

Primary Video Object Segmentation via Complementary CNNs and Neighborhood Reversible Flow

Jia Li^{1,2*}, Anlin Zheng¹, Xiaowu Chen^{1*}, Bin Zhou¹

¹State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University ²International Research Institute for Multidisciplinary Science, Beihang University

Abstract

This paper proposes a novel approach for segmenting primary video objects by using Complementary Convolutional Neural Networks (CCNN) and neighborhood reversible flow. The proposed approach first pre-trains C-CNN on massive images with manually annotated salient objects in an end-to-end manner, and the trained CCNN has two separate branches that simultaneously handle two complementary tasks, i.e., foregroundness and backgroundness estimation. By applying CCNN on each video frame, the spatial foregroundness and backgroundness maps can be initialized, which are then propagated between various frames so as to segment primary video objects and suppress distractors. To enforce efficient temporal propagation, we divide each frame into superpixels and construct neighborhood reversible flow that reflects the most reliable temporal correspondences between superpixels in far-away frames. Within such flow, the initialized foregroundness and backgroundness can be efficiently and accurately propagated along the temporal axis so that primary video objects gradually pop-out and distractors are well suppressed. Extensive experimental results on three video datasets show that the proposed approach achieves impressive performance in comparisons with 18 state-of-the-art models.

1. Introduction

Segmenting primary objects in images and videos is an important step in many computer vision applications. In recent years, the segmentation of primary image objects, namely image-based salient object detection, has achieved impressive success since powerful models can be directly trained on large image datasets by using Random Forest [12], Multi-instance Learning [42], Stacked Autoencoders [8] and Deep Neural Networks [13, 15, 18, 45]. In contrast, segmenting primary video objects remains a chal-



Figure 1. Primary objects may co-occur with or be occluded by various distractors. They may not always be the most salient ones in each separate frame but can consistently pop-out in most video frames (frames and masks are taken from the dataset [17]).

lenging task since the amount of video data with per-frame pixel-level annotations is much less than that of images, which may prevent the direct end-to-end training of spatiotemporal model. Moreover, due to the camera or object motion, the same primary objects may co-occur with or be occluded by various distractors in different frames (see Fig. 1), making them difficult to consistently pop-out throughout the whole video.

To address the problem of primary video object segmentation, there exist three major categories of models that can be recognized as fully-automatic, interactive and weakly-supervised, respectively. In these models, interactive ones require manually annotated primary objects in the first frame or several key frames before starting the automatic segmentation [5, 19, 26, 27], while weakly-supervised ones often assume that the semantic tags of primary video objects are known before segmentation so that additional tools like object detectors can be reused [31, 44]. However, the requirement of interaction or semantic tags prevents their usage in processing large-scale video data [40].

Beyond these models, fully-automatic models aim to directly segment primary objects in a single video [10, 22, 40, 36] or co-segment the primary objects shared by a collection of videos [2, 7, 41]. Typically, these models propose assumptions on what spatial attributes primary objects should have and how they should behave along the temporal axis. For example, Papazoglou and Ferrari [22] assume that foreground objects should move differently from its surrounding background in a good fraction of the video.

^{*} Correspondence should be addressed to Xiaowu Chen and Jia Li. E-mail: {chen, jiali}@buaa.edu.cn



Figure 2. Framework of the proposed approach. The framework consists of two major modules. The spatial module trains CCNN to simultaneously initialize the foregroundness and backgroundness maps of each frame. This module operates on GPU to provide pixel-wise predictions for each frame. The temporal module constructs neighborhood reversible flow so as to propagate foregroundness and backgroundness along the most reliable inter-frame correspondences. This module operates on superpixels for efficient temporal propagation.

They first initialize foreground maps with motion information and then refine them in the spatiotemporal domain so as to enhance the smoothness of foreground objects. Zhang et al. [40] assume that objects are spatially dense and their shapes and locations change smoothly across frames. A layered Directed Acyclic Graph based framework is presented for primary video object segmentation. Actually, similar assumptions can be found in many models [6, 14, 21, 34, 35] that perform impressively on small datasets (e.g., Seg-Track [30] and SegTrack V2 [14]). However, these models sometimes may fail on certain videos from larger datasets (e.g., Youtube-Objects [24] and VOS [17]) that contain many complex scenarios in which the assumptions may not always hold. Moreover, many of these models compute pixel-wise optical flow to temporally refine the initial spatial predictions between adjacent frames. On the one hand, adjacent frames often have similar content and demonstrate similar successes and failures in initial predictions, which may degrade the effect of temporal refinement even though such short-term optical flow can be very accurate. On the other hand, the optical flow between far-away frames may become not very accurate due to the probable occlusion and large displacements [3], making the long-term temporal propagation not very reliable.

Considering all these issues, this paper proposes an novel approach that efficiently predicts and propagates spatial foregroundness and backgroundness within neighborhood reversible flow for primary video object segmentation. The framework of the proposed approach is shown in Fig. 2, which consists of two main modules. In the spatial module, the Complementary Convolutional Neural Networks (CCN-N) is trained end-to-end on massive images with manually annotated salient objects so as to simultaneously handle two complementary tasks, i.e., foreground and backgroundness estimation, with two separate branches. By using CC-NN, we can obtain the initialized foregroundness and backgroundness maps on each individual frame. To efficiently and accurately propagate such spatial predictions between far-away frames, we further divide each frame into a set of superpixels and construct neighborhood reversible flow so as to depict the most reliable temporal correspondences between superpixels in different frames. Within such flow, the initialized spatial foregroundness and backgroundness are efficiently propagated along the temporal axis by solving a quadratic programming problem that has analytic solution. In this manner, primary objects can efficiently pop-out and distractors can be further suppressed. Extensive experiments on three video datasets show that the proposed approach acts efficient and achieves impressive performances compared with 18 state-of-the-art models (7 image-based & non-deep, 6 image-based & deep, 5 video-based).

The main contributions of this paper include: 1) we propose a simple yet effective framework for efficient and accurate primary video object segmentation; 2) we train endto-end CNNs that simultaneously address two dual tasks so that primary video objects can be detected from complementary perspectives; and 3) we construct neighborhood reversible flow between superpixels that can effectively propagate foregroundness and backgroundness along the most reliable inter-frame correspondences.

2. Training CCNN for Spatial Foregroundness and Backgroundness Initialization

Without loss of generality, primary video objects can be defined as the objects that consistently pop-out throughout a whole video [17]. In other words, distractors, which may have the capability of suddenly capturing human visual attention in certain video frames, will be well suppressed in the majority of the same video. Inspired by these facts, we propose to train deep neural networks that simultaneously predict both spatial foregroundness and backgroundness of a frame. The architecture of the proposed CCNN can be found in Fig. 3, which starts with a shared trunk and ends up with two separate branches.

The shared trunk are initialized from the VGG16 networks [28]. In our experiments, we remove all the pooling layers after CONV4_1 and conduct dilated convolution in all subsequent CONV layers to extend the receptive fields without loss of resolution [39]. Moreover, the last two fully connected layers are converted into convolutional layers with 7×7 and 1×1 kernels, respectively. The revised VG-G16 architecture takes a 280×280 image as the input, and outputs a 35×35 feature map with 4096 channels. Finally, a CONV layer with 1×1 kernels is adopted to compress the feature map into 128 channels.

After the shared trunk, the neural networks split into two separate branches that address two complementary tasks, *i.e.*, foregroundness and backgroundness estimation. Note that these two branches utilize the same input and the same architecture. That is, the input feature map first enter three parallel CONV layers with 3×3 kernels and dilation of 1, 2 and 3, respectively. This ensures the measurement of foregroundness and backgroundness at three consecutive scales. After that, the outputs of these three layers are concatenated, which are then compressed with an 1×1 CONV layer and output foregroundness and backgroundness maps via DECONV layers and sigmoid activation functions. In this manner, the foregroundness branch mainly focuses on detecting salient objects, while the backgroundness branch focuses on suppressing distractors.

As shown in Fig. 4, many foregroundness and backgroundness maps can already well depict primary video objects and distractors from various video frames (see Fig. 4 (a)-(d)). However, such maps are not always perfectly complementary. That is, they sometimes leave certain area mistakenly predicted in both the foreground and background branches (see Fig. 4 (h) for the black area in the fusion maps that are generated as the maximum of the foregroundness and backgroundness maps). Typically, Such imperfect predictions need to be further refined in temporal propagation, which will be discussed in the next section.

In the training stage, we collect massive images with labeled salient objects from four datasets for image-based



Figure 3. Architecture of CCNN. Here 'CONV $(3 \times 3/2)$ ' indicates a convolutional layer with 3×3 kernels and dilation of 2.



Figure 4. Foregroundness and backgroundness maps initialized by CCNN as well as their fusion maps (*i.e.*, maximum values from two maps). (a) and (e) Video frames, (b) and (f) foregroundness maps, (c) and (g) backgroundness maps, (d) and (h) fusion maps. We can see that the foregroundness and backgroundness maps can well depict salient objects and distractors in many frames (see (a)-(d)). However, they are not always perfectly complementary, leaving some area mistakenly predicted in both foreground and background maps (see the black area in fusion maps (h)).

salient object detection [15, 20, 37, 38]. To speed up the training process and reduce the number of parameters, we down-sample all images to 280×280 and their groundtruth saliency maps into 140×140 . Suppose that the output foregroundness and backgroundness maps of the uth training image are represented by \mathbf{X}_u and \mathbf{Y}_u , we optimize the parameters of CCNN by minimizing the overall Sigmoid Cross Entropy loss among $\{(\mathbf{X}_u, \mathbf{G}_u)\}_u$ and $\{(\mathbf{Y}_u, \mathbf{G}_u)\}_u$. Note that G_u is the ground-truth map of the *u*th image, in which zeros represent distractors and ones represent salient objects. In the training process, the loss in the two branches has equal weights, and we adopt different learning rates at the shared trunk (5×10^{-8}) and the two branches (5×10^{-7}) so that the knowledge from VGG16 can be largely maintained. Other hyper-parameters include batch size (4), momentum (0.9) and weight decay (0.0005).

3. Building Neighborhood Reversible Flow for Efficient Temporal Propagation

3.1. Neighborhood Reversible Flow

To extend spatial foregroundness and backgroundness to the spatiotemporal domain, we can propagate them along the temporal axis. In the propagation, we should first determine which frames should be referred to and how to build reliable inter-frame correspondences. Considering that there often exists rich redundancy among adjacent frames, we assume that far-away frames can bring more valuable cues on how to pop-out primary objects and suppress distractors. Unfortunately, the pixel-wise optical flow may suffer from large displacement in handling far-away frames. Toward this end, we construct neighborhood reversible flow based on superpixels, which is inspired by the concept of neighborhood reversibility in image search [11].

In building the flow, we first apply the SLIC algorithm [1] to divide a frame \mathcal{I}_u into N_u superpixels, denoted as $\{\mathcal{O}_{ui}\}$. To characterize a superpixel from multiple perspectives, we compute its average RGB, Lab and HSV colors as well as the horizontal and vertical positions. By normalizing these features into the same dynamic range [0, 1], we compute the $\ell 1$ distances between \mathcal{O}_{ui} and all superpixels in another frame \mathcal{I}_v . After that, we denote the k nearest neighbors of \mathcal{O}_{ui} in \mathcal{I}_v as $\mathcal{N}_k(\mathcal{O}_{ui}|\mathcal{I}_v)$. As a consequence, two superpixels are k-neighborhood reversible if they reside in the k nearest neighbors of each other. That is,

$$\mathcal{O}_{ui} \in \mathcal{N}_k(\mathcal{O}_{vj}|\mathcal{I}_u) \text{ and } \mathcal{O}_{vj} \in \mathcal{N}_k(\mathcal{O}_{ui}|\mathcal{I}_v).$$
 (1)

From (1), we find that the smaller k, the more tightly two superpixels are temporally correlated. Therefore, we initialize the flow between \mathcal{O}_{ui} and \mathcal{O}_{vj} as

$$f_{ui,vj} = \begin{cases} \exp(-2k/k_0) & \text{if } k \le k_0 \\ 0 & \text{otherwise} \end{cases}$$
(2)

where k_0 is a constant to suppress weak flow and k is a variable. A small k_0 will build sparse correspondences between \mathcal{I}_u and \mathcal{I}_v (e.g., $k_0 = 1$), while a large k_0 will cause dense correspondences. In this study, we empirically set $k_0 = 15$ and represent the flow between \mathcal{I}_u and \mathcal{I}_v with a matrix $\mathbf{F}_{uv} \in \mathbb{R}^{N_u \times N_v}$, in which the component at (i, j) equals to $f_{ui,vj}$. Note that we further normalize \mathbf{F}_{uv} so that each row sums up to 1. To ensure sufficient variation in content and depict reliable temporal correspondences, we only estimate the flow between a frame \mathcal{I}_u and the frames $\{\mathcal{I}_t | t \in \mathbb{T}_u\}$, where \mathbb{T}_u can be empirically set to $\{u - 30, u - 15, u + 15, u + 30\}$.

3.2. Temporal Propagation within Flow

The flow $\{\mathbf{F}_{uv}\}$ depicts how superpixels in various frames are temporally correlated, which can be used to fur-

ther propagate the spatial foregroundness and backgroundness. Typically, such temporal refinement can obtain impressive performance by solving a complex optimization problem with constraints like spatial compactness and temporal consistency. However, the time cost will also grow surprisingly high [44]. Considering the requirement of efficiency in many real-world applications, we propose to minimize an objective function that has analytic solution. For a superpixel \mathcal{O}_{ui} , its foregroundness x_{ui} and backgroundness y_{ui} can be initialized as

$$x_{ui} = \frac{\sum_{p \in \mathcal{O}_{ui}} \mathbf{X}_u(p)}{|\mathcal{O}_{ui}|}, \ y_{ui} = \frac{\sum_{p \in \mathcal{O}_{ui}} \mathbf{Y}_u(p)}{|\mathcal{O}_{ui}|}$$
(3)

where p is a pixel with foregroundness $\mathbf{X}_u(p)$ and backgroundness $\mathbf{Y}_u(p)$. $|\mathcal{O}_{ui}|$ is the area of \mathcal{O}_{ui} . For the sake of simplification, we represent the foregroundness and backgroundness scores of all superpixels in the *u*th frame with column vectors \mathbf{x}_u and \mathbf{y}_u , respectively. As a result, we can propagate such scores from \mathcal{I}_v to \mathcal{I}_u according to \mathbf{F}_{uv} :

$$\mathbf{x}_{u|v} = \mathbf{F}_{uv} \mathbf{x}_{v}, \ \mathbf{y}_{u|v} = \mathbf{F}_{uv} \mathbf{y}_{v}, \ \forall v \in \mathbb{T}_{u}.$$
(4)

After the propagation, the foregroundness vector $\hat{\mathbf{x}}_u$ and backgroundness vector $\hat{\mathbf{y}}_u$ can be refined by solving

$$\hat{\mathbf{x}}_{u} = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{x}_{u}\|_{2}^{2} + \lambda_{c} \sum_{v \in \mathbb{T}_{u}} \|\mathbf{x} - \mathbf{x}_{u|v}\|_{2}^{2},$$

$$\hat{\mathbf{y}}_{u} = \arg\min_{\mathbf{y}} \|\mathbf{y} - \mathbf{y}_{u}\|_{2}^{2} + \lambda_{c} \sum_{v \in \mathbb{T}_{u}} \|\mathbf{y} - \mathbf{y}_{u|v}\|_{2}^{2},$$
(5)

where λ_c is a positive constant whose value is empirically set to 0.5. Note that we adopt only the ℓ^2 norm in (5) so as to efficiently compute an analytic solution

$$\hat{\mathbf{x}}_{u} = \frac{1}{1 + |\mathbb{T}_{u}|} \left(\mathbf{x}_{u} + \lambda_{c} \sum_{v \in \mathbb{T}_{u}} \mathbf{F}_{uv} \mathbf{x}_{v} \right),$$

$$\hat{\mathbf{y}}_{u} = \frac{1}{1 + |\mathbb{T}_{u}|} \left(\mathbf{y}_{u} + \lambda_{c} \sum_{v \in \mathbb{T}_{u}} \mathbf{F}_{uv} \mathbf{y}_{v} \right).$$
(6)

By observing (4) and (6), we find that the propagation process is actually calculating the average foregroundness and backgroundness scores within a local temporal slice under the guidance of neighborhood reversible flow. After the temporal propagation, we turn superpixel-based scores into pixel-based ones as

$$\mathbf{M}_{u}(p) = \sum_{i=1}^{N_{u}} \delta(p \in \mathcal{O}_{ui}) \cdot \hat{x}_{ui} \cdot (1 - \hat{y}_{ui}), \qquad (7)$$

where \mathbf{M}_u is the importance map of \mathcal{I}_u that depict the presence of primary objects. $\delta(p \in \mathcal{O}_{ui})$ is an indicator function which equals to 1 if $p \in \mathcal{O}_{ui}$ and 0 otherwise. Finally, we calculate an adaptive threshold which equals to the 20% of the maximal pixel importance to binarize each frame, and a morphological closing operation is then performed to fill in the black area in the segmented objects.

4. Experiments

We test the proposed approach on three video datasets that have different ways in defining primary video objects. Details of these datasets are described as follows:

1) SegTrack V2 [14] is a classic dataset in video object segmentation that are frequently used in many previous works. It consists of 14 densely annotated video clips with 1,066 frames in total. Most primary objects in this dataset are defined as the ones with *irregular motion patterns*.

2) Youtube-Objects [24] contains a large amount of Internet videos and we adopt its subset [9] that contains 127 videos with 20,977 frames. In these videos, 2,153 keyframes are sparsely sampled and manually annotated with pixel-wise masks according to the video tags. In other words, primary objects in **Youtube-Objects** are defined from the perspective of *semantic attributes*.

3) VOS [17] contains 200 videos with 116,093 frames. On 7,467 uniformly sampled keyframes, all objects are presegmented by 4 subjects, and the fixations of another 23 subjects are collected in eye-tracking tests. With these annotations, primary objects are automatically selected as the ones whose average fixation densities over the whole video fall above a predefined threshold. If no primary objects that receive the highest average fixation density will be treated as the primary ones. Different from **SegTrack V2** and **Youtube-Objects**, primary objects in **VOS** are defined from the perspective of *human visual attention*.

On these three datasets, the proposed approach, denoted as **OUR**, is compared with 18 state-of-the-art models for primary and salient object segmentation, including:

1) Image-based & Non-deep (7): **RBD** [46], **SMD** [23], **MB+** [43], **DRFI** [12], **BL** [29], **BSCA** [25], **MST** [32].

2) Image-based & Deep (6): ELD [13], MDF [15], D-CL [16], LEGS [18], MCDL [45] and RFCN [33].

3) Video-based (5): ACO [10], NLC [4], FST [22], SAG [34] and GF [35].

In the comparisons, we adopt two sets of evaluation metrics, including the Intersection-over-Union (IoU) and the Precision-Recall- F_{β} . Similar to [17], the precision, recall and IoU scores are first computed on each frame, which are then separately averaged over each video and finally averaged over the whole dataset so as to generate the mean Average Precision (mAP), mean Average Recall (mAR) and mean Average IoU (mIoU). In this manner, the influence of short and long videos can be balanced. Furthermore, a unique F_{β} score is computed based on mAR and mAP:

$$F_{\beta} = \frac{(1+\beta^2) \cdot \text{mAP} \cdot \text{mAR}}{\beta^2 \cdot \text{mAP} + \text{mAR}},$$
(8)

where we set $\beta^2 = 0.3$ as in many previous works to emphasize precision more than recall in the evaluation. Among

these metrics, mAR and mAP explain why a model performs impressive or fail from the complementary perspectives of recall and precision, which can help to provide more insights beyond a single mIoU score.

4.1. Comparisons with State-of-the-art Models

The performances of our approach and 18 state-of-theart models on three video datasets are shown in Table 1. Some representative results of our approach are demonstrated in Fig. 5. From Table 1, we find that on Youtube-**Objects** and **VOS** our approach achieves the best F_{β} and mIoU scores, while on SegTrack V2 our approach ranks the second places (worse than NLC). This can be explained by the fact that SegTrack V2 contains only 14 videos, among which most primary objects have irregular motion patterns. Such videos often perfectly meet the assumption of NLC on motion patterns of primary objects, making it the best approach on SegTrack V2. However, when the scenarios being processed extend to datasets like VOS that are constructed without such "constraints" on motion patterns, the performance of NLC drops sharply as its assumption may sometimes fail (e.g., VOS contains many videos only with static primary objects and distractors as well as slow camera motion, see Fig. 5). These results further validate that it is quite necessary to conduct comparisons on larger datasets with daily videos (like VOS) so that models with various kinds of assumptions can be fairly evaluated.

Moreover, there exist some approaches on the three datasets that outperform our approach in recall, while some other approaches may achieve better or comparable precision with our approach. However, none of these approaches simultaneously outperforms our approach in both recall and precision so that our approach often have better overall performance. This may imply that the proposed approach is more balanced than previous works. By analyzing the results on the three datasets, we find that this phenomena may be caused by the conduction of complementary tasks in CCNN. By propagating both foregroundness and backgroundness, the black area shown in the fusion maps can be filled in, while the mistakenly popped-out distractors can be suppressed again, leading to balanced recall and precision.

From Table 1, we also find that there exist inherent correlations between salient image object detection and primary video object segmentation. As shown in Fig. 5, primary objects are often the most salient ones in many frames, which explains the reason that deep models like **ELD**, **R**-**FCN** and **DCL** outperforms many video-based models like **NLC**, **SAG** and **GF**. However, there are several key differences between the two problems. First, primary objects may not always be the most salient ones in all frames (as shown in Fig. 1). Second, inter-frame correspondences provide additional cues for separating primary objects and distractors, which depict a new way to balance recall and pre-

Models		SegTrackV2 (14 videos)			Youtube-Objects (127 videos)			VOS (200 videos)					
		mAP	mAR	F_{β}	mIoU	mAP	mAR	F_{β}	mIoU	mAP	mAR	F_{β}	mIoU
Image + Non-deep	DRFI [12]	.464	.775	.511	.395	.542	.774	.582	.453	.597	.854	.641	.526
	RBD [46]	.380	.709	.426	.305	.519	.707	.553	.403	.652	.779	.677	.532
	BL [29]	.202	.934	.246	.190	.218	.910	.264	.205	.483	.913	.541	.450
	BSCA [25]	.233	.874	.280	.223	.397	.807	.450	.332	.544	.853	.594	.475
	MB+ [43]	.330	.883	.385	.298	.480	<u>.813</u>	.530	.399	.640	.825	.675	.532
	MST [32]	.450	.678	.488	.308	.538	.698	.568	.396	.658	.739	.675	.497
	SMD [23]	.442	.794	.493	.322	.560	.730	.592	.424	.668	.771	.690	.533
d	MDF [15]	.573	.634	.586	.407	.647	.776	.672	.534	.601	.842	.644	.542
)ee	ELD [13]	.595	.767	.627	.494	.637	.789	.667	.531	.682	.870	.718	.613
-	DCL [16]	.757	.690	.740	.568	.727	.764	.735	.587	<u>.773</u>	.727	.762	.578
ge	LEGS [18]	.420	.778	.470	.351	.549	.776	.589	.450	.606	.816	.644	.523
ma	MCDL [45]	.587	.575	.584	.424	.647	.613	.638	.471	.711	.718	.713	.581
Ι	RFCN [33]	.759	.719	.749	.591	.742	.750	.744	.592	.749	.796	.760	.625
ed	NLC [4]	.933	.753	.884	.704	.692	.444	.613	.369	.518	.505	.515	.364
o-bas(ACO [10]	.827	.619	.767	.551	.683	.481	.623	.391	.706	.563	.667	.478
	FST [22]	.792	.671	.761	.552	.687	.528	.643	.380	.697	.794	.718	.574
ide	SAG [34]	.431	.819	.484	.384	.486	.754	.529	.397	.538	.824	.585	.467
	GF [35]	.444	.737	.489	.354	.529	.722	.563	.407	.523	.819	.570	.436
	OUR	.809	.802	.807	.670	.745	.798	.756	.633	.789	.870	.806	.710

Table 1. Performances of our approach and 18 models. Bold and underline indicate the 1st and 2nd performance in each column.

Table 2. Performance of key components of our approach on **VOS**. FG: foregroundness branch of CCNN, BK: backgroundness branch of CCNN, NRF: Neighborhood Reversible Flow

Step	mAP	mAR	F_{β}	mIoU
FG (pixel-wise)	.750	.879	.776	.684
BK (pixel-wise)	.743	.884	.771	.680
FG (Superpixel)	.730	.895	.762	.674
BK (Superpixel)	.722	.901	.756	.669
FG+BK+NRF	.789	.870	.806	.710

cision. Third, primary objects may be sometimes close to video boundary due to camera and object motion, making the boundary prior widely used in many salient object detection models no valid any more (*e.g.*, the cow in the 1st row of the center column of Fig. 5). Last but not least, salient object detection needs to distinguish a salient object from a fixed set of distractors, while primary object segmentation needs to consistently pop-out the same primary object from a varying set of distractors. To sum up, primary video object segmentation is a more challenging task that needs to be further explored from the spatiotemporal perspective.

4.2. Detailed Performance Analysis

Beyond performance comparison, we also conduct several experiments on **VOS**, the largest one of the three datasets, to find out how the proposed approach works in segmenting primary video objects.

Performance of CCNN branches. We conduct several experiments to assess both CCNN branches. As shown in Table 2, both CCNN branches show impressive performance in predicting primary video objects when their pixel-wise predictions are directly evaluated as the other 6 deep models. Although the foreground/backgroundness conversion from pixel to superpixel may slightly decrease the precision and increase the recall, the overall precision increases by a large extent after the temporal propagation, at the cost of small decrease in recall. Considering that high precision is much more difficult to reach than high recall, such trade-off can bring increasing F_{β} and mIoU scores.

In particular, the performances of both the foregroundness and backgroundness branches outperform all the other 6 deep image-based salient object detection models on **VOS**. By analyzing the results, we find that this may be caused by two reasons: 1) using more training data, and 2) simultaneously handling complementary tasks. To further explore the reasons, we retrain CCNN on the same MSRA10K dataset used by most deep models. In this case, the F_{β} (mIoU) scores of the foregroundness and backgroundness maps predicted by CCNN will decrease to 0.747 (0.659) and 0.745 (0.658), respectively. Note that both branches still outperform **RFCN** on **VOS** in terms of mIoU (but F_{β} is slightly worse).

To further explore the performance gain by using two complementary branches, we conduct two more experiments. First, if we cut down the backgroundness branch and retrain only the foreground branch, the performance decreases by about 0.9%. Second, if we re-train a network with two foreground branches, the F_{β} and mIoU scores decrease from 0.806 to 0.800 and 0.710 to 0.700, respective-



Figure 5. Representative results of our approach. Red masks are the ground-truth and green contours are the segmented primary objects.



Figure 6. Influence of parameters k_0 and λ_c to our approach.

ly. These two experiments indicate that, beyond learning more weights, the background branch does have learned some useful cues that are ignored by the foreground branch, which are expected to be high-level visual patterns of typical background distractors. These results also validate the idea of training deep networks by simultaneously handling two complementary tasks. Effectiveness of neighborhood reversible flow. To prove the effectiveness of neighborhood reversible flow, we test our approach with two new settings. In the first setting, we replace the correspondence from Eq. (2) to the Cosine similarity between superpixels. In this case, the F_{β} and mIoU scores of our approach on **VOS** drop to 0.795 and 0.696, respectively. Such performance is still better than the initialized foregroundness maps but worse than the performance when using the neighborhood reversible flow (F_{β} =0.806, mIoU = 0.710). This result validates the effectiveness of neighborhood reversibility in temporal propagation.

In the second setting, we set $\lambda_c = +\infty$ in Eq. (5), implying that primary objects in a frame are solely determined by the foregroundness and backgroundness propagated from other frames. When the spatial predictions of each frame are actually ignored in the optimization process, the F_{β} (mIoU) scores of our approach on **VOS** only decrease from 0.806 (0.710) to 0.790 (0.693), respectively. This result proves that the inter-frame correspondences encoded in the neighborhood reversible flow are quite reliable for efficient and accurate propagation along the temporal axis.

Parameter setting. In this experiment we smoothly vary two key parameters used in our approach, including the k_0 in constructing neighborhood flow and the λ_c that controls

Table 3. Performance of our approach on **VOS** when using different color space in constructing neighborhood reversible flow.

Color Space	mAP	mAR	F_{β}	mIoU
RGB	.785	.862	.801	.703
Lab	.786	.860	.802	.702
HSV	.787	.866	.804	.707
RGB+Lab+HSV	.789	.870	.806	.710

the strength of temporal propagation. As shown in Fig. 6, larger k_0 tends to bring slightly better performance, while our approach performs the best when $\lambda_c = 0.5$. In experiments, we set $k_0 = 15$ and $\lambda_c = 0.5$ in constructing the neighborhood reversible flow.

Selection of color spaces. In constructing the flow, we represent each superpixel with three color spaces. As shown in Table 3, a single color space performs slightly worse than their combinations. Actually, using multiple color spaces have been proved to be useful in detecting salient objects [12], while multiple color spaces make it possible to assess temporal correspondences from several perspectives with a small growth in time cost. Therefore, we choose to use RG-B, Lab and HSV color spaces in characterizing a superpixel.

Speed test. We test the speed of the proposed approach with a 3.4GHz CPU (only use single thread) and a NVIDIA GTX 1080 GPU (without batch processing). The average time cost of each key step of our approach in processing 400×224 frames are shown in Table 4. Note that the majority of the implementation runs on the Matlab platform with several key steps written in C (e.g., superpixel segmentation and feature conversion between pixels and superpixels). We find that our approach takes only 0.20s to process a frame, which is much faster than many video-based models (e.g., 19.0s for NLC, 6.1s for ACO, 5.8s for FST, 5.4s for SAG and 4.7s for GF). This may be caused by the fact that we only build correspondences on superpixels with the neighborhood reversibility, which is very efficient. Moreover, we avoid using complex optimization objectives and constraints. Instead, we use only simple quadratic optimization objectives so as to obtain analytic solutions. The high efficiency of our approach makes it possible to be used in some real-world applications.

Failure cases. Beyond the successful cases, we also show in Fig. 7 some failures. We find that failures can be caused by the way of defining primary objects. For example, the salient hand and caption in Fig. 7 (a)-(b) are not labeled as primary objects as the corresponding videos from **Youtube-Objects** are tagged with "dog" and "cow," respectively. Moreover, shadow (Fig. 7 (c)), reflection (Fig. 7 (d)) and regions similar to background (Fig. 7 (e)) are some other reasons that may cause unexpected failures.



Figure 7. Failure cases of our approach. Rows from top to bottom: video frames, ground-truth masks and our results.

TT 1 1 4 C	1 01		•	1
lable 4. S	preed of 1	kev steps	in our a	approach
	r	Join Party and Party		rrr

Key Step	Speed (s/frame)			
Initialization	0.07			
Superpixel & Feature	0.08			
Build Flow & Propagation	0.04			
Primary Object Seg.	0.01			
Total	~ 0.20			

5. Conclusion

In this paper, we propose a simple yet effective approach for primary video object segmentation. With Deep Neural Networks pre-trained on massive images from salient object datasets to handle complementary tasks, the foregroundness and backgroundness in a video frame can be effectively predicted from the spatial perspective. After that, such spatial predictions are efficiently propagated via the inter-frame flow that has the characteristic of neighborhood reversibility. In this manner, primary objects in different frames can gradually pop-out, while various types of distractors can be well suppressed. Extensive experiments on three video datasets have validated the effectiveness of the proposed approach.

In the future work, we tend to improve the proposed approach by fusing multiple ways of defining primary video objects like motion patterns, semantic tags and human visual attention. Moreover, we will try to develop a completely end-to-end spatiotemporal model for primary video object segmentation by incorporating the recursive mechanism.

Acknowledgement. This work was partially supported by National Natural Science Foundation of China (61672072, 61532003 and 61325011), and Fundamental Research Funds for the Central Universities.

References

- R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 2012.
- [2] W. C. Chiu and M. Fritz. Multi-class video co-segmentation with a generative multi-video model. In *CVPR*, 2013.

- [3] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.
- [4] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014.
- [5] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: Non-successive mask transfer and interpolation for video cutout. In SIGGRAPH ASIA, 2015.
- [6] K. Fragkiadaki, G. Zhang, and J. Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.
- [7] H. Fu, D. Xu, B. Zhang, S. Lin, and R. K. Ward. Objectbased multiple foreground video co-segmentation via multistate selection graph. *IEEE TIP*, 24(11):3415–3424, 2015.
- [8] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu. Background prior-based salient object detection via deep reconstruction residual. *IEEE TCSVT*, 25(8):1309–1321, 2015.
- [9] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In ECCV, 2014.
- [10] W.-D. Jang, C. Lee, and C.-S. Kim. Primary object segmentation in videos via alternate convex optimization of foreground and background distributions. In *CVPR*, 2016.
- [11] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE TPAMI*, 2010.
- [12] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013.
- [13] G. Lee, Y.-W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, 2016.
- [14] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.
- [15] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In CVPR, 2015.
- [16] G. Li and Y. Yu. Deep contrast learning for salient object detection. In CVPR, 2016.
- [17] J. Li, C. Xia, and X. Chen. A benchmark dataset and saliency-guided stacked autoencoder for video-based salient object detection. arXiv:1611.00135, 2016.
- [18] X. R. Lijun Wang, Huchuan Lu and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, 2015.
- [19] N. Maerki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In CVPR, 2016.
- [20] MSRA10K. http://mmcheng.net/gsal/.
- [21] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE TPAMI*, 2014.
- [22] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.
- [23] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank. Salient object detection via structured matrix decomposition. *IEEE TPAMI*, 2016.
- [24] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

- [25] Y. Qin, H. Lu, Y. Xu, and H. Wang. Saliency detection via cellular automata. In CVPR, 2015.
- [26] S. A. Ramakanth and R. V. Babu. Seamseg: Video object segmentation using patch seams. In CVPR, 2014.
- [27] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev. Instance-level video segmentation from object tracks. In *CVPR*, 2016.
- [28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [29] N. Tong, H. Lu, X. Ruan, and M.-H. Yang. Salient object detection via bootstrap learning. In CVPR, 2015.
- [30] D. Tsai, M. Flagg, and J. M.Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, 2010.
- [31] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic cosegmentation in videos. In ECCV, 2016.
- [32] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien. Real-time salient object detection with a minimum spanning tree. In *CVPR*, 2016.
- [33] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, 2016.
- [34] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In CVPR, 2015.
- [35] W. Wang, J. Shen, and L. Shao. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE TIP*, 24(11):4185–4196, 2015.
- [36] F. Xiao and Y. J. Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016.
- [37] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, 2013.
- [38] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013.
- [39] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.
- [40] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In CVPR, 2013.
- [41] D. Zhang, O. Javed, and M. Shah. Video object cosegmentation by regulated maximum weight cliques. In EC-CV, 2014.
- [42] D. Zhang, D. Meng, and J. Han. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE TPAMI*, 2016.
- [43] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Minimum barrier salient object detection at 80 fps. In *ICCV*, 2015.
- [44] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia. Semantic object segmentation via detection in weakly labeled video. In *CVPR*, 2015.
- [45] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, 2015.
- [46] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In CVPR, 2014.