train KDM with the ground truth PPE, and in the training stage, the parameters of feature extraction and LAM are kept unchanged, and we only fine-tune the parameters of KDM. The training data are augmented via randomly scaling, rotating, flipping, and cropped into size 713×713 for Cityscapes and 360×480 for Camvid. The stochastic gradient descent (SGD) is adopted in the training, with initial learning rate 0.1, 0.01 for LAM, KDM respectively, and poly learning rate strategy with power 0.9.

3) Evaluation Metrics: For evaluating accuracy, this paper adopts mean Intersection Over Union (mIOU) as criterion. IOU is the ratio of intersection to union between segmentation results and ground truth labels for each category, while mIOU is the average of IOU over all object categories. To evaluate efficiency, this paper measures the inference time of processing a frame in GTX 1080Ti. The inference time is calculated in a single scale with input size 713×713 for Cityscapes, and 360×480 for Camvid. In this paper, Time(ms) and FPS(f/s) both represent the inference time. Time(ms) is the average time needed to process one frame, and FPS(f/s) is the number of frames processed per second.

The mIOU and inference time of video semantic segmentation are calculated as the average over one period of frames. Let I^t denote the frame with ground truth labels in the video, (I^{t-m}, I^t) represent all possible groups in which I^{t-m} is a key frame and I^t is a non-key frame. For a fixed interval key frame selection strategy with period T_1, $m \in [0, T_1 - 1]$. For our proposed KDM selection strategy, $m \in [0, T_2 - 1]$, where the PPE between I^{t-m} and I^t is lower than the specified threshold, and the PPE between I^{t-T_2} and I^t is higher than the threshold. The mIOU and inference time are calculated as the average over (I^{t-m}, I^t).

C. Ablation Studies

1) Search Region R: Assume R is the length and width of searching region. This section evaluates the impact of searching region on segmentation performance, and selects suitable one for subsequent experiments, as shown in Table I.

TABLE I: Performance of different searching regions with PSPNet101 as the basic network on the Cityscapes val dataset. The best is in bold.

R	5	7	9	11	13
mIOU(%)	73.29	73.35	73.45	**73.64**	73.54
Time(ms)	131	137	139	154	187

In this section, the interval of key frame is fixed and set as 5. From Table I, we can make the following observations. First, the inference time is increased when enlarging the searching region, for the reason that the computing complexity of processing non-key frame is $O(CHW \times R^2)$. Second,

we observe the accuracy is also increased as R is increased, since more useful information in key frame is aggregated and transferred to non-key frame. Third, the accuracy achieves a peak for $R = 11$. We conjecture that too large searching region may introduce noise and cause feature mismatching, which have an adverse effect on the results. Therefore, a reasonable size is needed to be selected. Although the accuracy for $R = 11$ is the highest, the inference speed is slow. To get a trade-off between accuracy and efficiency, we choose $R = 9$ in the following experiments.

2) The Construction of KDM: This section shows several factors that affect the performance of key frame selection, as shown in Table II. With different criteria and inputs, the same decision network structure is used to predict the value, which contains one convolutional layer, one pooling layer and two full-connected layers. Two criteria are adopted to select key frame, one is label difference [15], which takes the ratio of different labels between key frame and current frame as a criterion, and the other one is our PPE. First, we compare the results with these two criteria. The mIOU of the ground truth PPE criterion is higher than the mIOU of ground truth label difference criterion. It verifies the superiority of our proposed key frame selection strategy, since our strategy considers the processing ability of modules against scene changes, which is ignored in label difference criterion. The mIOU of predicted PPE criterion is also higher than the mIOU of predicted label difference criterion.

Second, different inputs fed to decision network affect the error of predicting value, then affect the segmentation results. Same as the input of predicted label difference criterion [15], $F_l^k - F_l^{k+n}$ is fed to decision network to predict PPE. However, the error between predicted PPE and ground truth PPE is higher than the error in label difference predicting. To increase the positive correlation between input and PPE, we use $W^{k,k+n}F_l^k - F_l^{k+n}$ as input since PPE is calculated from the difference between $W^{k,k+n}p^k$ and p^{k+n}. Compared with the result using $F_l^k - F_l^{k+n}$ to predict PPE, the error is lower when using $W^{k,k+n}F_l^k - F_l^{k+n}$ as input, and the mIOU is increased. In addition, extra 3 ms is needed for calculating $W^{k,k+n}F_l^k$ and producing $W^{k,k+n}$ for key frames since $W^{k,k+n}$ is not accessible in key frame segmentation process.

Third, we compare the results from two different ground truth PPE calculations. One is our proposed way, which is defined in Eq.(6) and Eq.(7). The other is that the ground truth PPE is calculated as the difference between current frame segmentation results and the results generated by the proposed model, denoted as ME (Model Error). The mIOU of the ground truth ME criterion is higher than the mIOU of ground truth PPE criterion, since ME can better reflect the modeling gap between the proposed model and the key frame segmentation model. However, with the input of $W^{k,k+n}F_l^k - F_l^{k+n}$, the error between predicted ME and ground truth ME is higher than the error between predicted PPE and ground truth PPE, leading to much lower mIOU of predicted ME criterion. We conjecture that, since the results generated by the proposed model are not only influenced by the weighting propagation in LAM, but also influenced by other designs, e.g. the adaption of low-level feature and CC, the input of $W^{k,k+n}F_l^k - F_l^{k+n}$ cannot well predict ME. Our PPE is calculated using the difference between $W^{k,k+n}p^k$ and p^{k+n}, and it has strong positive correlation with the input, so the prediction error is lower than that of ME, and the corresponding mIOU is higher.

TABLE II: Performance of key frame selection strategies with different criteria and inputs on the basis of proposed feature propagation model and on the Cityscapes val dataset. 'Criterion' means the criterion to select key frame; 'Input' means the input fed to decision network; 'Error' means the mean absolute error between predicted value and ground truth value; 'mIOU' means the segmentation accuracy; 'Time' means the inference time.

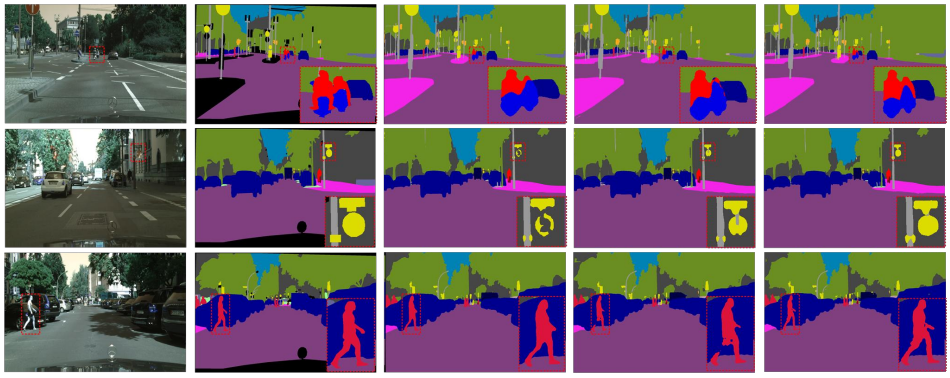
Criterion	Input	Error(%)↓	mIOU(%)↑	Time(ms)↓
Ground truth label difference [15]	-	-	75.92	-
Predicted label difference [15]	$F_l^k - F_l^{k+n}$	2.04	75.12	155
Ground truth PPE (ours)	-	-	76.3	-
Predicted PPE (ours)	$F_l^k - F_l^{k+n}$	2.55	75.26	155
Predicted PPE (ours)	$W^{k,k+n}F_l^k - F_l^{k+n}$	2.12	75.57	158
Ground truth ME	-	-	76.75	-
Predicted ME	$W^{k,k+n}F_l^k - F_l^{k+n}$	3.19	74.63	158

3) The Effectiveness of LAM and CC: To demonstrate the effectiveness of proposed method, this section enables and disables different modules. The results are summarized in Table III.

TABLE III: Ablation studies with different modules in our method on the Cityscapes val dataset. PSPNet101 is selected as the basic model. 'fixed schedule' means that key frame is selected every 5 frames.

Method	mIOU(%)	Time(ms)
Basic Model	79.49	321
LAM + fixed schedule	73.45	139
LAM + CC + fixed schedule	74.06	139
LAM + KDM	75.06	158
LAM + CC + KDM	75.57	158

The basic model achieves high accuracy, but with long inference time. LAM speeds up the pipeline, reducing inference time from 321 ms to 139 ms per frame, while having a 6% drop in accuracy. Using CC, the accuracy is increased by 0.6% while keeping inference speed unchanged. In addition to the quantitative analysis, we also visualize some segmentation results in Fig. 9. LAM is able to successfully achieve feature propagating for non-key frame segmentation. CC could promote the consistency of segmentation results within objects, e.g. traffic sign, and provide details for small objects, e.g. people in the distance. Fig. 10 shows the segmentation results of one video. The segmentation performance is good even if only using LAM when the interval between key frame and non-key frame is no more than 5. For the $k + 6$ and $k + 7$ frames, which are far from key frame, there are some misclassified pixels inside the pedestrian (marked by the white



(a) Image (b) Ground Truth (c) Basic Model (d) LAM (e) LAM + CC

Fig. 9: Visualized results of LAM and CC. The results in the red dotted box are enlarged in the lower right corner of the picture.

dotted box) when only using LAM, due to the local correlation in LAM. However, CC can correct these misclassified pixels since it considers long-range dependency among pixels. The segmentation results of LAM+CC are as good as those of basic model on $k+6$ and $k+7$ frames.

4) The Effectiveness of KDM: As shown in Table III, the accuracy is increased by 1.5% after applying KDM, while an extra 19 ms inference time is needed to check whether the frame is regarded as key frame.

We also compare several key frame selection strategies on the basis of proposed feature propagating model. In detail, it consists of three strategies: fixed interval selection, key frame selection strategy in LLVS [15], and our proposed KDM. The results are shown in Fig. 11. On the one hand, under the same key frame interval, the proposed strategy achieves higher accuracy than the other two strategies. On the other hand, under the same accuracy, less frequency of key frames is needed in our proposed strategy. The superiority of our key frame selection strategy over LLVS [15] can be ascribed to the fact that our strategy not only considers the influence of scene changing on key frame selection, but also considers the processing capability of the model for scene changing.

D. Segmentation Results on Cityscapes and Camvid

In this section, we present segmentation performance of the proposed method on two public video datasets, and compare it with recent video segmentation approaches.

1) Cityscapes: For achieving high accuracy, we choose PSPNet101 [7] as the basic model, which is image-based method. The comparisons with other high accuracy video semantic segmentation methods are listed in Table IV. The proposed method outperforms DAVSS, LLVS, DVRL, Accel-18, DFF, GRFP and Clockwork, with both higher segmentation accuracy and less inference time. Compared with DVS and THA, although our method has a little lower inference speed, it improves 5% and 3.7% mIOU of accuracy respectively. Note that although LMA and TMANet achieve the higher accuracy, their inference speeds are far slower than the basic model and other video-based methods. Benefiting from local attention mechanism, our method can obtain temporal correlation information among frames in an efficient way, and realize a good trade-off between accuracy and efficiency.

For realizing high efficiency, we choose Bisenet18 [44] as the basic model. As shown in Table V, the proposed method with fixed interval strategy gets the higher mIOU, and also has faster inference speed than EVS and DVS methods. Although THA has the fastest speed, its accuracy is the lowest, since it only considers feature propagation on one-to-one same position among frames, and this is unreasonable since the objects keep on moving. Our method considers feature propagation in a local scope, which overcomes the shortcoming of THA and improves accuracy greatly. Note that the computational complexity of THA is $O(CHW)$, and the computational complexity of our method is $O(CHW \times R^2)$. As $R^2 << HW$, the inference speed of proposed method is also fast but with much higher accuracy than THA shown in Table V.

TABLE IV: Segmentation performance of video-based methods with high accuracy on the validation sets of Cityscapes.

Method	Backbone	mIOU(%)↑	Time(ms)↓
PSPNet101 [7] (Basic Model)	ResNet101 [33]	79.49	321
TMANet [53]	ResNet50	80.3	500
LMA [54]	PSP-SS-SC	78.48	758
DAVSS [14]	DeepLabv3+	75.42	170
LLVS [15]	ResNet101	75.1	162
DVRL [23]	ResNet101	72.9	182
Accel-18 [16]	ResNet18	72.1	440
DVS [22]	ResNet101	70.2	87
DFF [17]	Resnet101	69.2	156
GRFP [18]	Resnet101	69.4	312
Clockwork [19]	FCN	67.7	141
THA [20]	ResNet101	71.87	128
Our method (fixed interval schedule)	ResNet101	74.06	139
Our method (KDM)	ResNet101	75.57	158

Fig. 12 visualizes some segmentation results obtained from our proposed method and two recent video semantic segmentation methods [14], [20]. The distance is set as 4 between key frame and the frame to be visualized for all methods.

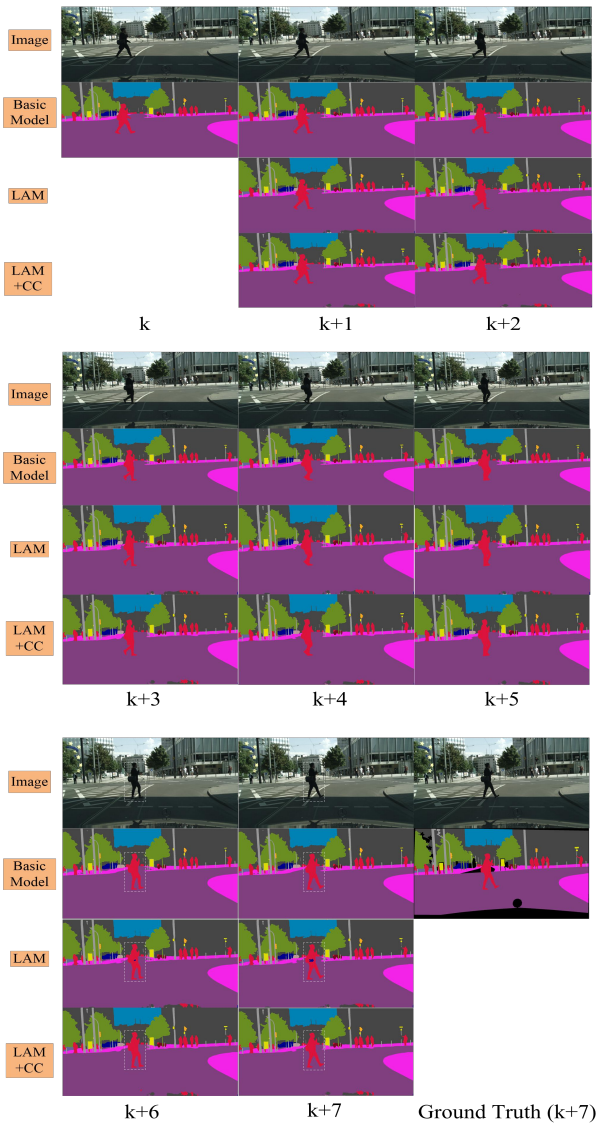


Fig. 10: The segmentation results of a video produced by Basic Model, LAM and CC, where k denotes key frame, and $k+n$ denotes non-key frame. In addition, we also give the ground truth label of $k+7$ frame for comparison.

TABLE V: Segmentation performance of video-based methods with high inference speed on the validation sets of Cityscapes.

Method	Backbone	mIOU(%)↑	FPS (f/s)↑
Bisenet18 [44] (Basic Model)	Bisenet18	73.8	50
EVS [21]	ICNet [62]	66.2	77
DVS [22]	ResNet18	62.6	30
THA [20]	LERNet	60.6	125
Our method (fixed interval schedule)	Bisenet18	66.8	83
Our method (KDM)	Bisenet18	67.5	75

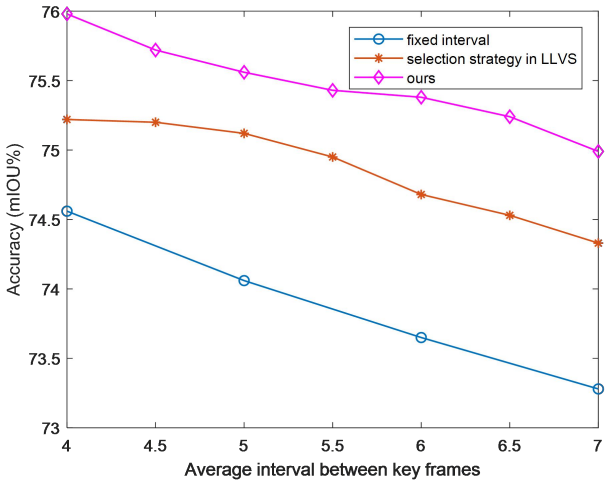


Fig. 11: The performance of different key frame selection strategies on the Cityscapes val dataset.

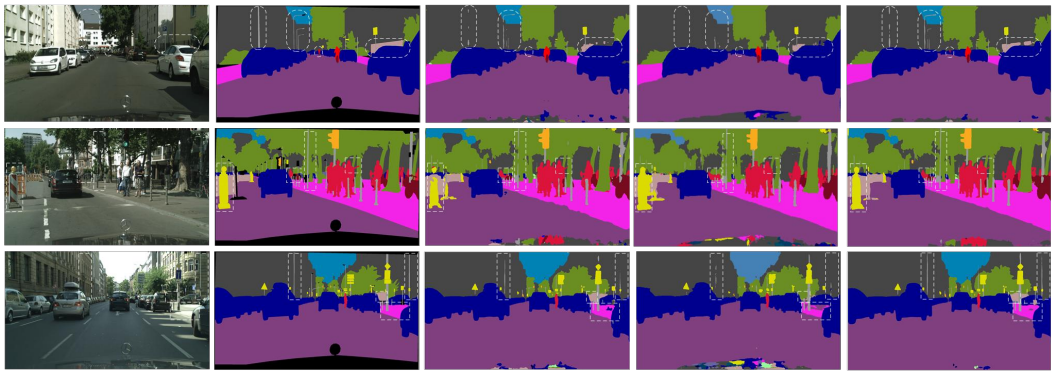
DAVSS [14] fails to segment small size objects or present the details of objects, e.g. thin telegraph pole, objects in the distance, and the arms of people. THA [20] fails to keep the semantic consistency within objects, e.g. incomplete segmentation of fence and traffic sign. The results from our proposed method are much closer to the ground truth, as shown in the white dotted box, better on both segmenting small objects and maintaining semantic consistency within objects.

2) Camvid: PSPNet50 [7] achieves a good trade-off of accuracy and efficiency on Camvid, so we choose it as the basic model on this dataset. The performances are presented in Table VI. Compared with the basic model, the proposed method with fixed interval strategy reduces nearly half of inference time while only decreasing 1.26% mIOU. Compared with other video-based methods except THA, our proposed method achieves the highest accuracy with the minimum inference time. The proposed method with fixed interval schedule outperforms THA in accuracy while only increasing 2 ms inference time.

TABLE VI: Segmentation performance on the test sets of Camvid.

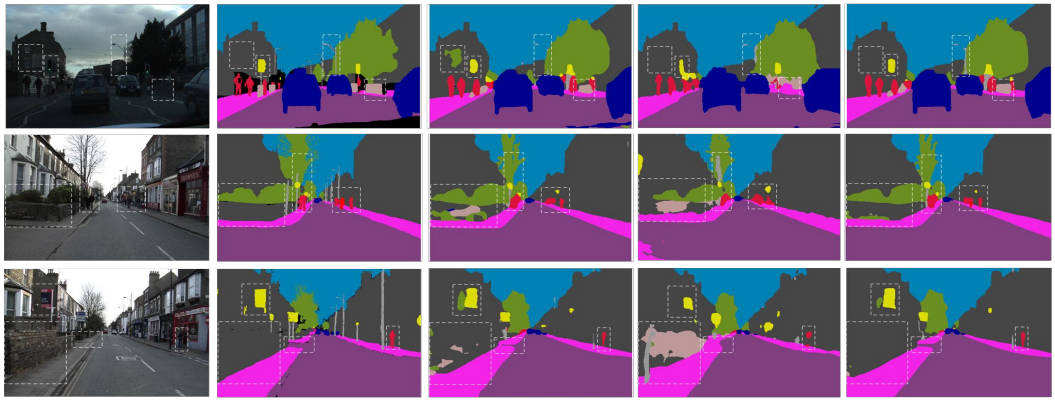
Method	Backbone	mIOU(%)	Time(ms)
PSPNet50 [7] (Basic Model)	ResNet50 [33]	73.79	77
DFF [17]	ResNet101	66.0	62
GRFP [18]	ResNet101	66.1	230
Accel-18 [16]	ResNet18	66.7	132
DAVSS [14]	DeepLabv3+	70.2	46
Netwarp [45]	Dilation	67.1	363
THA [20]	ResNet50	71.76	39
Our method (fixed interval schedule)	ResNet50	72.53	41
Our method (KDM)	ResNet50	73.03	50

The segmentation results of our method and two other recent methods [14], [20] on Camvid are shown in Fig. 13. The distance is set as 4 between key frame and the frame to be visualized for all methods. The proposed method performs better on reducing the number of misclassified pixels inside



(a) Image (b) Ground Truth (c) THA (d) DAVSS (e) Our method

Fig. 12: Visualized results on the Cityscapes dataset.



(a) Image (b) Ground Truth (c) THA (d) DAVSS (e) Our method

Fig. 13: Visualized results on the Camvid dataset.

the object and keeping semantic consistency. In addition, it also has better performance on small objects segmentation, e.g. people in the distance and traffic sign.

V. Conclusion

This paper proposes a novel approach to efficient video semantic segmentation, by leveraging two complementary modules for feature transformation, and an adaptive key frame selection strategy. Through a local attention based module, the high-level feature of key frame can be propagated to non-key frame in a local and high efficient way; through a consistent constraint module that considers long-range context information within non-key frame, semantic consistency of segmentation results can be promoted without increasing any inference burden; through a reasonable scheduling of key frames, the overall segmentation efficiency and performance can be both improved. Extensive experiments on two public video datasets have demonstrated that the proposed method achieves high efficiency in segmentation while maintaining satisfactory accuracy.

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[2] Y. Zhao, Z. Zhong, Z. Luo, G. H. Lee, and N. Sebe, "Source-free open compound domain adaptation in semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[3] L. Ding, H. Tang, and L. Bruzzone, "Lanet: Local attention embedding to improve the semantic segmentation of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 426–435, 2021.

[4] K. Wang, Y. Lin, L. Wang, L. Han, M. Hua, X. Wang, S. Lian, and B. Huang, "A unified framework for mutual improvement of slam and semantic segmentation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5224–5230.

[5] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation." *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[6] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE transactions on*

pattern analysis and machine intelligence, vol. 40, no. 4, pp. 834–848, 2017.

[9] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[10] X. Li, L. Zhang, G. Cheng, K. Yang, Y. Tong, X. Zhu, and T. Xiang, "Global aggregation then local distribution for scene parsing," in *IEEE Transactions on Image Processing*, vol. 30, 2021, pp. 6829–6842.

[11] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3684–3692.

[12] J. Ji, R. Shi, S. Li, P. Chen, and Q. Miao, "Encoder-decoder with cascaded crfs for semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1926–1938, 2020.

[13] K. Lin, L. Wang, K. Luo, Y. Chen, Z. Liu, and M.-T. Sun, "Cross-domain complementary learning using pose for multi-person part segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1066–1078, 2020.

[14] J. Zhuang, Z. Wang, and B. Wang, "Video semantic segmentation with distortion-aware feature correction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3128–3139, 2020.

[15] Y. Li, J. Shi, and D. Lin, "Low-latency video semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5997–6005.

[16] S. Jain, X. Wang, and J. 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*, 2019, pp. 8866–8875.

[17] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2349–2358.

[18] D. Nilsson and C. Sminchisescu, "Semantic video segmentation by gated recurrent flow propagation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6819–6828.

[19] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell, "Clockwork convnets for video semantic segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 852–868.

[20] J. Wu, Z. Wen, S. Zhao, and K. Huang, "Video semantic segmentation via feature propagation with holistic attention," *Pattern Recognition*, vol. 104, p. 107268, 2020.

[21] M. Paul, C. Mayer, L. V. Gool, and R. Timofte, "Efficient video semantic segmentation with labels propagation and refinement," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2873–2882.

[22] Y.-S. Xu, T.-J. Fu, H.-K. Yang, and C.-Y. Lee, "Dynamic video segmentation network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6556–6565.

[23] Y. Wang, M. Dong, J. Shen, Y. Wu, S. Cheng, and M. Pantic, "Dynamic face video segmentation via reinforcement learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6959–6969.

[24] S. Liu, K. Luo, A. Luo, C. Wang, F. Meng, and B. Zeng, "Asflow: Unsupervised optical flow learning with adaptive pyramid sampling," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[25] R. Zhao, R. Xiong, Z. Ding, X. Fan, J. Zhang, and T. Huang, "Mrdflow: Unsupervised optical flow estimation network with multi-scale recurrent decoder," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[26] S. An, Q. Liao, Z. Lu, and J.-H. Xue, "Efficient semantic segmentation via self-attention and self-distillation," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[27] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 578–587.

[28] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2604–2613.

[29] S. Park and Y. S. Heo, "Knowledge distillation for semantic segmentation using channel and spatial correlations and adaptive cross entropy," *Sensors*, vol. 20, no. 16, p. 4616, 2020.

[30] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[31] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[32] F. Lin, Z. Liang, S. Wu, J. He, K. Chen, and S. Tian, "Structtoken: Rethinking semantic segmentation with structural prior," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[36] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville, "ReSeg: A recurrent neural network-based model for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 41–48.

[37] B. Shuai, Z. Zuo, B. Wang, and G. Wang, "Dag-recurrent neural networks for scene labeling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3620–3629.

[38] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

[39] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 603–612.

[40] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[41] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[42] X. Weng, Y. Yan, S. Chen, J.-H. Xue, and H. Wang, "Stage-aware feature alignment network for real-time semantic segmentation of street scenes," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[43] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, and L. J. Latecki, "LEDNet: A lightweight encoder-decoder network for real-time semantic segmentation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1860–1864.

[44] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 325–341.

[45] R. Gadde, V. Jampani, and P. V. Gehler, "Semantic video cnns through representation warping," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 4463–4472.

[46] M. Ding, Z. Wang, B. Zhou, J. Shi, Z. Lu, and P. Luo, "Every frame counts: Joint learning of video segmentation and optical flow," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10 713–10 720.

[47] J. Fan, B. Liu, K. Zhang, and Q. Liu, "Semi-supervised video object segmentation via learning object-aware global-local correspondence," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8153–8164, 2021.

[48] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, F. Huang, and R. Klette, "Stfcn: spatio-temporal fully convolutional neural network for semantic segmentation of street scenes," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 493–509.

[49] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, and S. J. Maybank, "Asymmetric 3d convolutional neural networks for action recognition," *Pattern recognition*, vol. 85, pp. 1–12, 2019.

[50] J. Ji, S. Buch, A. Soto, and J. C. Niebles, "End-to-end joint semantic segmentation of actors and actions in video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 702–717.

[51] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie *et al.*, "Video scene parsing with predictive feature learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5580–5588.

[52] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8856–8865.
[53] H. Wang, W. Wang, and J. Liu, "Temporal memory attention for video semantic segmentation," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 2254–2258.
[54] M. Paul, M. Danelljan, L. Van Gool, and R. Timofte, "Local memory attention for fast video semantic segmentation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1102–1109.
[55] P. Huang, J. Han, N. Liu, J. Ren, and D. Zhang, "Scribble-supervised video object segmentation," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 339–353, 2021.
[56] R. Cong, J. Lei, H. Fu, F. Porikli, Q. Huang, and C. Hou, "Video saliency detection via sparsity-based reconstruction and propagation," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4819–4831, 2019.
[57] Y. Liu, C. Shen, C. Yu, and J. Wang, "Efficient semantic video segmentation with per-frame inference," in *European Conference on Computer Vision*. Springer, 2020, pp. 352–368.
[58] P. Wen, R. Yang, Q. Xu, C. Qian, Q. Huang, R. Cong, and J. Si, "Dmvos: Discriminative matching for real-time video object segmentation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2048–2056.
[59] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.
[60] D. Zhang, J. Han, L. Yang, and D. Xu, "Spftn: A joint learning framework for localizing and segmenting objects in weakly labeled videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 2, pp. 475–489, 2018.
[61] F. Lin, H. Xie, C. Liu, and Y. Zhang, "Bilateral temporal re-aggregation for weakly-supervised video object segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4498–4512, 2021.
[62] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 405–420.

Shumin An received her M.Eng. degree in electronic and communication engineering from the University of Science and Technology of China in 2018. She is currently pursuing the Ph.D. degree in the Department of Electronic Engineering, Tsinghua University. Her research interests include image processing, pattern recognition and artificial intelligence.

Qingmin Liao received the Ph.D. degree in signal processing and telecommunications from the University of Rennes 1 in 1994. He is a Professor in the Department of Electronic Engineering and the Shenzhen International Graduate School, Tsinghua University. His research interests include image/video processing, transmission, analysis, biometrics and their applications.

Zongqing Lu received the Ph.D. degree in signal processing from Xidian University in 2007. He is an Assistant Professor in the Department of Electronic Engineering, Tsinghua University. His research interests include image processing and machine learning.

Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is a Professor in the Department of Statistical Science, University College London. His research interests include statistical classification, high-dimensional data analysis, pattern recognition and image processing. He is an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Cybernetics, and IEEE Transactions on Neural Networks and Learning Systems.