Pyramid Constrained Self-Attention Network for Fast Video Salient Object Detection

Yu-Chao Gu[1*], Li-Juan Wang[1*], Zi-Qin Wang[2], Yun Liu[1], Ming-Ming Cheng[1†], Shao-Ping Lu[1]

[1]TKLNDST, CS, Nankai University

[2]The University of Sydney

{ycgu,wlj,nk12csly}@mail.nankai.edu.cn

slu@nankai.edu.cn

ziqin.wang.edu@gmail.com

Abstract

Spatiotemporal information is essential for video salient object detection (VSOD) due to the highly attractive object motion for human's attention. Previous VSOD methods usually use Long Short-Term Memory (LSTM) or 3D ConvNet (C3D), which can only encode motion information through step-by-step propagation in the temporal domain. Recently, the non-local mechanism is proposed to capture long-range dependencies directly. However, it is not straightforward to apply the non-local mechanism into VSOD, because i) it fails to capture motion cues and tends to learn motion-independent global contexts; ii) its computation and memory costs are prohibitive for video dense prediction tasks such as VSOD. To address the above problems, we design a *Constrained Self-Attention* (CSA) operation to capture motion cues, based on the prior that objects always move in a continuous trajectory. We group a set of CSA operations in *Pyramid* structures (PCSA) to capture objects at various scales and speeds. Extensive experimental results demonstrate that our method outperforms previous state-of-the-art methods in both accuracy and speed (110 FPS on a single Titan Xp) on five challenge datasets. Our code is available https://github.com/guyuchao/PyramidCSA.

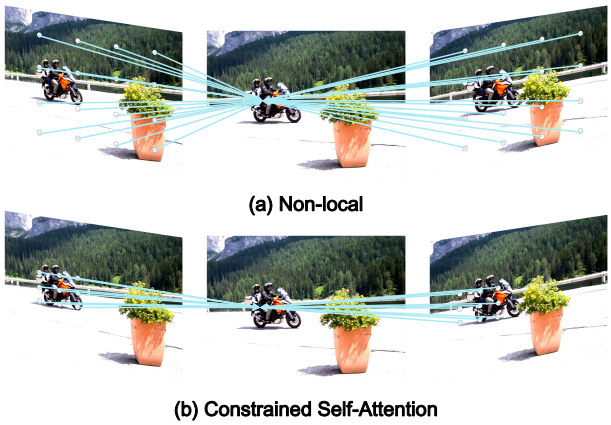


Figure 1: The reference area of the non-local operation and the CSA. (a) The non-local measures the pair-wise relation on all space-time position. (b) The CSA measures the pair-wise relation in the neighbor area of query position in consecutive frames.

Introduction

Video Salient Object Detection (VSOD) aims at locating the most attractive object in video sequences. It usually serves as a pre-processing step for many real-time applications, such as video tracking (Wu, Li, and Luo 2014), video segmentation (Wang, Shen, and Porikli 2015) and human-computer interaction (Xu et al. 2016). Therefore, both efficiency and accuracy are important for the VSOD model design.

Since object motion is highly attractive to human's attention, spatiotemporal information is essential for VSOD. Previous state-of-the-art VSOD approaches (Le and Sugimoto 2018; Li et al. 2018a; Song et al. 2018; Fan et al. 2019) mainly rely on some traditional techniques, including 3D ConvNet (C3D) (Tran et al. 2015), ConvLSTM (Xingjian et al. 2015), and optical flow (Dosovitskiy et al. 2015), to capture the temporal information. However, these traditional techniques can only process adjacent areas at a specific time in the temporal domain, it is difficult for them to capture long-range temporal information directly.

The recently proposed non-local neural network (Wang et al. 2018) generalizes self-attention mechanism (Vaswani et al. 2017) in machine translation. It can model long-range temporal dependencies in video classification by learning pairwise relationships among feature elements of different frames. However, in VSOD, we find that the non-local operation would focus on global contexts rather than motion cues. Since the non-local is a distance-independent operation in space and time, it tends to learn global contexts that fit all queries.

In order to model motion dependencies in the video segment, we design an alternative to non-local, named *Constrained Self-Attention* (CSA). CSA is based on the motion prior that objects always move in a continuous trajectory. As illustrated in Fig. 1, instead of learning dense pairwise relationships globally, CSA learns relations in a neighbor

*Both authors contributed equally to this work.

†Corresponding author: cmm@nankai.edu.cn.

(a) Query (b) CSA#1 (c) CSA#2 (d) NL#1 (e) NL#2

Figure 2: Visualization of the attention map produced by the non-local operation and the CSA. We query two positions (#1, #2) in frame T and get attentions of frames T and $T + 5$. The non-local module produces motion-independent attention because it outputs similar attention maps for different query positions. The similar observation that the non-local module outputs query-independent attention in object detection task was made in (Cao et al. 2019). The CSA is constrained by motion prior, which focuses on motion cues.

area around the query position across several frames. CSA is forced to focus on local motion patterns instead of learning global contexts. Besides, CSA significantly reduces the overhead of memory and computation of non-local. Considering the salient object can be various scales and move at different speeds, we propose to group a set of CSA operations in a *Pyramid* structure (PCSA). There are several merits of applying PCSA to VSOD:

- Instead of previous methods' encoding temporal information progressively, PCSA can capture temporal information in the video segment directly.
- PCSA can capture motion information of multiscale objects at various speeds.
- PCSA allows for lower memory and computation overhead to capture temporal information.

We carry experiments on six challenge VSOD datasets and achieve new state-of-the-art results. Our model can reach 110 FPS at a single Titan Xp GPU. Experiments show our PCSA outperforms the non-local in capturing motion information, with only 0.5% FLOPs and 2% memory consumption of non-local. Furthermore, we demonstrate our model on unsupervised video object segmentation (UVOS) task. Without CRF post-processing, our model is the first real-time method which achieves comparable performance to previous state-of-the-art methods.

Related Work

VSOD Architecture

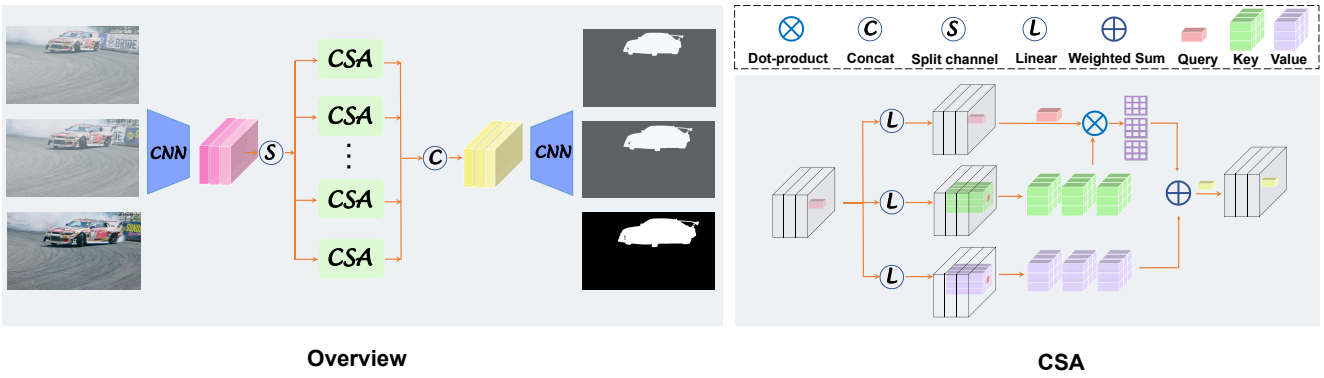
Conventional VSOD models (Xi et al. 2016; Liu et al. 2016; Tu et al. 2016; Zhang et al. 2015) usually exploit handcrafted features. With the success of deep learning, VSOD models use deep neural network to extract salient features. Then, several mechanisms are explored to extend static salient detection to VSOD. (Le and Sugimoto 2018) use C3D to extract features and then construct a spatiotemporal graph to ensure time coherence. (Li et al. 2018a; Tang et al. 2018; Chen et al. 2018) utilize optical flow to exploit motion information. The computation cost of C3D and optical flow is usually expensive. Recent state-of-the-art methods (Song et al. 2018; Fan et al. 2019) use the ConvLSTM structure to propagate temporal information progressively. Comparing to previous works, our PCSA can directly capture motion cues in a video segment, which is more efficient.

Self-Attention

Self-Attention (Vaswani et al. 2017) is proposed to model long-range dependencies in machine translation. It works by measuring the pair-wise relationships of all feature elements and aggregating information based on the relationships. Non-local Neural Network (Wang et al. 2018) proposes a more general representation of self-attention mechanism. It can exploit long-range information in video classification. Following the non-local operation, several works are presented to exploit the relationships of feature elements and reduce the computation cost. (Yue et al. 2018) learn explicit correlations of feature elements in both space-time and channels. They study a compact representation of kernel functions to reduce the cost of the non-local operation. In dense prediction tasks, for example, semantic segmentation, directly applying the non-local to measure the relationships of all feature elements is impractical due to the high resolution of feature maps. (Huang et al. 2018) propose a criss-cross method to learn sparse relationships, which significantly reduce the computation and memory cost of non-local. To the best of our knowledge, there is no previous work focusing on applying self-attention mechanism to capture motion cues in VSOD. Our method bridges this gap and shows improvement over non-local, both in efficiency and accuracy for capturing motion cues.

Video Object Segmentation

Video Object Segmentation (VOS) (Perazzi et al. 2016) is related to VSOD, and it mainly includes Unsupervised VOS (UVOS) (Song et al. 2018) and semi-supervised VOS (Wang et al. 2019b). Semi-supervised VOS aims to segment specific objects which are assigned by the first frame, while UVOS predicts masks for primary objects in a video, with no other hints such as reference masks. UVOS is a typical application of VSOD, since the primary object in a video can be detected by VSOD. A significant difference between UVOS



Overview CSA

Figure 3: Illustration of proposed Pyramid Constrained Self-Attention (PCSA) network. A video segment with T frames ($T = 3$ for example) is fed into CNN encoder to extract static salient features. The static features are split into g groups with C/g channels ($g = 4$ for example), where C is the channel numbers of static features. We use g parallel CSA with different window sizes and dilations to capture motion information.

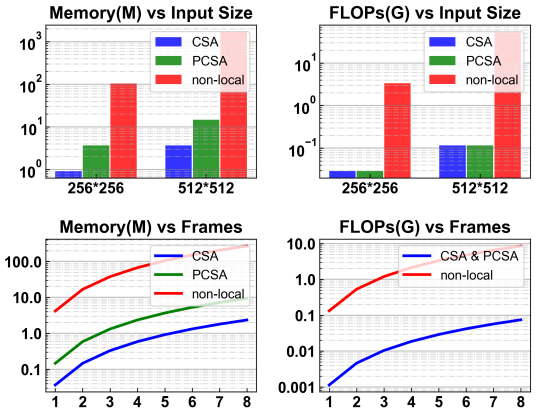


Figure 4: Memory and FLOPs vs frames and input size of our method and the non-local. We set default $H = W = 256$, $T = 5$ and $C = 32$. Then we study the influence of different input sizes and frames to the non-local and our proposed method. Notably, the PCSA has the same FLOPs with the CSA operation.

and VSOD is that the prediction of VSOD models is a probability saliency map while UVOS models output a binary segmentation. In this work, we show our method achieves the best accuracy-efficiency trade-off in UVOS task.

Proposed Method

Review of the Non-local Operation

We first revisit the non-local operation (Wang et al. 2018) and discuss the problems when applying it to VSOD.

Revisit. Suppose a video segment with T frames is fed into the encoder, and we obtain the extracted features $\mathbf{X} \in \mathbb{R}^{T\times H\times W\times C}$ as the input of the non-local operation, where C, H, W denote the channels, height and width of the feature map, respectively. The general non-local operation has three steps, i.e. linear embedding, affinity measuring, and context aggregating.

In the linear embedding step, (Wang et al. 2018) suggest using $1\times1\times1$ convolution $\theta(\cdot), \phi(\cdot), g(\cdot)$ as linear function to get the embedding features:

$$\mathbf{Q} = \theta(\mathbf{X}), \mathbf{K} = \phi(\mathbf{X}), \mathbf{V} = g(\mathbf{X}). \tag{1}$$

Here, $\mathbf{Q} \in \mathbb{R}^{THW\times C}$, $\mathbf{K} \in \mathbb{R}^{THW\times C}$ and $\mathbf{V} \in \mathbb{R}^{THW\times C}$. In the affinity measuring step, the non-local operation adopts pairwise function f to compute the affinity between the feature elements in $\mathbf{Q}$ and $\mathbf{K}$. (Wang et al. 2018) show that different choices of f have similar performance, thus we use the dot-product as f. Affinity measuring step can be formulated as

$$\mathbf{W}_{att} = f(\mathbf{Q}, \mathbf{K}) = \mathbf{Q}\mathbf{K}^{\mathrm{T}}. \tag{2}$$

Here, $\mathbf{W}_{att} \in \mathbb{R}^{THW\times THW}$ encodes the pair-wise affinity between all space-time positions in the feature map. In the context aggregating step, the resulting feature $\mathbf{Y} \in \mathbb{R}^{THW\times C}$ can be viewed as a weighted sum of the embedding feature $\mathbf{V}$ with weight $\mathbf{W}_{att}$:

$$\mathbf{Y} = \mathbf{W}_{att}\mathbf{V}. \tag{3}$$

In order to incorporate the non-local operation into any pre-trained networks, the common practice is to use a residual connection:

$$\mathbf{Z} = \mathbf{X} + \beta\mathbf{Y}. \tag{4}$$

Here, β is a scale parameter which initializes to zero.

Discussion. Non-local operation is powerful in video classification task. However, it cannot be applied to VSOD directly for several reasons.

Firstly, we find that the non-local operation tends to capture global contexts instead of motion cues in VSOD. As shown in Fig. 2, the non-local operation learns similar attention map for different querying positions. (Cao et al. 2019) has a similar observation that the non-local operation learns a query-independent attention in the object detection task.

Different from the object detection task, global contexts are not suitable for predicting salient objects in a video, because object motion is more attractive to human's attention (Itti, Koch, and Niebur 1998). Based on the prior that objects always move in a continuous trajectory, we can constrain the affinity-measuring and context-aggregating region in consecutive frames to the neighbor area of querying position. The constrained self-attention aggregates information from sparse positions, preventing from learning the motion-independent responses to fit all queries. We will make a further illustration for this point in the next subsection.

Secondly, the computation and memory costs of the non-local operation are expensive. Because the non-local operation needs to compute pair-wise affinity between all feature elements, the memory usage and the FLOPs for the non-local are quadratic function with respect to the frames T and the feature resolution $H \times W$. Different from video classification task, which is unnecessary to preserve spatial resolution, VSOD usually outputs high resolution feature map, thus leading to prohibitive computation and memory overhead. Fig. 4 shows the overhead comparison between the non-local operation and our PCSA.

Pyramid Constrained Self-Attention

In this subsection, we first introduce the *Constrained Self-Attention* (CSA) operation, which can capture motion cues effectively. Then we present a *Pyramid* architecture to group several CSA (PCSA) to handle multi-scale objects at various speeds in the video.

Constrained Self-Attention. In order to capture motion cues in a video segment with frames T, we first get static salient features of size $T \times H \times W$ from backbone. Then we use three linear functions to project feature into three subspaces, i.e. query $\mathbf{Q}$, key $\mathbf{K}$ and value $\mathbf{V}$ space. As shown in Fig. 3, the object in the first frame shares similar position with that in adjacent frames. Based on this prior, when we query a feature element $x_q = (t, h, w)$ in $\mathbf{Q}$, surrounding area of x_q in feature $\mathbf{K}$ is used to measure affinity. By introducing a response window with radius r and dilation d, the surrounding area $S(x_q, \mathbf{K})$ of x_q can be formulated as

$$S(x_q, \mathbf{K}) = \left\{ \mathbf{K}_{(t',h',w')} \right\}_{t'=1,h'=h-dr,w'=w-dr}^{T,h+dr,w+dr}, \quad (5)$$

where $S(x_q, \mathbf{K}) \in \mathbb{R}^{TR^2 \times C}$. The affinity function Eq. 2 can be reformulated as

$$\mathbf{W}_{att} = f(\mathbf{Q}, \mathbf{K}) = \mathbf{Q} S(\mathbf{Q}, \mathbf{K})^{\mathrm{T}}. \quad (6)$$

Softmax function is used to normalize the attention weight $\mathbf{W}_{att}$. Then we augment the static salient feature through aggregating feature in all given frames weighted by attention $\mathbf{W}_{att}$. The aggregating step in Eq. 3 can be rewritten as

$$\mathbf{Y} = \mathbf{W}_{att} S(\mathbf{Q}, \mathbf{V}). \quad (7)$$

The attention visualization of our CSA can be found in Fig. 2. The non-local learns a global context, while we learn affinity-based motion cues by constraining the response area. Comparing to ConvLSTM, which needs step-by-step propagating temporal information, the CSA can directly access information in multiple frames. The CSA operation can be plugged into pretrained static SOD backbone through a residual connection:

$$\mathbf{Z} = \mathbf{X} + \mathbf{Y}. \quad (8)$$

Comparing to the non-local operation, our computation and memory overhead is quadratic function of frames T, but linear function of feature size $H \times W$. Comparison results can be found in Fig. 4. The CSA is more efficient than the non-local operation. We further demonstrate the effectiveness of the CSA in the following experiments.

Pyramid Combination. The CSA mentioned above can capture local motion patterns by adopting a motion prior. But the dynamic scene is rather complicated. Multi-scale objects move at various speeds. Single window size in the CSA cannot adapt various motions. Inspired by the multi-head mechanism in machine translation (Vaswani et al. 2017), we project feature into different subspaces with different learned linear functions. Different from multi-head in (Vaswani et al. 2017), we combine multi-head and multi-scale learning. Specifically, we first split the input feature $\mathbf{X}$ into g groups $\{\mathbf{X}_i, i = 1, 2, ...g\}$ along channel. Then we use several parallel CSA $\{CSA_i, i = 1, 2, ...g\}$ with different window sizes $r = (r_1, r_2, ...r_g)$ and dilations $d = (d_1, d_2, ...d_g)$ to extract motion cues from different $\mathbf{X}_i$. The result features $\{\mathbf{Y}_i, i = 1, 2, ...g\}$ are concatenated, then we perform a linear combination with $1 \times 1 \times 1$ convolution to get spatiotemporal feature $\mathbf{Y}$. The feature $\mathbf{Y}$ is added to input feature $\mathbf{X}$ through residual connection in Eq. 8.

The single CSA with the fixed window size R has limitation of lossing moving target caused by various speeds and scales. Comparing to the single CSA, the PCSA uses multi-head mechanism, incorporating each head with different window sizes and dilations to adapt different motion situations. Notably, the computation overhead of CSA are linear function of channels number C. The PCSA first splits channels into multiple groups, thus its computation overhead is the same as single CSA. Fig. 4 shows the computation and memory overhead comparison. We evaluate the effectiveness of the PCSA operation in the following experiments.

Implementation

Network Architecture. Our network is built upon MobileNetV3 (Howard et al. 2019), a light-weight backbone. For encoder, we change the third and the last convolution stride from 2 to 1, in order to preserve spatial resolution in feature map. We add a modified RFB (Liu, Huang, and others 2018) block at the head of backbone. Specially, we replace the vanilla convolution in RFB block by separable convolution. For PCSA module, we set $g = 4$, $r = \{3, 4\}$ and $d = \{1, 2\}$. For decoder, we use the second stage output of the encoder as low-level feature. Then we get the spatiotemporal feature from PCSA output. We use a dilated convolution to reduce the spatiotemporal feature dimension from 128 to 32. Bilinear interpolation is applied to upsample spatiotemporal feature to match the low-level feature size. We

Test Dataset		MBD	MSTM	STBP	SCOM	SCNN	DLVS	FGRN	MBNM	PDBM	SSAV	Ours
DAVIS	max F↑	0.470	0.429	0.544	0.783	0.714	0.708	0.783	0.861	0.855	0.861	**0.880**
	S↑	0.597	0.583	0.677	0.832	0.783	0.794	0.838	0.887	0.882	0.893	**0.902**
	MAE↓	0.177	0.165	0.096	0.048	0.064	0.061	0.043	0.031	0.028	0.028	**0.022**
FBMS	max F↑	0.487	0.500	0.595	0.797	0.762	0.759	0.767	0.816	0.821	**0.865**	0.831
	S↑	0.609	0.613	0.627	0.794	0.794	0.794	0.809	0.857	0.851	**0.879**	0.866
	MAE↓	0.206	0.177	0.152	0.079	0.095	0.091	0.088	0.047	0.064	**0.040**	0.041
ViSal	max F↑	0.692	0.673	0.622	0.831	0.831	0.852	0.848	0.883	0.888	0.939	**0.940**
	S↑	0.726	0.749	0.629	0.762	0.847	0.881	0.861	0.898	0.907	0.943	**0.946**
	MAE↓	0.129	0.095	0.163	0.122	0.071	0.048	0.045	0.020	0.032	0.020	**0.017**
SegV2	max F↑	0.554	0.526	0.640	0.764	-	-	-	0.716	0.800	0.801	**0.810**
	S↑	0.618	0.643	0.735	0.815	-	-	-	0.809	0.864	0.851	**0.865**
	MAE↓	0.146	0.114	0.061	0.030	-	-	-	0.026	0.024	**0.023**	0.025
VOS	max F↑	0.562	0.336	0.403	0.690	0.609	0.675	0.669	0.670	0.742	0.742	**0.747**
	S↑	0.661	0.551	0.614	0.712	0.704	0.760	0.715	0.742	0.818	0.819	**0.827**
	MAE↓	0.158	0.145	0.105	0.162	0.109	0.099	0.097	0.099	0.078	0.073	**0.065**
DAVSOD	max F↑	0.342	0.344	0.410	0.464	0.532	0.521	0.573	0.520	0.572	0.603	**0.655**
	S↑	0.538	0.532	0.568	0.599	0.674	0.657	0.693	0.637	0.698	0.724	**0.741**
	MAE↓	0.228	0.211	0.160	0.220	0.128	0.129	0.098	0.159	0.116	0.092	**0.086**
Runtime (s)↓		0.02	0.02	49.49	38.8	38.5	0.47	0.09	2.63	0.05	0.05	**0.009**

Table 1: Comparison with previous state-of-the-art methods on six challenge VSOD datasets. ↑ means larger is better and ↓ means smaller is better. **Bold** means the state-of-the-art performance. - means the method is trained on this dataset. We public the runtime to handle one frame in the last row.

concat low-level feature and spatiotemporal feature, then use a convolution (kernel=3) to get final prediction.

Loss Function. During the training phase, we use a binary cross entropy loss function. We denote our prediction as P, the Ground Truth of saliency map is G. Then binary cross entropy loss L_{bce} can be defined as

$$L_{bce}(P,G) = -\frac{1}{N}\sum_{i=1}^{N}\left[g_i log(p_i) + (1-p_i) log(1-s_i)\right]. \tag{9}$$

Here, N is the number of pixels.

Training Protocol. Our model is built based on pytorch (Paszke et al. 2019) repository. Following previous methods, we use an image saliency dataset to pre-train our backbone. Then we use video datasets to finetune our PCSA module. Total training procedure takes 15 hours on 4×rtx 2080ti.

Pre-train phase We remove the PCSA module and pre-train our backbone with the training set of an image dataset, i.e. DUTS (Wang et al. 2017) and two video datasets, i.e. DAVIS (Perazzi et al. 2016) and DAVSOD (Fan et al. 2019). We use adam optimizer with initial learning rate 2e-4 and batch sizes 36. The learning rate decays with poly scheduler (decay rate=0.9). We resize input images to 256×448. The data augmentation methods contain randomly flip, randomly crop, and multi-scale training. We use five scales $\{0.5, 0.75, 1, 1.25, 1.75\}$ when training. Pre-training takes about 7 hours with total 15 epoches.

Method		FLOPs	Mem	maxF	MAE	S
baseline		-	-	.856	.029	.886
+ NL		+10.19G	+321M	.861	.028	.890
+ CSA	r=3	+0.05G	+1.61M	.869	.026	.895
	r=5	+0.14G	+4.48M	.875	.024	.898
	r=7	+0.27G	+8.76M	.873	.024	.897
+ PCSA		+0.05G	+6.44M	.880	.022	.902

Table 2: Quantitative results of different models on DAVIS test set.

Finetune phase After pre-training, we incorporate our PCSA to the final stage of encoder. We use two video datasets mentioned above to train whole network. We set $T = 5$ and batch sizes 12 in our experiments. The initial learning rate of PCSA and the backbone is set to 10^{-4} and 10^{-6}, respectively. The learning rate schedule is the same as pre-train phase. We randomly choose interval (=$\{1,2,3\}$) to sample frames in video for augmentation. Randomly flip and randomly crop are also used to augment the training samples. The finetune phase takes about 9 hours with total 15 epoches.

Experiments

Experiment Setup

Dataset. We benchmark our method on six public VSOD datasets, i.e. FBMS (Ochs, Malik, and Brox 2013), DAVIS

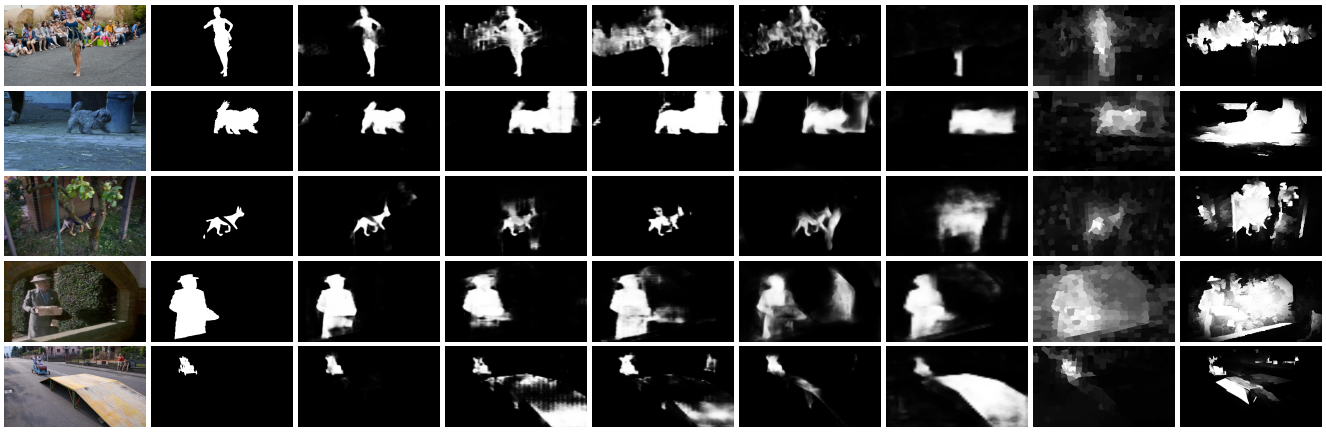


Figure 5: Visual comparison with previous state-of-the-art methods. We show the results in the several complex scenes. Our method can generate accurate saliency map in different complex scenes.

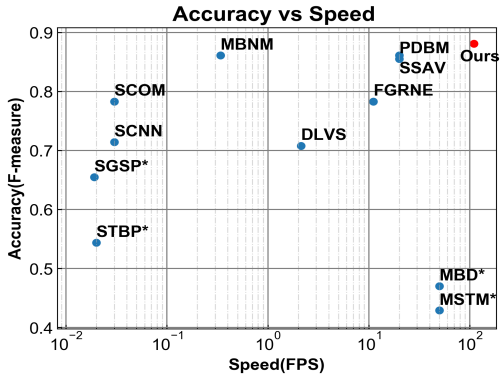


Figure 6: Accuracy vs Speed on DAVIS dataset. * means cpu time. Our method locates at the upper right hand corner, thus demonstrates its efficiency and effectiveness.

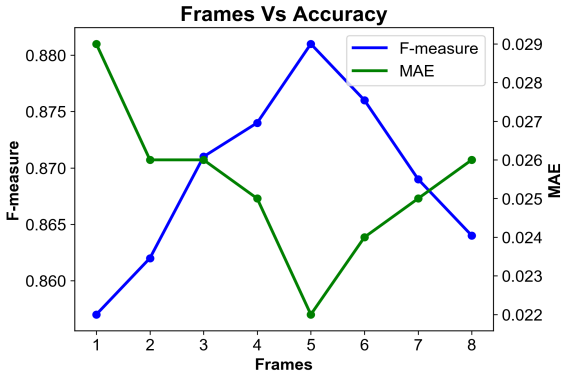


Figure 7: Sensitive analysis of frame numbers in a video segment.

(Perazzi et al. 2016), DAVSOD (Fan et al. 2019), SegTrack-V2 (Li et al. 2013), VOS (Li, Xia, and Chen 2017) and ViSal (Wang, Shen, and Shao 2015). Totally, the whole test dataset contains 155 videos.

Metrics. We use three criterions to evaluate our results, i.e. MAE, Fmeasure and Smeasure. F-measure is defined as

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}. \quad (10)$$

As suggested in (Achanta et al. 2009), β^2 is set to 0.3. We report the max Fmeasure as previous works (Fan et al. 2019; Li et al. 2018a) done. MAE measures absolute pixel errors between ground truth and our prediction:

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)|. \quad (11)$$

Smeasure (Fan et al. 2017) is a newly introduced measurement, focusing on the similarity of object shape.

Performance Comparison

We compare our method with eleven previous state-of-the-art methods, including four conventional methods: MBD (Zhang et al. 2015), MSTM (Tu et al. 2016), STBP (Xi et al. 2016), SGSP (Liu et al. 2016) and seven deep-learning based methods: SCOM (Chen et al. 2018), SCNN (Tang et al. 2018), DLVS (Wang, Shen, and Shao 2017), FGRN (Li et al. 2018a), MBNM (Li et al. 2018b), PDBM (Song et al. 2018), SSAV (Fan et al. 2019). We use evaluation code provided by (Fan et al. 2019) for a fair comparison. Quantitative comparison results are summarized in Tab. 1. Our PCSA achieves state-of-the-art results in five datasets. We improve SSAV (Fan et al. 2019) 8.45% on DAVSOD and 2.3% on DAVIS. Besides, we achieve $5.5\times$ faster speed than SSAV. Fig. 5 shows the visual comparison between our method and previous methods in several complex scenes. Results show our method can capture motion cues in complex environments and produces accurate results.

Runtime Analysis. We measure the accuracy and speed of different methods. Speed is tested on Intel(R) Core(TM) i7-

Dataset	Metrics	MotAdapt	COSNet	EpO+	LSMO	PDB	ARP	LVO	FSEG	LMP	AGS	Ours
DAVIS	$\mathcal{J}Mean$	77.2	80.5	80.6	78.2	77.2	76.2	75.9	70.7	70.0	79.7	78.1
	$\mathcal{J}Recall$	87.8	93.1	95.2	89.1	90.1	91.1	89.1	83.5	85.0	91.1	90.1
	$\mathcal{F}Mean$	77.4	79.5	75.5	75.9	74.5	70.6	72.1	65.3	65.9	77.4	78.5
	$\mathcal{F}Recall$	84.4	89.5	87.9	84.7	84.4	83.5	83.4	73.8	79.2	85.8	88.2
Use CRF			√	√	√	√		√		√	√	
Realtime												√

Table 3: Quality comparison with UVOS methods, the results are taken down from DAVIS public leaderboard (https://davischallenge.org/davis2016/soa_compare.html). Our method achieves real-time speed and comparable results without using CRF post-processing.

4790K CPU and a single Titan Xp GPU. Results are shown in Fig. 6. Conventional methods MBD and MSTM can reach 50 FPS but they are inaccurate. PDBM and SSAV can reach 20 FPS, but they are still marginally lower than real-time. Our method locates at top-right, which demonstrates its efficiency and accuracy.

Ablation Study

Effectiveness of the PCSA. We set the pretrained backbone as our baseline. Then we evaluate the effectiveness of our proposed PCSA. We also compare the PCSA with the non-local operation and the single CSA. We investigate the configurations of using different window sizes R on the single CSA operation. From Tab. 2, we can find the non-local module only achieves 0.5% improvement of baseline. Our PCSA improves 2.9% of baseline and needs only 2% FLOPs of the non-local. Besides, we find with the increase of the window size, the accuracy do not increase consistently. The single CSA with small window size $r = 3$ fails to capture salient object in fast speed. The single CSA with larger window size focuses more on global context instead of motion information. The non-local operation is the special case of the largest window size of CSA. The PCSA integrates multi-scale temporal information, which outperforms the single CSA operation with the same FLOPs.

Sensitive to frame numbers in one segment. Our method is sensitive to choose appropriate frames when exploiting motion information. We choose $T = \{1, 2, 3, 4, 5, 6, 7, 8\}$ for evaluation. From Fig. 7, we can find the static images $(t = 1)$ are not beneficial to our PCSA. When T is set to 5, the model achieves the best result. Continuously increasing T degrades the performance of the PCSA. Because larger time interval may cause objects to disappear in the reference window, which is harmful for the PCSA to learn motion cues.

Performance on Unsupervised Video Object Segmentation

We compare our method with eight state-of-the-art methods: MotAdapt (Siam et al. 2019), COSnet (Lu et al. 2019), EpO+ (Faisal et al. 2019), LSMO (Tokmakov, Schmid, and Alahari 2019), PDB (Song et al. 2018), ARP (Koh and Kim 2017), LVO (Tokmakov, Alahari, and Schmid 2017b), FSEG (Jain, Xiong, and Grauman 2017), LMP (Tokmakov, Alahari, and Schmid 2017a), AGS (Wang et al. 2019a). Following the evaluation setting of UVOS, we use region similarity $\mathcal{J}$, boundary accuracy $\mathcal{F}$ for evaluation, as suggested in (Perazzi et al. 2016). We do not use CRF post-processing. Instead, we adopt a simple threshold method (threshold=0.4) to binarize the saliency prediction. Tab. 3 demonstrates the evaluation results. None of previous methods can reach real-time speed while our method can run at 110 FPS. We achieve the best accuarcy-efficiency trade-off in UVOS task.

Conclusion

We propose a pyramid constrained self-attention for VSOD in this paper, which can capture motion cues efficiently. We constrain the reference area of the non-local operation when querying a position. Such a constraint is based on motion prior, and prevents network from learning global context rather than motion cues in VSOD. The pyramid structure is used to group several CSA operations for multi-scale objects and various speeds. Experiments show our PCSA can effectively capture motion cues with much less computation and memory usage than the non-local. Our model can reach 110 FPS on a TitanXp GPU, and achieves outstanding results in both VSOD and UVOS tasks.

Acknowledgment

This work was supported by Major Project for New Generation of AI (No. 2018AAA010040003), Tianjin Natural Science Foundation (No. 18JCYBJC41300 and No. 18ZXZNGX00110).

References

Achanta, R.; Hemami, S.; Estrada, F.; and Süsstrunk, S. 2009. Frequency-tuned salient region detection. In *IEEE CVPR*.

Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv preprint arXiv:1904.11492*.

Chen, Y.; Zou, W.; Tang, Y.; Li, X.; Xu, C.; and Komodakis, N. 2018. Scom: Spatiotemporal constrained optimization for salient object detection. *IEEE TIP* 27(7):3345–3357.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015.

Flownet: Learning optical flow with convolutional networks. In *IEEE ICCV*, 2758–2766.

Faisal, M.; Akhter, I.; Ali, M.; and Hartley, R. 2019. Exploiting geometric constraints on dense trajectories for motion saliency. *arXiv preprint arXiv:1909.13258*.

Fan, D.-P.; Cheng, M.-M.; Liu, Y.; Li, T.; and Borji, A. 2017. Structure-measure: A new way to evaluate foreground maps. In *IEEE ICCV*, 4548–4557.

Fan, D.-P.; Wang, W.; Cheng, M.-M.; and Shen, J. 2019. Shifting more attention to video salient object detection. In *IEEE CVPR*.

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2018. Ccnet: Criss-cross attention for semantic segmentation. *arXiv preprint arXiv:1811.11721*.

Itti, L.; Koch, C.; and Niebur, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI* 1254–1259.

Jain, S. D.; Xiong, B.; and Grauman, K. 2017. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *IEEE CVPR*, 2117–2126. IEEE.

Koh, Y. J., and Kim, C.-S. 2017. Primary object segmentation in videos based on region augmentation and reduction. In *IEEE CVPR*, 7417–7425. IEEE.

Le, T.-N., and Sugimoto, A. 2018. Video salient object detection using spatiotemporal deep features. *IEEE TIP* 27(10):5002–5015.

Li, F.; Kim, T.; Humayun, A.; Tsai, D.; and Rehg, J. M. 2013. Video segmentation by tracking many figure-ground segments. In *IEEE ICCV*, 2192–2199.

Li, G.; Xie, Y.; Wei, T.; Wang, K.; and Lin, L. 2018a. Flow guided recurrent neural encoder for video salient object detection. In *IEEE CVPR*, 3243–3252.

Li, S.; Seybold, B.; Vorobyov, A.; Lei, X.; and Jay Kuo, C.-C. 2018b. Unsupervised video object segmentation with motion-based bilateral networks. In *ECCV*, 207–223.

Li, J.; Xia, C.; and Chen, X. 2017. A benchmark dataset and saliency-guided stacked autoencoders for video-based salient object detection. *IEEE TIP* 27(1):349–364.

Liu, Z.; Li, J.; Ye, L.; Sun, G.; and Shen, L. 2016. Saliency detection for unconstrained videos using superpixel-level graph and spatiotemporal propagation. *IEEE TCSVT* 27(12):2527–2542.

Liu, S.; Huang, D.; et al. 2018. Receptive field block net for accurate and fast object detection. In *ECCV*, 385–400.

Lu, X.; Wang, W.; Ma, C.; Shen, J.; Shao, L.; and Porikli, F. 2019. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *IEEE CVPR*, 3623–3632.

Ochs, P.; Malik, J.; and Brox, T. 2013. Segmentation of moving objects by long term video analysis. *IEEE TPAMI* 36(6):1187–1200.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, 8024–8035.

Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; and Sorkine-Hornung, A. 2016. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE CVPR*, 724–732.

Siam, M.; Jiang, C.; Lu, S.; Petrich, L.; Gamal, M.; Elhoseiny, M.; and Jagersand, M. 2019. Video object segmentation using teacher-student adaptation in a human robot interaction (hri) setting. In *2019 International Conference on Robotics and Automation (ICRA)*, 50–56. IEEE.

Song, H.; Wang, W.; Zhao, S.; Shen, J.; and Lam, K.-M. 2018. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 715–731.

Tang, Y.; Zou, W.; Jin, Z.; Chen, Y.; Hua, Y.; and Li, X. 2018. Weakly supervised salient object detection with spatiotemporal cascade neural networks. *IEEE TCSVT*.

Tokmakov, P.; Alahari, K.; and Schmid, C. 2017a. Learning motion patterns in videos. In *IEEE CVPR*, 3386–3394.

Tokmakov, P.; Alahari, K.; and Schmid, C. 2017b. Learning video object segmentation with visual memory. In *IEEE ICCV*, 4481–4490.

Tokmakov, P.; Schmid, C.; and Alahari, K. 2019. Learning to segment moving objects. *International Journal of Computer Vision* 127(3):282–301.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *IEEE ICCV*, 4489–4497.

Tu, W.-C.; He, S.; Yang, Q.; and Chien, S.-Y. 2016. Real-time salient object detection with a minimum spanning tree. In *IEEE CVPR*, 2334–2342.

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 *NIPS*, 5998–6008.

Wang, L.; Lu, H.; Wang, Y.; Feng, M.; Wang, D.; Yin, B.; and Ruan, X. 2017. Learning to detect salient objects with image-level supervision. In *IEEE CVPR*, 136–145.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *IEEE CVPR*, 7794–7803.

Wang, W.; Song, H.; Zhao, S.; Shen, J.; Zhao, S.; Hoi, S. C.; and Ling, H. 2019a. Learning unsupervised video object segmentation through visual attention. In *IEEE CVPR*, 3064–3074.

Wang, Z.; Xu, J.; Liu, L.; Zhu, F.; and Shao, L. 2019b. Ranet: Ranking attention network for fast video object segmentation. In *IEEE ICCV*.

Wang, W.; Shen, J.; and Porikli, F. 2015. Saliency-aware geodesic video object segmentation. In *IEEE CVPR*, 3395–3402.

Wang, W.; Shen, J.; and Shao, L. 2015. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE TIP* 24(11):4185–4196.

Wang, W.; Shen, J.; and Shao, L. 2017. Video salient object detection via fully convolutional networks. *IEEE TIP* 27(1):38–49.

Wu, H.; Li, G.; and Luo, X. 2014. Weighted attentional blocks for probabilistic object tracking. *The Visual Computer* 30(2):229–243.

Xi, T.; Zhao, W.; Wang, H.; and Lin, W. 2016. Salient object detection with spatiotemporal background priors for video. *IEEE TIP* 26(7):3425–3436.

Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 802–810.

Xu, N.; Price, B.; Cohen, S.; Yang, J.; and Huang, T. S. 2016. Deep interactive object selection. In *IEEE CVPR*, 373–381.

Yue, K.; Sun, M.; Yuan, Y.; Zhou, F.; Ding, E.; and Xu, F. 2018. Compact generalized non-local network. In *NIPS*, 6510–6519.

Zhang, J.; Sclaroff, S.; Lin, Z.; Shen, X.; Price, B.; and Mech, R. 2015. Minimum barrier salient object detection at 80 fps. In *IEEE ICCV*, 1404–1412.