Video Instance Segmentation with a Propose-Reduce Paradigm

Huaijia Lin[1*] Ruizheng Wu[1*] Shu Liu[2] Jiangbo Lu[2] Jiaya Jia[1,2]
[1]The Chinese University of Hong Kong [2]SmartMore
{linhj, rzwu, leojia}@cse.cuhk.edu.hk {sliu, jiangbo}@smartmore.com

Abstract

Video instance segmentation (VIS) aims to segment and associate all instances of predefined classes for each frame in videos. Prior methods usually obtain segmentation for a frame or clip first, and merge the incomplete results by tracking or matching. These methods may cause error accumulation in the merging step. Contrarily, we propose a new paradigm – Propose-Reduce, to generate complete sequences for input videos by a single step. We further build a sequence propagation head on the existing image-level instance segmentation network for long-term propagation. To ensure robustness and high recall of our proposed framework, multiple sequences are proposed where redundant sequences of the same instance are reduced. We achieve state-of-the-art performance on two representative benchmark datasets – we obtain 47.6% in terms of AP on YouTube-VIS validation set and 70.4 % for J&F on DAVIS-UVOS validation set.

1. Introduction

Video instance segmentation (VIS), proposed in [53], is a task to segment all instances of the predefined classes in each frame. Segmented instances are linked in the entire video. It is important in the field of video understanding, and can be applied to video editing, autonomous driving, *etc*. Unlike image-level instance segmentation, VIS requires not only detection and segmentation of each frame, but also tracking of objects in the video, which make it a very challenging task.

Recently, several approaches were proposed for this task [53, 7, 1, 2, 29]. Based on the patterns of generating instance sequences, existing frameworks can be roughly categorized into two paradigms: 'Track-by-Detect' (Fig. 1(a)) and 'Clip-Match' (Fig. 1(b)). The 'Track-by-Detect' paradigm detects and segments instances for each individual frame, followed by obtaining instance sequences with frame-by-frame tracking [53, 7]. Differently, 'Clip-Match' adopts the divide-and-conquer strategy. It divides an entire video into multiple short overlapped clips, and obtains VIS results for each clip and generates instance sequences with clip-by-clip matching [2, 1]. Both of the paradigms need two independent steps to generate a complete sequence. They both generate multiple incomplete sequences (*i.e*., frames or clips) from a video, and merge (or complete) them by tracking/matching at the second stage. Intuitively, these paradigms are vulnerable to error accumulation in the process of merging sequences, especially when occlusion or fast motion exists.

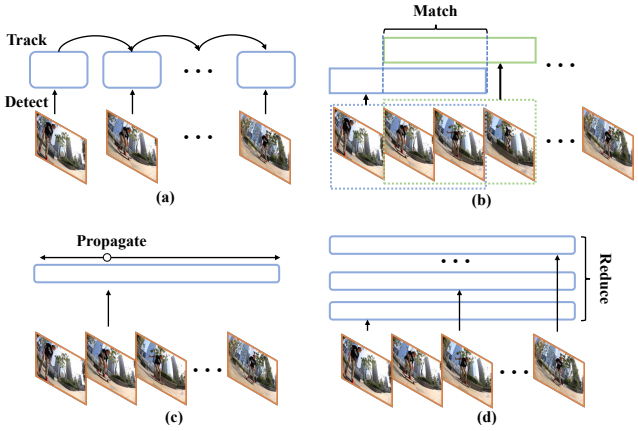

Figure 1. Four paradigms of generating instance sequences in VIS. (a) **Track-by-Detect** links detected instances via frame-by-frame tracking. (b) **Clip-Match** matches overlapped sub-sequences between video-clips. (c) **An alternative** propagates detected instances from one key frame to the rest of a video. (d) Our proposed paradigm, named **Propose-Reduce**, generates instance sequence proposals based on multiple key frames and reduces redundant sequences of the same instances.

To avoid error accumulation brought by merging incomplete sequences, one intuitive solution is to generate a complete instance sequence for an entire video with only one step. As shown in Fig. 1(c), starting from any key frame of a video, we can obtain instance sequences by propagating the instance segmentation results from this frame to all others. However, the propagation quality from different start-

*Equal Contribution.

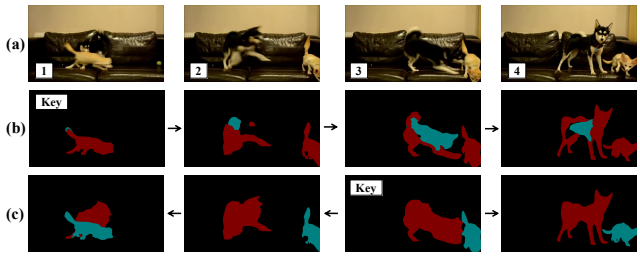

Figure 2. Effect of propagation from different key frames. (a) Four ordered frames where severe occlusion occurs in the first frame. (b) Propagating from the first frame leads to inaccurate segmentation results caused by error accumulation. (c) Propagating from the third frame produces satisfactory segmentation masks, due to reasonable segmentation results in the key frame.

ing key frames varies a lot (as shown in Fig. 2). One key frame may only contain part of the instances in a video, inappropriate for the whole sequence.

For robust propagation and high recall to cover enough instances, we propose a new paradigm, named *Propose-Reduce* (Fig. 1(d)). It first produces sequence proposals from multiple key frames and reduces the redundant sequence proposals of the same instances. This paradigm not only discards the step of merging incomplete sequences, but also achieves robust results considering multiple key frames.

The idea behind Propose-Reduce is proved effective in the task of image-level object detection. Methods for this task can be classified as one-stage [36, 25] and two-stage [37, 21] detection frameworks. Compared with the one-stage frameworks, the two-stage ones first generate a large number of candidate proposals by the region proposal network (RPN) [37] to ensure high recall, and then reduce abundant proposals by non-maximum suppression (NMS) as post-processing. The great performance of the two-stage detection frameworks shows the potential of our Propose-Reduce in the video domain.

Based on the above analysis, to propagate instance segmentation from each key frame to all other frames, we design an additional module for long-term propagation, since a frame to be propagated can be far from the key frame. We propose attaching a sequence propagation head (*Seq-Prop head*) upon a widely-used image-level instance segmentation network: Mask R-CNN [14]. It enables sharing backbone features for multiple heads of different functions of classification head, bounding box head, mask head and sequence propagation head. With the sharing backbone, our propagation module is light-weighted.

Besides, we adopt a memory propagation strategy on every key frame to enable long-term propagation. After obtaining sequence proposals from all key frames, we implement a variant of NMS to reduce redundant proposals at the sequence level. With the above design, our overall framework is neat and can be trained in an end-to-end fashion. The overall contributions are summarized below.

- We propose a new paradigm – *Propose-Reduce*, for the task of video instance segmentation. This paradigm ensures high recall and does not require error-accumulating tracking/matching modules.
- Based on the paradigm, we propose a variant of Mask R-CNN for videos, named *Seq Mask R-CNN*. By adding an extra sequence propagation head upon Mask R-CNN, temporal relation is established across frames.
- Our framework achieves new state-of-the-art results on YouTube-VIS [53] validation set with 47.6% in $\mathcal{AP}$, as well as DAVIS-UVOS [6] validation set with J&F score 70.4 %.

2. Related Work

Image-Level Instance Segmentation Image-level instance segmentation is a classical computer vision task with many solutions [14, 16, 24, 47, 10, 51, 4, 35] proposed. They can be mainly divided into top-down [14, 16, 24, 4, 18], bottom-up [23, 30], and direct segmentation methods [47, 51]. Among these methods, top-down structure is popular for high performance. It first utilizes detectors to detect objects and then segments them based on detected bounding boxes.

One representative method is Mask R-CNN [14]. It is built on a two-stage detector [37], adds a mask head for segmentation upon the detector, and keeps the original classification head as well as the bounding box head in the detector. We design our framework based on Mask R-CNN from image to video domain. We introduce an extra sequence propagation head as well as a new paradigm for both spatial and temporal processing in the video instance segmentation task. Our method is simple and surprisingly effective.

Video Instance Segmentation Video instance segmentation (VIS) was introduced in [53], which requires classification, segmentation and tracking on instances simultaneously in a video. existing work [53, 7, 1, 2, 29, 19] can be divided into two types based on the way of sequence generation.

One straight-forward paradigm is 'Track-by-Detect' [53, 7, 29] with two parts: detection and tracking. In the detection part, instances are located with existing image-level instance segmentation methods [14] in a frame-by-frame manner. The detected instances are associated among different frames in the tracking part. Another paradigm is summarized as 'Clip-Match' based methods [1, 2]. It divides an entire video into multiple short clips and completes the VIS task in a clip-by-clip manner via propagation [2]

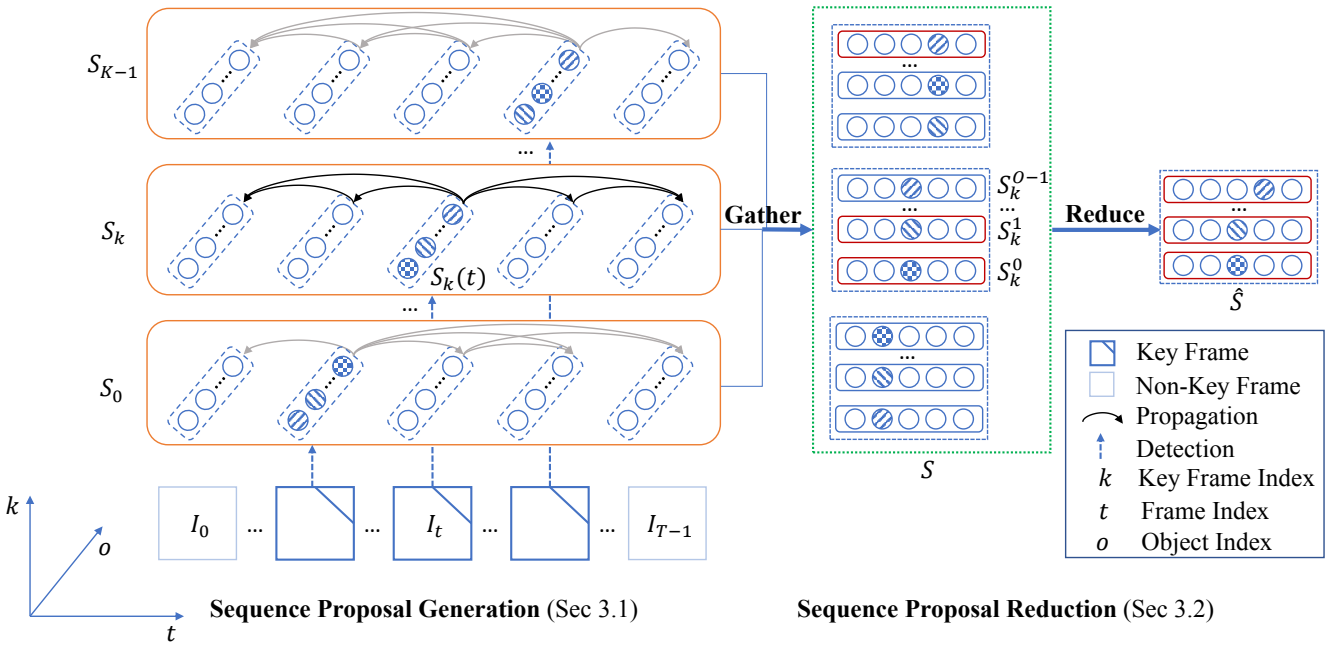

Figure 3. The **Propose-Reduce** paradigm consists of two stages. In the **Sequence Proposal Generation** stage, a sequence set S_k is generated by first detecting O instances at the k^{th} key frame. We assume frame t selected as the k^{th} key frame for convenience. Instance set $S_k(t)$ at frame t are then propagated to the whole video with *memory K-Propagation* (Sec. 3.2.2). $K \times O$ sequences $\{S_k^o\}$ are gathered to form a redundant set S, which is reduced to the final sequence set $\hat{S}$ in the **Sequence Proposal Reduction** stage. Different texture in circles differentiates among instances.

or spatial-temporal embedding [1]. Neighboring clips are merged with matching (*e.g.*, bipartite graph matching).

Semi-supervised Video Object Segmentation Semi-supervised video object segmentation (VOS) [33, 34] refers to the problem of segmenting specified objects in videos given the annotated first frame. Research [5, 32, 31, 50, 43, 20, 13, 26] was extensive. The most related methods to our framework are propagation-based, which propagate segmentation masks from the annotated first frame to the rest of the videos. Early research work [32, 17, 54, 50] propagates in a frame-by-frame pipeline, which is fragile and easily fails in distant frames due to occlusion and fast motion.

Recent memory-based method STM [31] resolves the problem in long-term propagation. Several methods [48, 55, 26, 39] were proposed to improve the performance of STM. In this paper, the propagation strategy of our Seq-Prop head is inspired by STM. Compared with STM that adopts two separate backbones for extracting features, we make the Seq-Prop head light-weighted by sharing the same feature backbone with other heads in Mask R-CNN.

Unsupervised Video Object Segmentation Compared with semi-supervised VOS, no annotated frame is given in unsupervised VOS (UVOS) [6]. UVOS can be regarded as a variant of VIS, while VIS segments objects with pre-defined classes. UVOS is to segment class-agnostic salient objects. Recent work detects salient objects [45, 27, 40, 56]. Besides, topological structures [42, 44] were proposed to obtain object segmentation in temporal domain. For example, RNN [42] structure or graph convolution network [44] can be utilized. Similar to VIS, detect-by-track [29] and clip-match based [1] methods can be applied to UVOS, which first generate instance segmentation on a single frame (or clip) and track (or match) objects across frames or clips. Our paradigm can also be applied with the classification head modified from multi-class classification to two-class one (*i.e.*, foreground and background).

3. Proposed Method

We propose the paradigm **Propose-Reduce** for the task of video instance segmentation. As shown in Fig. 3, the paradigm consists of two stages. Redundant sequence proposals are generated in the first stage (Sec. 3.1). We obtain instance segmentation on K selected key frames (Sec. 3.1.1). The segmentation results are then propagated to the whole video (Sec. 3.1.2) with our proposed **Seq Mask R-CNN** framework (Sec. 3.1.3). To reduce the redundancy in sequence proposals, in the second stage (Sec. 3.2), a sequence reduction method is applied to all sequences for final

Algorithm 1: Memory K-Propagation

Input: Video frames $\{I_t | \ t = 0, ..., T-1\}$, Key frames number K.
Output: Instance sequence proposal set S.
for $k = 0; k<K; k \leftarrow k+1$ **do**
  $t = g(k)$; // (Eq. 1)
  $S_k(t) \leftarrow \text{Detect}(I_t)$;
  /* Forward Direction */
  $\mathcal{M} \leftarrow \{S_k(t)\}$;
  for $i = t+1; i<T; i \leftarrow i+1$ **do**
    $S_k(i) \leftarrow \text{Propagate}(\mathcal{M}, I_i)$; // (Sec. 3.1.3)
    $\mathcal{M} \leftarrow \mathcal{M} \cup S_k(i)$;
  end
  /* Backward Direction */
  $\mathcal{M} \leftarrow \{S_k(t)\}$;
  for $j = t-1; j \geq 0; j \leftarrow j-1$ **do**
    $S_k(j) \leftarrow \text{Propagate}(\mathcal{M}, I_j)$; // (Sec. 3.1.3)
    $\mathcal{M} \leftarrow \mathcal{M} \cup S_k(j)$;
  end
  $S_k \leftarrow (S_k(0), S_k(1), ..., S_k(T-1))$;
  $k \leftarrow k+1$;
end
$S \leftarrow S_0 \cup S_1 \cup ... \cup S_{K-1}$; // Gather
return S;

sequence set as output.

3.1. Sequence Proposals Generation

3.1.1 Key Frames Selection

To generate sequence proposals, we first select K key frames to obtain their image-level instance segmentation masks. Specifically, for a T-frame video, the K key frames $\{I_{g(0)}, I_{g(1)}, ..., I_{g(K-1)}\}$ are selected at fixed intervals evenly, given by

$$g(k) = \max\{\lfloor T/K \rfloor, \ 1\} \times k, \ \ k = 0, ..., K-1. \quad (1)$$

The number of key frames plays an important role in our design. As described in Sec. 1, when selecting only one key frame, it degrades to paradigm (c) in Fig. 1, where the final results are highly dependent of the instance segmentation quality in the selected key frames. However, when many key frames are selected, the computational cost of detection and propagation would increase. We accordingly choose a small number of key frames in our experiments.

For each key frame, we generate its instance segmentation results by multiple heads (*i.e.*, bbox, classification and mask head) in Seq Mask R-CNN. For non-key frames, we only extract the backbone and FPN [21] features of these frames for the following propagation step. It saves computation.

3.1.2 Memory K-Propagation

The instance masks on K selected key frames are propagated bi-directionally to obtain K sets of mask sequences, *i.e.*, $\{S_0, S_1, ..., S_{K-1}\}$, as illustrated in Alg. 1. After all propagation finished, we gather K sequence sets from different key frames into one set $S = S_0 \cup S_1 \cup ... \cup S_{K-1}$.

As shown in Alg. 1, we maintain a memory $\mathcal{M}$ to alleviate error accumulation in long-term propagation [31]. It stores the encoded feature of previously segmented frames and propagates mask information to the current frame. The operations on $\mathcal{M}$ (*e.g.*, read and update) are similar to that of STM [31]. The difference is that the memory in [31] is utilized to propagate from the annotated first frame to the end of a video, while our work propagates the estimated mask from a key frame to the beginning and end of the video for K times.

Directly applying STM to our paradigm requires another two backbones to extract features for memory and query. Instead, we design an additional propagation module that can be seamlessly inserted into the image-level instance segmentation frameworks.

3.1.3 Sequence Mask R-CNN

We incorporate a propogation head (Seq-Prop head) on the top of Mask R-CNN for memory K-Propagation, which is called Sequence Mask R-CNN (Seq Mask R-CNN).

Architecture The architecture of Seq Mask R-CNN is shown in Fig. 4. It is based on outputting instance segmentation results for a single image, to which we add an extra propagation head that propagates instance masks to other frames.

Fig. 4 illustrates an example that takes two frames as input. We refer to them as the guidance frame (the t^{th} frame) and the query frame (the $(t+\delta)^{th}$ frame). For the guidance frame, we take the estimated mask M_g and largest FPN [21] feature P_2^g as input for encoding feature F_g. For a query frame, we take its P_2^q feature as input to obtain feature F_q.

With the two encoded feature maps, we utilize a non-local operation (NL) [46] to propagate mask information from the guidance frame to the query one and obtain the propagated feature $F_{g \to q}$. Finally, $F_{g \to q}$ and the largest backbone feature C_2^q from the query frame are utilized for decoding and generating the query mask M_q. The FPN feature P_2^q and the backbone feature C_2^q are utilized in encoding and decoding respectively, since they contain the richest semantic and detailed information on multiple instances.

Training In the training stage, we randomly select two frames as input for memory efficiency, *i.e.*, one guidance and one query frame. In one epoch, the query frame in a pair of frames is selected once per frame per video. The guidance frame is randomly sampled from the same video.

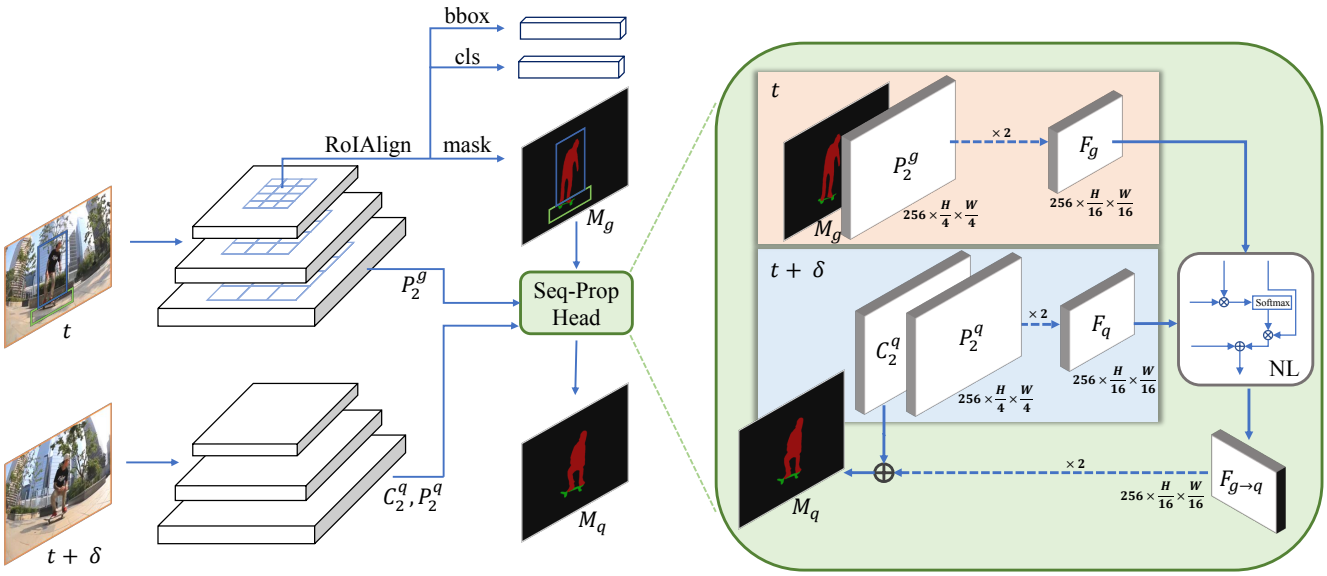

Figure 4. Framework of **Seq Mask R-CNN**. We adopt **Seq-Prop head** on Mask R-CNN for propagating instance masks from the guidance frame at time t to a query frame at time $t+\delta$. P_2^g, P_2^q are the largest FPN [21] feature of input images, and C_2^q is the largest backbone feature. **NL** is a non-local operation [46]. '$\times 2$' denotes 2 consecutive residual blocks. '$\otimes$' and '$\oplus$' denote matrix multiplication and summation, respectively. Detailed architectures are illustrated in the supplementary files.

Besides, in order to make the Seq-Prop head learn to propagate from the imperfect segmented masks, we utilize the estimated instance masks instead of the ground-truth ones as the guidance input for training. It makes the head more robust at the inference stage.

To train our overall framework, we adopt a multi-task loss $\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask} + \mathcal{L}_{prop}$. The classification loss $\mathcal{L}_{cls}$, bounding-box loss $\mathcal{L}_{box}$, and mask loss $\mathcal{L}_{mask}$ are the same as those in Mask R-CNN [14]. As for the propagation loss $\mathcal{L}_{prop}$ utilized to train Seq-Prop head, we adopt a scale-balanced soft IoU loss [20], since the Seq-Prop head propagates multi-scale instances masks at the same time.

Inference During the stage of inference, the guidance frame input is replaced by a memory pool (illustrated in Alg. 1), which stores the encoded features from the frames that have been propagated. Specifically, for each iteration of propagation, memory is updated by appending the encoded feature of the current frame, which increases model's robustness to occluded instances [31].

By sharing backbone features with the other three heads in Seq Mask R-CNN, our propagation head discards two heavy encoders for memory and query in STM [31].

3.2. Sequence Proposals Reduction

Redundant sequence proposals exist after the first stage where sequences of the same instance may be generated for multiple times from different key frames. To reduce redundancy, inspired by NMS [12, 11, 37, 38] that is widely used in image-level instance segmentation in post-processing, we adopt a variant of NMS for sequences reduction. To apply it to sequences, three key elements in NMS need to be defined, i.e., the input sequence sets, sequence score and sequences IoU.

Input Sequence Set In the stage of sequence proposal generation, we obtain K sets of sequence proposals $\{S_0, S_1, ..., S_{K-1}\}$ gathered as S. For each sequence set S_k, we have its corresponding mask $M(S_k)$ and classification score $C(S_k)$. We set the maximum instance number in a key frame as O. Then S_k can be represented as a set of instance sequences $\{S_k^o\}$, where $o \in [0, O\text{-}1]$. Correspondingly, their masks and scores are defined as the set of $\{M(S_k^o)\}$ and $\{C(S_k^o)\}$, where $M(S_k^o) \in \{0,1\}^{T\times H\times W}$ and the score of each sequence $C(S_k^o)$ is defined later.

Accordingly, we obtain the input sequence set as $S = \{S_k^o\}$, where $k \in [0, K\text{-}1], o \in [0, O\text{-}1]$, which consists of a maximum of $K \times O$ instance sequences. Since the instance number per key frame is less than O in most cases, many sequences in S are empty and the sequence number is much less than $K \times O$. Our target is to reduce the redundant sequence set S into a final sequence set $\widehat{S}$.

Sequence Score The score for each instance sequence $C(S_k^o)$ reflects its priority to be selected. To represent the priority of an instance sequence, we consider all frames in this sequence. For each instance, on any frame I_t, we obtain its classification score $C(S_k^o(t)) \in [0,1]^{|\mathbb{C}|}$ from the classification head of Seq Mask R-CNN, where $|\mathbb{C}|$ indicates the number of instance classes. We average the scores among all frames and take the max score among $|\mathbb{C}|$ classes

as the score of this instance sequence. The score for each sequence $C(S_k^o)$ is defined as

$$C(S_k^o) = \max_{|\mathbb{C}|} \frac{1}{T} \sum_{t=0}^{T-1} C(S_k^o(t)) \ . \tag{2}$$

Sequences IoU Intersection-over-union (IoU) between two sequences measures their overlap. We calculate the mask IoU instead of the bounding-box IoU to measure the overlap more precisely. We denote the masks of two sequences as $M(S_k^o)$ and $M(S_{\tilde{k}}^{\tilde{o}})$, where $M(S_k^o)$ indicates the mask sequence of the o^{th} instance from the k^{th} key frame, and $M(S_{\tilde{k}}^{\tilde{o}})$ is similarly defined. Then the IoU between two sequences, *i.e.*, $IoU(S_k^o$ and $S_{\tilde{k}}^{\tilde{o}})$, is computed as

$$IoU(S_k^o, S_{\tilde{k}}^{\tilde{o}}) = \frac{\sum_{t=0}^{T-1} |M(S_k^o(t)) \cap M(S_{\tilde{k}}^{\tilde{o}}(t))|}{\sum_{t=0}^{T-1} |M(S_k^o(t)) \cup M(S_{\tilde{k}}^{\tilde{o}}(t))|} \ , \tag{3}$$

where $M(S_k^o(t))$ and $M(S_{\tilde{k}}^{\tilde{o}}(t))$ are the masks of the t^{th} frame from the two sequences S_k^o and $S_{\tilde{k}}^{\tilde{o}}$, respectively.

With the defined sequence set, sequence score and sequences IoU, we directly apply the traditional NMS algorithm to the sequence set to reduce the redundant sequences. More details of this algorithm is included in our supplementary files. The sequence set $\widehat{S}$ after NMS is our final result for the task of VIS.

4. Experiments

4.1. Datasets

YouTube-VIS [53] YouTube-VIS dataset is currently the largest dataset for video instance segmentation task. It contains 2,238 training videos and 302 validation videos, with 40 categories involved. Validation scores are evaluated on online benchmark. Similar to image instance segmentation [22] , the benchmark adopts Average Precision ($\mathcal{AP}$) and Average Recall ($\mathcal{AR}$) metrics to evaluate the sequence accuracy averaged over the category set.

DAVIS-UVOS [6] DAVIS-UVOS dataset is proposed for unsupervised video object segmentation for salient generic objects. It contains 60 training videos and 30 validation videos with high-quality annotations. This task can be viewed as a special case of video instance segmentation with 2 categories (foreground and background). In the evaluation stage, it considers no more than 20 predicted sequences in a video and measures the average between $\mathcal{J}$ scores (the mean IoU between the estimated mask and ground-truth) and $\mathcal{F}$ scores (the F-measure of the estimated mask boundaries) via bipartite graph matching.

4.2. Implementation Details

The training data in the above datasets is not sufficient, resulting in over-fitting. To alleviate this issue, we adopt 80K training images in the COCO [22] dataset (image instance segmentation) for compensation (also adopted in [1]). For each image in COCO, we augment it with $\pm 30^\circ$ rotation to generate a three-frame pseudo video. For the training on YouTube-VIS, we only select images with overlapping categories in COCO. For DAVIS-UVOS, we select all images from COCO and treat all annotated instances as one category, *i.e.*, foreground.

Our training consists of two stages, *i.e.*, **main-training stage** and **finetuning stage**. In the main-training stage, we first train our model on the mixed dataset including COCO and the video dataset (*i.e.*, YouTube-VIS, DAVIS-UVOS) for 4 epochs with 640×320 input size. In the finetuning stage, for YouTube-VIS dataset, the model is trained on this dataset with the same input size, while the model finetuned on DAVIS-UVOS dataset takes 854×480 size as input. We finetune the models for 5 epochs for both datasets.

All models are trained with 6 NVIDIA Titan X GPUs, implemented by PyTorch. The training time takes about 2-4 days for each dataset. We set K as 6 for YouTube-VIS and 4 for DAVIS-UNVOS (see Sec. 4.4). More details are included in our supplementary files.

4.3. Main Results

YouTube-VIS The quantitative results on YouTube-VIS are included in Table 1. We list the backbone [15, 52, 3] used in different methods for fair comparison. MaskProp, the SOTA method, takes a strong backbone (*i.e.*, STSN [3]-ResNeXt-101) to extract spatial-temporal features, and a stronger detection head (*i.e.*, HTC [8]) to refine detection results iteratively. In contrast, our best model only uses ResNeXt-101 to extracting spatial representation features and the vanilla head in Mask R-CNN for detection. Our model already outperforms MaskProp by 1% in terms of $\mathcal{AP}$ and 3.4% in terms of $\mathcal{AR}$@10.

The large improvement of recall stems from the sampling strategy on multiple key frames. Note that MaskProp adopts a post-process that refines masks to gain 1.9% improvement, while Seq Mask R-CNN does not adopt this post-process. The previous best method that only extracts spatial features is EnsembleVIS, which combines four separate networks into a complex system, including detection [14], classification [52], re-identification [29], and segmentation [9]. Our *single-model* method surpasses EnsembleVIS by 2.8% in $\mathcal{AP}$ and 4.3% in $\mathcal{AR}$@10.

DAVIS-UVOS We also evaluate our approach on DAVIS-UVOS dataset, as shown in Table 2. The SOTA method UnOVOST combines multiple models (*e.g.*, Mask R-CNN [14], PWC-Net [41] and ReID Net [49]) into a complex system. Our *single-model* method with ResNet-101 backbone achieves a comparable performance. With a stronger backbone (*i.e.*, ResNeXt-101), our method outperforms UnOVOST in both $\mathcal{J}$ and $\mathcal{F}$ scores. Compared with SOTA

Paradigm	Method	Backbone	HR-Ref	$\mathcal{AP}$	$\mathcal{AP}$@50	$\mathcal{AP}$@75	$\mathcal{AR}$@1	$\mathcal{AR}$@10
Track-by-Detect	MaskTrack [53]	ResNet-50		30.3	51.1	32.6	31.0	35.5
	SipMask [7]	ResNet-50		33.7	54.1	35.8	35.4	40.1
	EnsembleVIS [28]	ResNeXt-101*	✓	44.8	-	48.9	42.7	51.7
Clip-Match	STEm-Seg [1]	ResNet-50		30.6	50.7	33.5	31.6	37.1
	STEm-Seg [1]	ResNet-101		34.6	55.8	37.9	34.4	41.6
	MaskProp [2]	ResNeXt-101	✓	44.3	-	48.3	-	-
	MaskProp [2]	STSN [3]-ResNeXt-101		44.7	-	-	-	-
	MaskProp [2]	STSN [3]-ResNeXt-101	✓	46.6	-	51.2	44.0	52.6
Propose-Reduce	Ours	ResNet-50		40.4	63.0	43.8	41.1	49.7
	Ours	ResNet-101		43.8	65.5	47.4	43.0	53.2
	Ours	ResNeXt-101		**47.6**	**71.6**	**51.8**	**46.3**	**56.0**

Table 1. Quantitative results of video instance segmentation in YouTube-VIS validation set. 'HR-Ref' indicates post-processing that resizes cropped masks to a large resolution and refines details with extra convolutional layers. *: EnsembleVIS adopts multiple models and their largest backbone is ResNeXt-101.

Method	Backbone	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$-Mean	$\mathcal{F}$-Mean
RVOS [42]	ResNet-101	41.2	36.8	45.7
STEm-Seg [1]	ResNet-101	64.7	61.5	67.8
UnOVOST [29]	ResNet-101*	67.9	66.4	69.3
Ours	ResNet-101	68.3	65.0	71.6
Ours	ResNeXt-101	**70.4**	**67.0**	**73.8**

Table 2. Quantitative results of unsupervised video object segmentation on DAVIS-UVOS validation set. *: UnOVOST combines multiple models and their largest backbone is ResNet-101.

single-model method STEm-Seg, our method with the same backbone (*i.e.*, ResNet-101) surpasses it by 3.6% in $\mathcal{J}\&\mathcal{F}$.

Visualization We further present the comparison with previous paradigms ([7, 1]) in long-term occlusion scenarios, as shown in Fig. 5. Salient objects (bear/surfboard) are occluded by trees/waves in multiple frames. Track-by-Detect [7] fails to re-identify the same instance with distorted appearance. Clip-Match [1] treats them as two instances due to being out of the matching scopes. In contrast, our paradigm generates complete sequences via long-term propagation. More visual results are shown in Fig. 6. Our method fails to propagate masks with highly consistently-occluded instances of the same category (*i.e.*, person).

4.4. Ablation Experiments

All the ablation experiments are conducted with the ResNeXt-101 [52] backbone on YouTube-VIS and DAVIS-UVOS validation sets.

Training Stage We conduct experiments to study the effect of different training stages (Sec. 4.2), as shown in Table 3. With the finetuning stage only, the large drop of performance indicates that insufficient video data leads to over-fitting. Adopting the main-training stage only alleviates over-fitting. It is still hard to reach the performance by

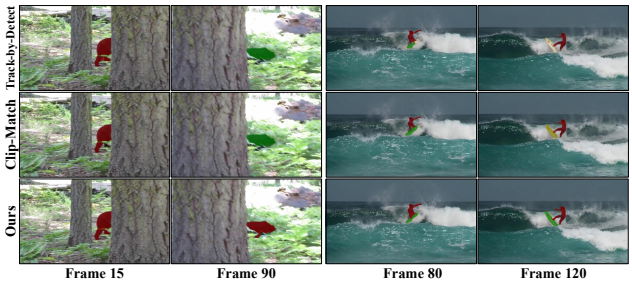

Figure 5. Visual comparison of different paradigms on challenging scenarios. Frames are sampled before and after occlusion.

Variants	YouTube-VIS	DAVIS-UVOS
Main-training only	46.2	67.3
Finetuning only	40.8	48.9
Both	**47.6**	**70.4**

Table 3. Training data analysis on YouTube-VIS and DAVIS-UVOS validation set. 'Both' denotes two-stage training, including main-training and finetuning. We report $\mathcal{AP}$ for YouTube-VIS and $\mathcal{J}\&\mathcal{F}$ for DAVIS-UVOS.

two-stage training, since there exist a domain gap between image (*i.e.*, COCO) and video datasets (*i.e.*, YouTube-VIS, DAVIS-UVOS).

Sequence Reduction Tab. 4 reports the ablation results with and without sequence reduction. Its effects on YouTube-VIS and DAVIS-UVOS are different for their evaluation metric. The evaluation metric in YouTube-VIS is sensitive to false positive. Sequence reduction significantly increases $\mathcal{AP}$ at the cost of a slight decrease of $\mathcal{AR}$. For DAVIS that does not penalize false positives, sequence reduction stably increases all the metrics.

Category-Aware Reduction In the evaluation of category-aware metrics (*e.g.*, $\mathcal{AP}$), new redundancy ap-

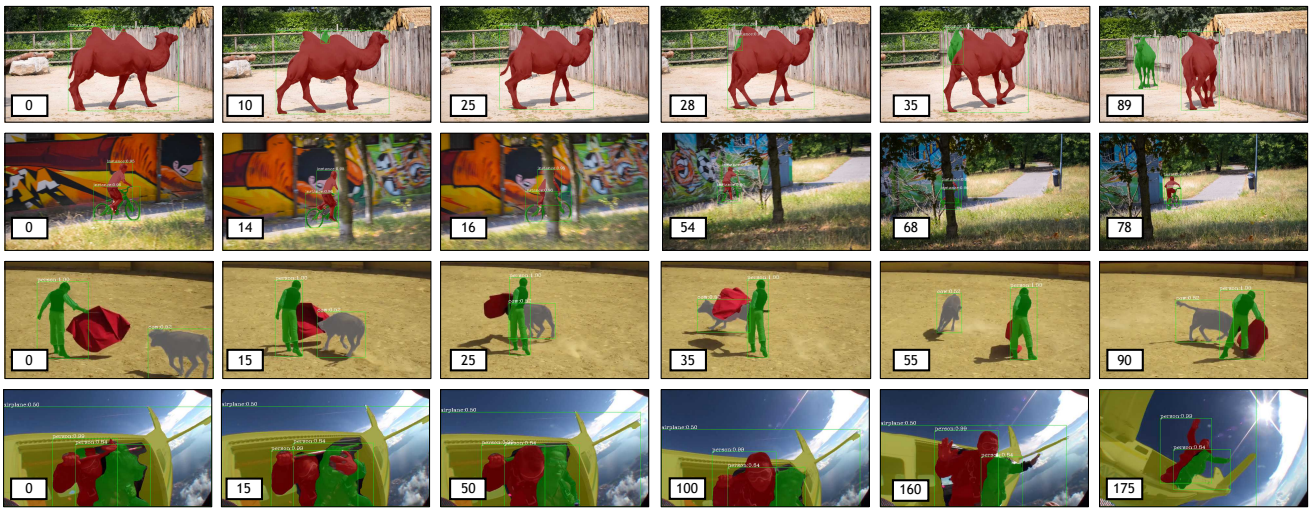

Figure 6. Visual results on DAVIS-UVOS and YouTube-VOS. Frames are sampled at challenging moments (*e.g.*, fast motion). We also show in the last row a failure case of overlapped same-category instances, where the arm of one occluded person is segmented to the other. Category 'Instance' in DAVIS-UVOS denotes the salient generic object.

Backbone	Seq. Reduce	YouTube $\mathcal{AP}$	YouTube $\mathcal{AR}$@100	DAVIS $\mathcal{J}$	DAVIS $\mathcal{F}$
ResNet-101		19.3	**55.1**	62.4	69.5
	✓	**43.8**	53.2	**65.0**	**71.6**
ResNeXt-101		20.7	**58.1**	64.9	70.9
	✓	**47.6**	56.0	**67.0**	**73.8**

Table 4. Sequence reduction analysis on YouTube-VIS and DAVIS-UVOS validation set.

CA. Reduce	ResNet-50	ResNet-101	ResNeXt-101
	40.4	43.8	47.6
✓	**41.5**	**45.1**	**48.3**

Table 5. Ablations of Category-Aware Reduction (CA. Reduce) on different backbones. We reports $\mathcal{AP}$ for YouTube-VIS.

pears after the category assignment. Sequences assigned to the same category conflict with each other in the final evaluation. These redundant sequences can be filtered by applying the same sequence reduction techniques (Sec. 3.2) for each category. Ablation results on Tab. 5 demonstrate that such a category-aware reduction post-processing stably improves the accuracy on different backbones.

Key Frames In our inference paradigm, the number of key frames (hyper-parameter K) plays an important role in controlling the trade-off between accuracy and efficiency. As shown in Fig. 7, our model performs poorly when $K = 1$ since one sequence proposal is sensitive to segmentation quality at the key frame (see also Fig. 2) and may miss some instances. When sampling more key frames, accuracy increases dramatically as sufficient sampling improves robustness and recall for the same instance. With more key frames sampled, accuracy fluctuates, probably because the quality of instance sequences varies among different key

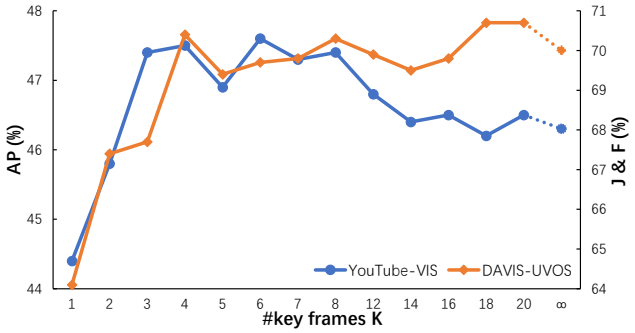

Figure 7. Key frame analysis on YouTube-VIS and DAVIS-UVOS validation set. '#key frames' denotes the number of selected key frames in a video, where $K = \infty$ means all frames are key frames.

frames and the estimated sequence score (Eq. (2)) in NMS cannot effectively reflect the priority of the sequences.

Note that the accuracy in YouTube-VIS gradually drops as $K \geq 8$, since its evaluation metric (Sec. 4.1) is sensitive to false positives from residual redundant sequences after NMS. We set K to 6 and 4 for YouTube-VIS and DAVIS-UVOS by default.

5. Conclusion

In this paper, we have proposed a new paradigm to tackle the task of video instance segmentation, which requires no tracking/matching part to avoid error accumulation and ensures high recall. Following the paradigm, we design our framework named Seq Mask R-CNN, which incorporates a newly-designed propagation head on Mask R-CNN. The extensive experiments verify the effectiveness of our framework. Besides, this work provides a new perspective on the VIS task, which motivates future work on extending image-level methods to video domain.

References

[1] Ali Athar, Sabarinath Mahadevan, Aljoša Ošep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020. 1, 2, 3, 6, 7

[2] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020. 1, 2, 7

[3] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks. In *ECCV*, 2018. 6, 7

[4] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 2

[5] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 3

[6] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv:1905.00737*, 2019. 2, 3, 6

[7] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020. 1, 2, 7

[8] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019. 6

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 6

[10] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *ICCV*, 2019. 2

[11] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 5

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 5

[13] Bhat Goutam, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *ECCV*, 2020. 3

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 5, 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[16] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019. 2

[17] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for multiple object tracking. *arXiv:1703.09554*, 2017. 3

[18] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020. 2

[19] Chung-Ching Lin, Ying Hung, Rogerio Feris, and Linglin He. Video instance segmentation tracking with a modified vae architecture. In *CVPR*, 2020. 2

[20] Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Agss-vos: Attention guided single-shot video object segmentation. In *ICCV*, 2019. 3, 5

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 4, 5

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[23] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *ICCV*, 2017. 2

[24] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 2

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[26] Xinkai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020. 3

[27] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. Learning video object segmentation from unlabeled videos. In *CVPR*, 2020. 3

[28] Jonathon Luiten, Philip Torr, and Bastian Leibe. Video instance segmentation 2019: A winning approach for combined detection, segmentation, classification and tracking. In *ICCVW*, 2019. 7

[29] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. Unovost: Unsupervised offline video object segmentation and tracking. In *WACV*, 2020. 1, 2, 3, 6, 7

[30] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NeurIPS*, 2017. 2

[31] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 3, 4, 5

[32] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 3

[33] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 3

[34] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 3

[35] Lu Qi, Xiangyu Zhang, Yingcong Chen, Yukang Chen, Jian Sun, and Jiaya Jia. Pointins: Point-based instance segmentation. *arXiv preprint arXiv:2003.06148*, 2020. 2

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2, 5

[38] Rasmus Rothe, Matthieu Guillaumin, and Luc Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *ACCV*, 2014. 5

[39] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020. 3

[40] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018. 3

[41] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 6

[42] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, 2019. 3, 7

[43] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 3

[44] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, 2019. 3

[45] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven CH Hoi, and Haibin Ling. Learning unsupervised video object segmentation through visual attention. In *CVPR*, 2019. 3

[46] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 4, 5

[47] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020. 2

[48] Ruizheng Wu, Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Memory selection network for video propagation. In *ECCV*, 2020. 3

[49] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 2019. 6

[50] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 3

[51] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 2020. 2

[52] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 6, 7

[53] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1, 2, 6, 7

[54] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 3

[55] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *CVPR*, 2020. 3

[56] Tianfei Zhou, Jianwu Li, Shunzhou Wang, Ran Tao, and Jianbing Shen. Matnet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE TIP*, 2020. 3