Hybrid Instance-Aware Temporal Fusion for Online Video Instance Segmentation

Xiang Li, [1,2*] **Jinglu Wang,** [2] **Xiao Li,**[2] **Yan Lu** [2]

[1] Department of Electrical and Computer Engineering, Carnegie Mellon University
[2] Microsoft Research Asia
xl6@andrew.cmu.edu, {jinglwa, xili11, yanlu}@microsoft.com

Abstract

Recently, transformer-based image segmentation methods have achieved notable success against previous solutions. While for video domains, how to effectively model temporal context with the attention of object instances across frames remains an open problem. In this paper, we propose an online video instance segmentation framework with a novel instance-aware temporal fusion method. We first leverage the representation, i.e., a latent code in the global context (instance code) and CNN feature maps to represent instance- and pixel-level features. Based on this representation, we introduce a cropping-free temporal fusion approach to model the temporal consistency between video frames. Specifically, we encode global instance-specific information in the instance code and build up inter-frame contextual fusion with hybrid attentions between the instance codes and CNN feature maps. Inter-frame consistency between the instance codes is further enforced with order constraints. By leveraging the learned hybrid temporal consistency, we are able to directly retrieve and maintain instance identities across frames, eliminating the complicated frame-wise instance matching in prior methods. Extensive experiments have been conducted on popular VIS datasets, i.e. Youtube-VIS-19/21. Our model achieves the best performance among all online VIS methods. Notably, our model also eclipses all offline methods when using the ResNet-50 backbone.

Introduction

Video instance segmentation (VIS), aiming at simultaneously classifying, segmenting, and tracking object instances, attracts increasing attention recently due to boosting interest in video analysis. VIS methods are categorized into offline and online methods according to two kinds of inputs, i.e., clip- and frame-wise inputs respectively. Offline methods obtain impressive accuracy thanks to modeling spatial-temporal correlation throughout the whole clip (Bertasius and Torresani 2020; Athar et al. 2020; Wang et al. 2021b), but they inevitably show limitations on real streaming applications. Online methods are more practical for streaming applications, but their performance of existing methods are far from that of offline methods because of their imperfection in modeling frame-to-frame (F2F) communications. We focus on F2F communications for online VIS task.

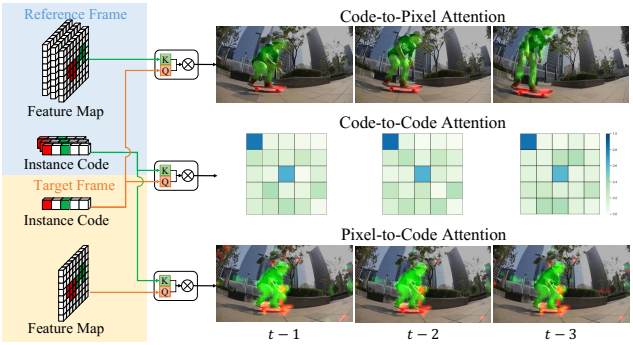

Figure 1: Visualization of attention maps in inter-frame attention layers. The red and green colors in the instance code and feature map indicate different instances. "Q" and "K" represent the Query and Key operations respectively. All queries come from the target frame and keys from reference frames. We can find the attention responding on references $\{t-1, t-2, t-3\}$ are consistent.

Some online methods (Yang et al. 2021; Fu et al. 2020; Li et al. 2021b) model F2F communication only at pixel level. Modeling higher level semantics including instance-aware information is limited and expensive. Other methods (Li et al. 2021a; Gong et al. 2021; Fu et al. 2020) model F2F communication at instance level by first cropping out RoI features with detected boxes to obtain instance-level proxies and then associating or fusing such cropped features. The F2F communication of box-based method heavily relies on the detection accuracy. Although sophisticated feature alignment can be employed, the F2F communication is still incomplete and biased because the cropped instances are isolated from the global context. All prior methods either model the F2F communication at solely pixel or instance level, while no joint communications at hybrid levels are discussed.

In this work, we focus on improving online VIS by introducing novel hybrid instance-aware F2F communications. Considering the limitation of box-based methods in video frame communication, we build up our method based on the state-of-the-art box-free image segmentation framework,

*This work was done when Xiang Li was an intern at MSRA.

i.e., MaX-DeepLab (Wang et al. 2021a). Inspired by Max-DeepLab, we employ a global latent code as well as CNN feature maps for jointly representing instance-aware features. We term the global code *instance code* as it could capture instance-aware high-level clues in the global context in our VIS task. Benefiting from this hybrid representation, we enforce the temporal consistency in both instance code and feature map by two designs. First, we model the F2F communications at hybrid level by employing cross-attentions between both instance code and feature maps. Second, we enhance the cross-frame consistency of instance features by utilizing and further consolidating the slot consistency of instance code during training with a novel consistency constraint. In inference stage, the instance identity can be direct associated with slot indices, thus greatly reduce the cross-frame instance matching cost.

Our method is designed with two key insights. First, adopting such a box-free representation to VIS task enables us to remove the reliance on "detection-and-cropping" approach for instance-level F2F communication; instead, we are able to build up a more comprehensive and expressive inter-frame communication at hybrid pixel and instance level with unified attention operations. Figure 1 illustrates an example of the inter-frame attentions between instance code and feature maps. The instance code activates pixels belonging to the same instance across frames; instance code tends to activate other codes of the same instance; pixels of the same instance get feedback attention from instance code. Second, recent literature (Carion et al. 2020; Wang et al. 2021a) has shown that the instance code tends to be ordered and correlated with each other regarding to instance position and class. This inherent property de facto fits the instance consistency for VIS task in the sense that instance prediction tend to be consistent across frames.

Our main contributions are summarized as follows:

- We propose a box-free, detection-free and matching-free online VIS framework. It abandons complicated matching operations as well as heavy decoders for matching instance identities in previous online VIS methods.
- We introduce an instance-aware temporal fusion method, which enables instance-level feature aggregation without cropping feature maps, bringing a new way to combining historical information for instance segmentation.
- Our model achieves state-of-the-art results on Youtube-VIS 2019 and Youtube-VIS 2021 with 41.3 mAP and 35.8 mAP respectively, outperforming previous state-of-art methods by a large margin.

Related Works

Video Instance Segmentation

Video instance segmentation requires classifying and segmenting each instance in a frame and assigning the same instance with the same identity across frames. The methods proposed for the VIS task work either in an online or offline fashion. For online methods, Mask-Track-RCNN (Yang, Fan, and Xu 2019) is the first attempt to address the VIS problem which extends the Mask-RCNN (He et al. 2017) with a tracking head to associate instance identities. Followed by Mask-Track-RCNN, SipMask (Cao et al. 2020) and SG-Net (Liu et al. 2021) build the tracking head on top of the modified one-stage still-image instance segmentation method FCOS (Tian et al. 2019) and Blender-Mask (Chen et al. 2020) to achieve better speed and performance. CrossVIS (Yang et al. 2021) introduces the cross-over learning scheme and global instance embedding to learn better features for robust instance tracking and segmentation. Different from online methods, another track of VIS takes the entire video as input and works in an offline fashion. MaskProp leverages Hybrid Task Cascade Network (Bertasius and Torresani 2020) and heavily post-processing to associate and refine the predictions from Mask-RCNN (He et al. 2017) for VIS task. More recently, VISTR (Wang et al. 2021b) brings a new way to tackle the VIS problem, which utilizes a transformer encoder on top of the convolutional backbone and match instance identities by transformer decoder. In VISTR, the same instances are predicted by a set of pre-defined slots in the instance queries thus it can inference in an end-to-end manner without redundant matching operations.

Vision Transformer

Transformer (Vaswani et al. 2017), proposed for neural machine translation, has been widely used in the computer vision field. The transformer shows great generalization and has been adapted in multiple tasks, such as classification (Chen et al. 2018b; Bello et al. 2019; Ramachandran et al. 2019), image segmentation (Wang et al. 2021a,b), object detection (Carion et al. 2020; Zhu et al. 2020), image generation (Parmar et al. 2018; Cornia et al. 2020; Yang et al. 2020) and video recognition (Neimark et al. 2021; Girdhar et al. 2019). Since the attention mechanism and fully connection layers (FC) in transformer make it computational expensive, most methods only apply transformer on down-sampled feature maps. For example, DETR (Carion et al. 2020) and VISTR (Wang et al. 2021b) overlap a transformer on top of a CNN to conduct end-to-end object detection and segmentation respectively. On other hand, MaX-Deeplab (Wang et al. 2021a) inserts the transformer into the CNN and enables communication between different stages of it.

Method

Overview

We leverage the transformer-based network to tackle the VIS problem in a bottom-up online fashion. Unlike previous methods (Wang et al. 2021b) which simply stack a transformer on top of the convolutional neural network (CNN), we insert the transformer layers in the CNN by leveraging a stand-alone latent code to encode the instance information. Figure 2 shows an overview of our method. We utilize intra-frame attention layers to extract instance code for the current frame and propagate the historical information by involving inter-frame and intra-frame attention layers alternatively. We then use skip connections (Ronneberger, Fischer, and Brox 2015) to get low-level contextual information and use dynamic convolution (Jia et al. 2016; Tian, Shen, and

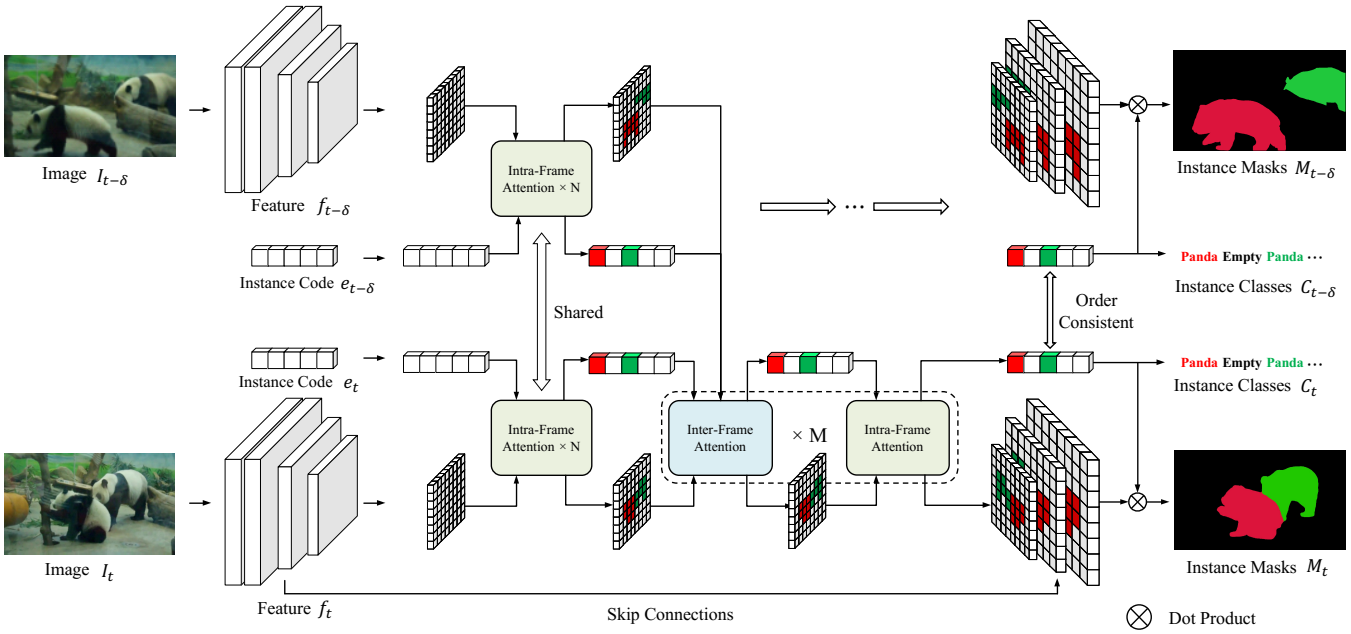

Figure 2: Overview of the proposed framework. We enforce the temporal consistency in VIS by introducing hybrid F2F communications. Two main components are highlighted, i.e., intra-frame attention for linking current instance code and feature maps, and inter-frame attention for fusing hybrid (pixel- and instance-level) temporal information in adjacent frames. The first N intra-frame attention layers are integrated into the convolutional backbone followed by M alternate attention layers. The final instance codes are constrained to be consistent across frames.

Chen 2020) to generate the final segmentation maps. Here, no sophisticated matching or post-processing is required for instance identity matching as they directly correspond to the (frame-consistent) code slot indices.

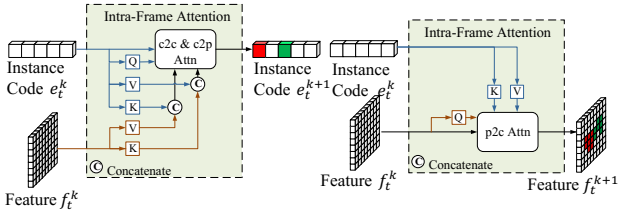

Figure 3: Intra-frame attention. The intra-frame attention at k-th layer links current frame instance code e_t^k and features f_t^k. The c2c & c2p attention probes high-level instance relevant features and fuses them back to the instance code. The p2c attention adjusts the pixel feature map based on the instance code.

Hybrid Representation for Video Frame

We unitize a global latent code (i.e. instance code) as well as CNN feature maps to jointly represent instance-aware features for each frame.

Instance code. Inspired by the works (Jaegle et al. 2021; Wang et al. 2021a), which use a latent space to encode task-specific information, we introduce instance code e, a $L \times D$ vector to VIS task, where the L is the maximum detected instance number in a frame and D is the feature dimension for each instance. Our instance code represents both the class and mask information of one instance for each slot in an order-aware fashion; thus, we can directly use slot indices to represent instance identities. This differs our approaches from MaX-Deeplab (Wang et al. 2021a) which takes the code as L permutation invariant ones and (Yang, Fan, and Xu 2019) which only represents instance appearance information.

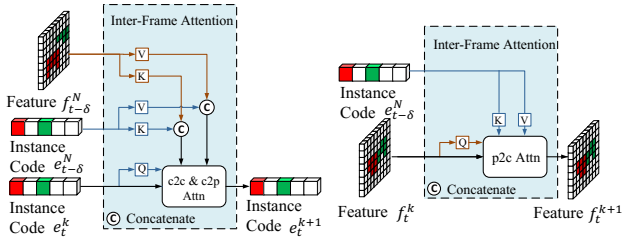

Figure 4: Inter-frame attention. The intra-frame attention at k-th layer enables communication between current frame features e_t^k, f_t^k and past frame features $e_{t-\delta}^k$, $f_{t-\delta}^k$. The c2c & c2p and p2c attentions fuse instance relevant features to instance code and pixel feature map respectively.

Instance-aware Frame Communication

We enhance the frame representations with frame-wise correlation at both instance level and pixel level with multiple types of attention layers based on the transformer.

Intra-frame attention. Intra-frame attention layer constructs the relation between instance code and feature map by three types of attention - code-to-code (c2c), code-to-pixel (c2p), and pixel-to-code (p2c). The c2c and c2p attention aim to probe the instance relevant feature from pixel feature map and instance code respectively. The p2c attention adjusts the pixel feature map based on the instance code.

As shown in Figure 3, given a pixel feature map at k-th layer $f_t^k \in \mathbb{R}^{H\times W\times D_f}$ with height H, width W, dimension D_f and an instance code $e_t^k \in \mathbb{R}^{L\times D_e}$ with length L, dimension D_e, we conduct the c2c and c2p attention simultaneously. We learn the query Q_t^e, key K_t^e, K_t^f and value V_t^e, V_t^f by linear projections from code e_t^k and feature f_t^k respectively. The output of intra-frame c2c & c2p attention can be computed as

$$e_{intra} = \sum_{n=1}^{HW+L} \text{softmax}(Q_t^e \cdot K_t^{e\oplus f}) V_t^{e\oplus f} \tag{1}$$

where $K_t^{e\oplus f} = [K_t^e; K_t^f]$ and $V_t^{e\oplus f} = [V_t^e; V_t^f]$. For simplicity, we will omit the layer numbers in Equation 1 hereafter. Specifically, the output e_{intra} is fused to input e_t by residual connection.

Similarly, the output of intra-frame p2c attention can be computed as

$$f_{intra} = \sum_{n=1}^{L} \text{softmax}(Q_t^f \cdot K_t^e) V_t^e \tag{2}$$

Like intra-frame c2c & c2p attention, the output of p2c attention f_{intra} is fused to the input f_t by residual connection.

Inter-frame attention. Inter-frame attention aims to construct temporal correlation and fuse contextual features across frames. Our key insight for inter-frame attention is that, given a clip of video frames, the same instance may appear in slightly different spatial localization while the appearance of it should be similar. Therefore, the instance code of adjacent frames should encode similar representations. To the end, we propose inter-frame c2c, c2p and p2c attentions to combine temporal information.

Specifically, given a target frame I_t and a set of reference $\mathcal{R} = \{I_{t-\delta}\}$, $\delta = 1, \ldots, N_{ref}$, we denote the instance code and feature map of target frame as e_t and f_t; for reference frame, the instance code and feature map after the first N intra-frame attention layers can be denote as $\{e_{t-\delta}^N\}$ and $\{f_{t-\delta}^N\}$, $\delta = 1, \ldots, N_{ref}$.

To compute inter-frame attention with multiple reference frames, we first concatenate all N_{ref} reference codes ($\{e_{t-\delta}^N\}_{\delta=1}^{N_{ref}}$) and feature maps ($\{f_{t-\delta}^N\}_{\delta=1}^{N_{ref}}$) and then add learnable positional encoding (Dosovitskiy et al. 2020) to them. Let us denote the processed reference inputs as $e_{ref}^N \in \mathbb{R}^{N_{ref}L\times D_e}$ and $f_{ref}^N \in \mathbb{R}^{N_{ref}H\times W\times D_f}$. Similarly, we add learnable positional encoding to the target inputs and denote them as $e_{tgt} \in \mathbb{R}^{L\times D_e}$ and $f_{tgt} \in \mathbb{R}^{H\times W\times D_f}$. After that, we project the instance codes and feature maps to corresponding queries, keys, and values by linear transformations. Figure 4 illustrates an example with $N_{ref} = 1$ reference frame.

We fuse temporal information to target frame code by inter-frame c2c & c2p attention. Specifically, a query Q_{tgt}^e computed from target code is leveraged to probe the reference code and feature map. Let us denote keys and values from reference code and feature map as K_{ref}^e, K_{ref}^f and V_{ref}^e, V_{ref}^f respectively. The output of inter-frame c2c & c2p attention can be computed as

$$e_{inter} = \sum_{n=1}^{HW+L} \text{softmax}(Q_{tgt}^e \cdot K_{ref}^{e\oplus f}) V_{ref}^{e\oplus f} \tag{3}$$

where $K_{ref}^{e\oplus f} = [K_{ref}^e; K_{ref}^f]$ and $V_{ref}^{e\oplus f} = [V_{ref}^e; V_{ref}^f]$. To adjust pixel feature map accordingly, inter-frame p2c attention is utilized and can be computed as

$$f_{inter} = \sum_{n=1}^{L} \text{softmax}(Q_{tgt}^f \cdot K_{ref}^e) V_{ref}^e \tag{4}$$

Compared to previous pixel-level temporal attention modules (Fu et al. 2020; Yang, Wei, and Yang 2020), our code-based attention introduces the instance-aware fusion without redundant feature cropping, which enables us to leverage the feature-level consistency throughout time.

Network Design

Our method employs an encoder-decoder-based structure equipped with a transformer integrated in the encoder.

Encoder. Unlike previous transformer-based VIS methods that overlap a transformer on top of the backbone, we insert the intra-frame attention layers into the last stage of the ResNet-50 (He et al. 2016) backbone to better extract instance code. Following (Wang et al. 2021a), we also use additional pixel-wise attention in intra-frame layers in the backbone to better extract global information for segregated or partially occluded masks in the VIS task.

After intra-frame attention, both the instance code and pixel-level feature maps are fed into the inter-frame attention layer to fuse temporal context. Following the conventional transformer, we iteratively repeat the inter-frame and intra-frame attention M times before the decoder.

Decoder. We follow the decoder structure of Deeplab-V3+ (Chen et al. 2018a) to fuse the low-level features. From the decoder features, we predict instance classes and instance masks separately. For instance class prediction, two fully connected layers and a softmax are applied on instance code before the final instance class output. For instance mask prediction, we leverage dynamic convolution (Jia et al. 2016) to correspond the instance code with masks. In particular, we learn dynamic filters $\theta_t \in \mathbb{R}^{N_e\times D_{fout}}$ from instance code by another two fully connection layers, where N_e is the length of instance code and D_{fout} the dimension of the upsampled feature map. Let the upsampled feature map be $f_{out} \in \mathbb{R}^{H_o\times W_o\times D_{fout}}$. The mask prediction $M_t \in \mathbb{R}^{N_e\times H_o\times W_o}$ can be compute as

$$M_t = softmax(\theta_t f_{out}^T) \tag{5}$$

where the softmax is applied on the N_e dimension.

Method	Backbone	AP	AP50	AP75	AR1	AR10
Offline Methods						
STEm-Seg (Athar et al. 2020)	ResNet-50	30.6	50.7	33.5	31.6	37.1
STEm-Seg (Athar et al. 2020)	ResNet-101	34.6	55.8	37.9	34.4	41.6
MaskProp (Bertasius and Torresani 2020)	ResNet-50	40.0	-	42.9	-	-
VisTR (Wang et al. 2021b)	ResNet-50†	36.2	59.8	36.9	37.2	42.4
VisTR (Wang et al. 2021b)	ResNet-101†	40.1	64.0	45.0	38.3	44.9
Purpose-Reduce (Lin et al. 2021)	ResNet-50	40.4	63.0	43.8	41.1	49.7
Online Methods						
MaskTrack R-CNN (Yang, Fan, and Xu 2019)	ResNet-50	30.3	51.1	32.6	31.0	35.5
MaskTrack R-CNN (Yang, Fan, and Xu 2019)	ResNet-101	31.9	53.7	32.3	32.5	37.7
SipMask (Cao et al. 2020)	ResNet-50	33.7	54.1	35.8	35.4	40.1
CompFeat (Fu et al. 2020)	ResNet-50	35.3	56.0	38.6	33.1	40.3
STMask (Li et al. 2021a)	ResNet-50	33.5	52.1	36.9	31.1	39.2
STMask (Li et al. 2021a)	ResNet-101	36.8	56.8	38.0	34.8	41.8
SG-Net (Liu et al. 2021)	ResNet-50	34.8	56.1	36.8	35.8	40.8
SG-Net (Liu et al. 2021)	ResNet-101	36.3	57.1	39.6	35.9	43.0
QueryInst (Fang et al. 2021)	ResNet-50	36.2	57.3	39.7	36.0	42.0
CrossVIS (Yang et al. 2021)	ResNet-50	36.3	56.8	38.9	35.6	40.7
CrossVIS (Yang et al. 2021)	ResNet-101	36.6	57.3	39.7	36.0	42.0
Ours	ResNet-50‡	41.3	61.5	43.5	42.7	47.8

Table 1: Comparison to state-of-the-art video instance segmentation on Youtube-VIS-2019 val set. ResNet-50† and ResNet-101† means transformer on top of ResNet-50 and ResNet-101 respectively. ResNet-50‡ indicates ResNet-50 with intra-frame and inter-frame layers.

Loss Function

As an online VIS method, one main challenge of our method is to maintain the instance identity consistent throughout the video sequence. To tackle this problem, we consider the instance predictions in both current and past frames during training.

To train the network, we assign a ground-truth to each instance prediction by searching a permutation of L elements $\sigma \in \mathcal{S}_L$ with the highest similarity. Let us denote the L instance predictions in arbitrary time τ as $\{y_\tau^i\}_{i=1}^L = \{p_\tau^i(c), m_\tau^i\}_{i=1}^L$ and the ground-truths $\{\hat{y}_\tau\}_{i=1}^L = \{\hat{c}_\tau^i, \hat{m}_\tau^i\}_{i=1}^L$, where $p_\tau^i(c)$ and m_τ^i represents the probability of class c (including $\emptyset$) and mask of the i-th instance respectively. Since VIS tasks assume the prediction as true positive only if it has accurate class prediction as well as mask prediction, we calculate the similarity as

$$\text{Sim} = \text{Sim}_{mask} \times \text{Sim}_{cls} \tag{6}$$

where the Sim_{mask} and Sim_{cls} are similarity in terms of mask and class respectively. To retain the order of instance code, we consider t and $t-1$ frames simultaneously when compute the similarity. In particular, the Sim_{mask} and Sim_{cls} can be computed as

$$\text{Sim}_{mask} = \mathbb{1}_{\{c_t^i \neq \emptyset\}} \text{Dice}([m_t^{\sigma(i)}, m_{t-1}^{\sigma(i)}], [\hat{m}_t^i, \hat{m}_{t-1}^i]) \tag{7}$$

$$\text{Sim}_{cls} = \mathbb{1}_{\{c_t^i \neq \emptyset\}} [p_t^{\sigma(i)}(\hat{c}_t^i) + p_{t-1}^{\sigma(i)}(\hat{c}_{t-1}^i)] \tag{8}$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function and σ is a permutation of slot indices. Dice indicates the Dice loss (Milletari, Navab, and Ahmadi 2016).

Given the optimal assignment $\hat{\sigma}$, the total loss $\mathcal{L}$ can be boiled down to three main components, $\mathcal{L}_{pos}$ for the K matched instance predictions, $\mathcal{L}_{neg}$ for the $L-K$ unmatched instance predictions, and auxiliary losses $\mathcal{L}_{aux}$ (Wang et al. 2021a) to facilitate the training:

$$\mathcal{L} = \lambda_{inst}\mathcal{L}_{pos} + (1-\lambda_{inst})\mathcal{L}_{neg} + \lambda_{aux}\mathcal{L}_{aux} \tag{9}$$

where λ_{inst} and λ_{aux} are scalars to balance the losses, $\mathcal{L}_{neg} = -\sum_{K+1}^{N_e} \log p_t^{\hat{\sigma}(i)}(\emptyset)$ and $\mathcal{L}_{pos} = -\sum_{i=1}^{K}[k_{mask} \cdot \text{Dice}(\text{m}_\text{t}^{\hat{\sigma}(\text{i})}, \hat{\text{m}}_\text{t}^\text{i}) + k_{cls} \cdot \log p_t^{\hat{\sigma}(i)}(\hat{c}_i)]$. The k_{mask} and k_{cls} are weights for mask term and class term respectively.

Instance Identity Matching

Different from previous online methods that crop instances then leverage multiple cues to match instances across frames, we directly assume the non-empty predictions from the same slot of the instance code have the same identity. To enhance model robustness, if the mask in i-th slot in time t has an IoU larger than 0.5 with mask in j-th slot in time $t-1$, we directly assume they share the same identity without considering the slot indices.

Experiment

Dataset and Evaluation Metric

We evaluate our method on two extensively used VIS datasets - Youtube-VIS-2019 and Youtube-VIS-2021.

- **Youtube-VIS-2019** Youtube-VIS-2019 has 40 categories, 4,883 unique video instances, and 131k high-quality manual annotations. There are 2,238 training videos, 302 validation videos, and 343 test videos in it.
- **Youtube-VIS-2021** Youtube-VIS-2021 is an improved version of the Youtube-VIS-2019 dataset, which contains 8,171 unique video instances and 232k high-quality manual annotations. There are 2,985 training videos, 421 validation videos, and 453 test videos in this dataset.

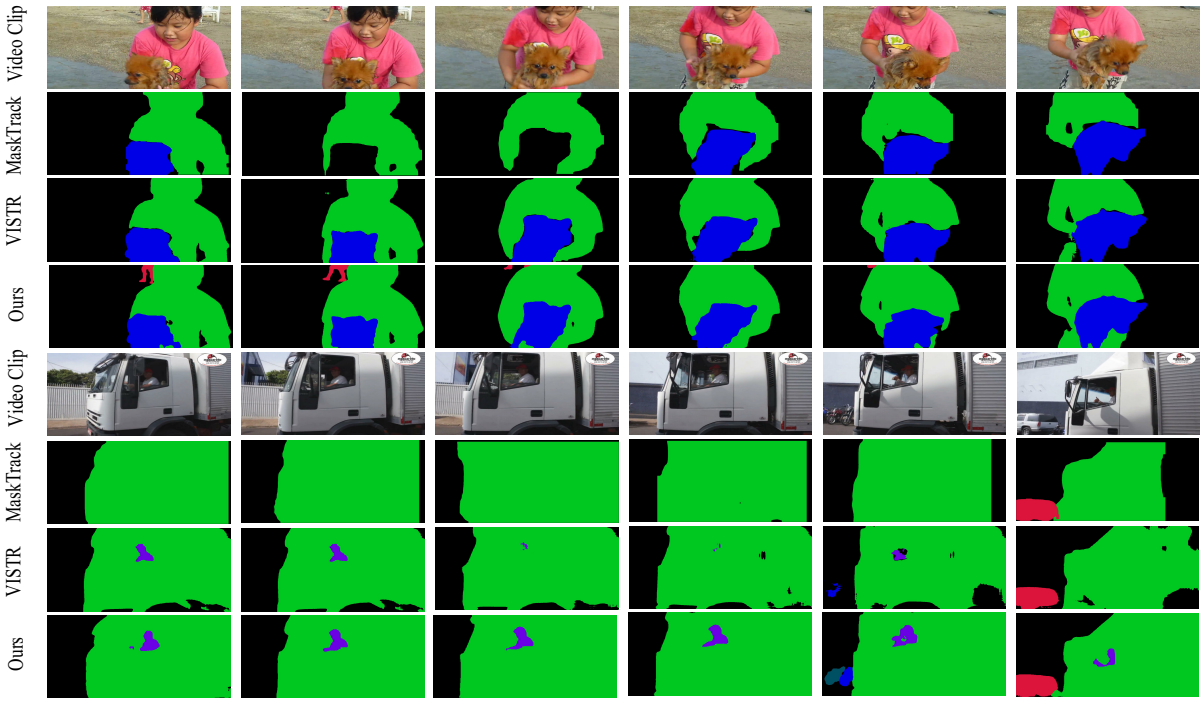

Figure 5: Comparison to state-of-the-art video instance segmentation methods, MaskTrack (Yang, Fan, and Xu 2019) and VISTR (Wang et al. 2021b). Each row represents the same video clip. For each clip, colors indicate instance identities (best viewed in color). We show that our method generates more accurate and temporally consistent results, while MaskTrack R-CNN and VISTR attempt to miss instances for the cases where instances are overlapped or small.

The evaluation metric for this task is defined as the area under the precision-recall curve with different IoUs as thresholds.

Implementation Details

We implement our method with the Tensorflow2 framework. Following previous methods (Fu et al. 2020; Lin et al. 2021), we first pre-train our model with both Youtube-VIS and overlapped categories on COCO dataset (Lin et al. 2014) then finetune the model on the Youtube-VIS dataset. All frames are resized and padded to 641×641 during training and inference. We train our model for 35k iterations with a "poly" learning rate policy where the learning rate is multiplied by $(1 - \frac{iter}{iter_{max}})^{0.9}$ for each iteration with an initial learning rate of 0.001 to all experiments. The $\text{batchsize} = 32$ and an adam (Kingma and Ba 2014) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\text{weight decay} = 0$ is leveraged. Multi-scale training is adopted to obtain a strong baseline. We select adjacent three frames as reference frames if not specific.

Main Results

In this section, we compare our method with previous state-of-art methods in terms of Youtube-VIS 2019 and Youtube-VIS 2021 datasets.

Quantitative results. We compare our method against state-of-the-art VIS methods on Youtube-VIS 2019 dataset in Table 1. (1) Compared with online methods. Notably, our method achieves a mAP of 41.3 which is the best among all online methods and eclipses the previous state-of-art online method CrossVIS by a large margin even if it is with a stronger ResNet-101 backbone. (2) Compared with offline methods. Our method also outperforms all offline VIS methods when using the same ResNet-50 backbone. For another transformer-based method VisTR, which leverages a transformer on top of the backbone and matches instance with a sequential level strategy, our method outperforms it by 4.3 mAP when using the same ResNet-50 backbone and 1.2 mAP when it is using the stronger ResNet-101 backbone. Moreover, we also report the results on the recently introduced Youtube-VIS 2021 dataset in Table 2. Our method also achieves the best result, 35.8 mAP, on the newly imported dataset.

Qualitative results. We present our qualitative result in Figure 5 and compare it against VisTR (Wang et al. 2021b). The result indicates that VisTR fails to segment and track instances when they are overlapped. In contrast, our method shows great accuracy and robustness even in very challenging scenarios. This implies that our transformer-based network equipped with instance code generates more accurate and temporally consistent results than simply adopting conventional transformer architecture (Vaswani et al. 2017) on top of the conventional backbone.

Method	AP	AP50	AP75	AR1	AR10
MaskTrack	28.6	48.9	29.6	26.5	33.8
SipMask	31.7	52.5	34.0	30.8	37.8
CrossVIS	34.2	54.4	37.9	30.4	38.2
Ours	35.8	56.3	39.1	33.6	40.3

Table 2: Comparison to state-of-the-art video instance segmentation on Youtube-VIS-2021 val set. All results presented are generated by ResNet-50 backbone.

Layer Number	AP	AP50	AP75	AR1	AR10
1	40.1	61.1	41.9	41.4	46.2
2	41.3	61.5	43.5	42.7	47.8
3	40.8	61.8	42.0	41.9	46.4
4	40.2	61.0	41.9	41.6	46.1

Table 4: Alternate attention layers. Note that the first N intra-frame attention are not included in the layer number.

Frame	AP	AP50	AP75
1	39.7	60.9	41.2
2	40.5	59.7	42.1
3	41.3	61.5	43.5
4	41.3	61.7	43.5

Table 6: Reference frame number. The reference frames are selected from the adjacent frames before target frame.

Ablation Study

We conduct extensive ablation studies on Youtube-VIS-2019 to show the effectiveness of different components of our method.

Inter-frame attentions. To investigate the effectiveness of the inter-frame attention layer, we train models by disabling different attentions. As shown in Table 3, the model performance plummets to 36.9 mAP when disabling all inter-frame attentions. If only adopting inter-frame p2c attention, the performance will also drop about 2.0 mAP compared to the default setting. In addition, we found that when we leverage all attention types in the inter-frame layer, the pair-wise matching strategy only brings a marginal gain to the overall performance. This indicates that by using the inter-frame attentions, the model can automatically learn temporal consistency without additional supervision.

Alternate temporal transformer layer number. We design our temporal transformer referring to the conventional transformer architecture (Vaswani et al. 2017), where the inter-frame layers in reference frames correspond to the conventional transformer encoder and the alternate transformer layers in the target frame correspond to the conventional transformer decoder. We ablate on the alternate transformer layers number to investigate how many layers are enough to fuse the temporal features. Table 4 shows the results of different numbers of transformer layers. We found that a small number of 2 leads to the best performance. This demonstrates that it is easier to learn better features by using a swallow temporal fusion network.

PM	p2c	c2c&c2p	AP	AP50	AP75
✓			36.9	57.9	40.5
	✓		38.8	59.7	40.4
✓	✓		39.4	60.6	41.4
	✓	✓	40.7	61.3	42.7
✓	✓	✓	41.3	61.5	43.5

Table 3: Inter-frame attention. PM, p2c and c2c&c2p represents pair-wise matching strategy, inter-frame p2c attention and inter-frame c2c & c2p attention respectively.

Slot Number	AP	AP50	AP75	AR1	AR10
10	41.3	61.5	43.5	42.7	47.8
15	41.2	61.9	43.5	42.3	47.6
20	41.1	62.6	42.5	41.9	47.8
25	41.3	61.2	44.0	42.7	47.2

Table 5: Slot number of instance embedding. Youtube-VIS-2019 has a maximum of 10 and a minimum of 1 instance in a frame.

Mask slot number. The slot number of instance code represents the maximum number of detected instances in a frame. When the instance number in a frame is smaller than the slot number, the remaining unmatched slots will predict empty class and be assumed as negative samples to be supervised separately. Since the inter-frame attention calculates the attention map separately for each slot in the instance code while conducting softmax among all slots, the redundant slots in the instance code may influence the overall performance. However, as shown in Table 5, we found the performance retaining robust even with numerous redundant slots existing. The results indicate that both intra-frame and inter-frame attentions are robust to redundant slots. This property enables our method to handle the scenario that has extremely few instances.

Reference frame number. To investigate the importance of the temporal information to our method, we conduct experiments by teasing the reference frame number to the inter-frame attention layer. As illustrated in Table 6, with the reference frame number varying from 1 to 4, the mAP of our method gradually increases from 39.7 to 41.3, which verifies the necessity of the temporal information to our model.

Conclusion

In this paper, we propose a novel instance-aware temporal fusion method for VIS, which enables the temporal fusion at hybrid level (pixel-, instance-, cross-level). Based on that, we further simplify the instance identity association operation. Notably, our method achieves the best result among both Youtube-VIS-2019 and Youtube-VIS-2021 benchmarks. Moreover, extensive study shows that our instance-aware temporal fusion leads to remarkable improvement to the VIS performance.

References

Athar, A.; Mahadevan, S.; Osep, A.; Leal-Taixé, L.; and Leibe, B. 2020. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *European Conference on Computer Vision*, 158–177. Springer.

Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; and Le, Q. V. 2019. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3286–3295.

Bertasius, G.; and Torresani, L. 2020. Classifying, segmenting, and tracking object instances in video with mask propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9739–9748.

Cao, J.; Anwer, R. M.; Cholakkal, H.; Khan, F. S.; Pang, Y.; and Shao, L. 2020. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, 1–18. Springer.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 213–229. Springer.

Chen, H.; Sun, K.; Tian, Z.; Shen, C.; Huang, Y.; and Yan, Y. 2020. BlendMask: Top-down meets bottom-up for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8573–8581.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018a. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; and Feng, J. 2018b. A^2-Nets: Double Attention Networks. *arXiv preprint arXiv:1810.11579*.

Cornia, M.; Stefanini, M.; Baraldi, L.; and Cucchiara, R. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10578–10587.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fang, Y.; Yang, S.; Wang, X.; Li, Y.; Fang, C.; Shan, Y.; Feng, B.; and Liu, W. 2021. Instances as Queries. arXiv:2105.01928.

Fu, Y.; Yang, L.; Liu, D.; Huang, T. S.; and Shi, H. 2020. Compfeat: Comprehensive feature aggregation for video instance segmentation. *arXiv preprint arXiv:2012.03400*.

Girdhar, R.; Carreira, J.; Doersch, C.; and Zisserman, A. 2019. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 244–253.

Gong, T.; Chen, K.; Wang, X.; Chu, Q.; Zhu, F.; Lin, D.; Yu, N.; and Feng, H. 2021. Temporal ROI Align for Video Object Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1442–1450.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Jaegle, A.; Gimeno, F.; Brock, A.; Zisserman, A.; Vinyals, O.; and Carreira, J. 2021. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*.

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Gool, L. V. 2016. Dynamic filter networks. *Advances in neural information processing systems*, 29: 667–675.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, M.; Li, S.; Li, L.; and Zhang, L. 2021a. Spatial Feature Calibration and Temporal Fusion for Effective One-stage Video Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11215–11224.

Li, X.; Wang, J.; Li, X.; and Lu, Y. 2021b. Video Instance Segmentation by Instance Flow Assembly. *arXiv preprint arXiv:2110.10599*.

Lin, H.; Wu, R.; Liu, S.; Lu, J.; and Jia, J. 2021. Video instance segmentation with a propose-reduce paradigm. *arXiv preprint arXiv:2103.13746*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, D.; Cui, Y.; Tan, W.; and Chen, Y. 2021. SG-Net: Spatial Granularity Network for One-Stage Video Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9816–9825.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, 565–571. IEEE.

Neimark, D.; Bar, O.; Zohar, M.; and Asselmann, D. 2021. Video transformer network. *arXiv preprint arXiv:2102.00719*.

Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; and Tran, D. 2018. Image transformer. In *International Conference on Machine Learning*, 4055–4064. PMLR.

Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; and Shlens, J. 2019. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Tian, Z.; Shen, C.; and Chen, H. 2020. Conditional convolutions for instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 282–298. Springer.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9627–9636.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, H.; Zhu, Y.; Adam, H.; Yuille, A.; and Chen, L.-C. 2021a. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5463–5474.

Wang, Y.; Xu, Z.; Wang, X.; Shen, C.; Cheng, B.; Shen, H.; and Xia, H. 2021b. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8741–8750.

Yang, F.; Yang, H.; Fu, J.; Lu, H.; and Guo, B. 2020. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5791–5800.

Yang, L.; Fan, Y.; and Xu, N. 2019. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5188–5197.

Yang, S.; Fang, Y.; Wang, X.; Li, Y.; Fang, C.; Shan, Y.; Feng, B.; and Liu, W. 2021. Crossover learning for fast online video instance segmentation. *arXiv preprint arXiv:2104.05970*.

Yang, Z.; Wei, Y.; and Yang, Y. 2020. Collaborative video object segmentation by foreground-background integration. In *European Conference on Computer Vision*, 332–348. Springer.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.