Deeply Supervised 3D Recurrent FCN for Salient Object Detection in Videos

Trung-Nghia Le
ltnghia@nii.ac.jp

Akihiro Sugimoto
sugimoto@nii.ac.jp

Department of Informatics
The Graduate University for Advanced Studies (SOKENDAI), Tokyo, Japan
National Institute of Informatics
Tokyo, Japan.

Abstract

This paper presents a novel end-to-end 3D fully convolutional network for salient object detection in videos. The proposed network uses 3D filters in the spatiotemporal domain to directly learn both spatial and temporal information to have 3D deep features, and transfers the 3D deep features to pixel-level saliency prediction, outputting saliency voxels. In our network, we combine the refinement at each layer and deep supervision to efficiently and accurately detect salient object boundaries. The refinement module recurrently enhances to learn contextual information into the feature map. Applying deeply-supervised learning to hidden layers, on the other hand, improves details of the intermediate saliency voxel, and thus the saliency voxel is refined progressively to become finer and finer. Intensive experiments using publicly available benchmark datasets confirm that our network outperforms state-of-the-art methods. The proposed saliency model also effectively works for video object segmentation.

1 Introduction

Salient object detection (SOD) is useful for high-level tasks in computer vision such as action recognition, video re-targeting, and video captioning. Recent advances using deep learning have demonstrated the superior performance in SOD[17, 20, 31]. Existing methods for SOD, however, employ image-based deep models and apply them frame-by-frame to compute saliency, resulting in failure of fully capturing dynamics of moving objects. This is because deep features used there do not exploit temporal information over frames, which is crucial in dealing with videos. Effectively integrating spatiotemporal information into deep models to compute 3D deep features for SOD is still open to challenge.

To learn deep features of a video, 2D Convolutional Neural Network (CNN) treats multiple frames as different channels and learns them by 2D convolution kernels, while 3D CNN learns video frames directly by 3D convolution kernels[29]. The 3D CNN has better ability to model both appearance and motion than the 2D CNN; thus the 3D CNN is well-suited for video representation. Indeed, detected 3D deep features are generic and compact[29], and they have been applied to some video analysis tasks. Particularly, 3D deep features learned using 3D CNNs show good performance on action recognition[29], video classification[7], video captioning[35], and visual attention[7]. 3D deep features, however, have not yet been exploited for SOD.

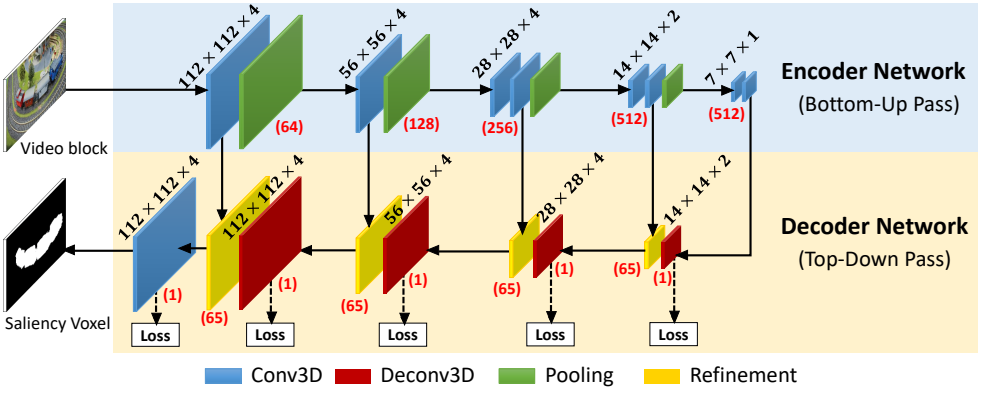


Figure 1: The architecture of the proposed DSRFCN3D model for an input video block with the resolution of $112 \times 112 \times 4$. The number in black outside a layer represents the size of output signals of that layer, and the number in red at the bottom represents the number of learning filters used in the layer.

For the recognition task, the feature map obtained from the last 3D convolution layer of the 3D CNN is fed into some fully connected layers to obtain a good deep feature. However, this feature map is not suitable for the pixel-to-pixel prediction because its resolution is much lower than that of the video block input. It has to be thus transferred to the pixel-level domain. Similarly to image based deep models[26], upsampling layers are attached into the 3D CNN to reconstruct object boundaries at the original input resolution, which forms a 3D Fully Convolutional Network (FCN)[30]. The 3D FCN consists of two parts: the deep feature computation from the input through the encoder network as the bottom-up pass, and the task-based prediction using the deep feature through the decoder network as the top-down pass. The 3D FCN indeed demonstrates effective end-to-end prediction on video analysis problems such as video semantic segmentation, optical flow estimation, and video coloring[30]. To improve the performance further, the skip-connection architecture is also used for the 3D FCN. This architecture enables to merge spatial and temporal information from low-level features in encoder network layers and high-level knowledge in decoder network layers into higher layers. However, the skip-connection still have not fully succeeded in demonstrating details of an object shape and contour. A refinement mechanism in cooperation with the skip-connection is thus required for learning contextual information detailing object shape and contour.

Motivated by observation above, we propose an end-to-end Deeply Supervised 3D Recurrent Fully Convolutional Network (DSRFCN3D) for SOD in videos (Fig. 1). Our DSRFCN3D consists of an encoder network, a decoder network, and a refinement. The encoder network extracts a 3D deep feature, which involves both spatial and temporal information, from an input video block. The decoder network computes the accurate saliency voxel from the 3D deep feature by progressively improving the intermediate saliency voxel through supervised learning at every hidden 3D deconvolution layer[13]. Here the saliency voxel is the block with the same size as the input video block in which pixel-level saliency values are computed at each frame in the block. The refinement, on the other hand, enriches the feature map at each layer by combining skip-connection between the bottom-up and the top-down passes and enhancement of contextual information through the recurrent convolutional layer.

With these functions of DSRFCN3D, object shape appearance from encoder network layers and semantic information from decoder network layers are effectively integrated into a feature map in terms of 3D deep features, which is then boosted up to finer and finer saliency voxel prediction. Accordingly, our proposed network generates high-quality saliency voxels and efficiently presents the boundaries of salient objects. Extensive experiments on several challenging video saliency benchmark datasets show that DSRFCN3D outperforms the state-of-the-arts on the SOD problem. We also applied our saliency model to video object segmentation (VOS) to demonstrate how our model works even for other related tasks.

2 Proposed Method

The input of our network is a video block with size $H \times W \times L$, where H, W, and L are the frame height, the frame width, and the temporal length (i.e., the number of frames), respectively. Each frame of a video is resized into size 112×112, and the resized video is first divided into 4-frame blocks, and then fed to DSRFCN3D to output saliency voxels consisting of the 4-frame blocks with the same size as the input (i.e., 112×112) in which saliency value at the pixel level is predicted at each frame.

2.1 Network Architecture

Figure 1 illustrates the detail configuration of DSRFCN3D. DSRFCN3D is composed of an encoder network, a decoder network, and a refinement. The encoder network is adapted from the C3D network[29], and we develop our decoder network using the idea of the 3D deconvolutional network and deeply-supervised network[15].

Encoder Network:

We develop our encoder network using the C3D network[29]. Namely, we use only 3D convolution layers of five scale levels and remove all the fully connected layers. In our encoder network, all the 3D convolution layers use filters with the size of $3 \times 3 \times 3$, the stride of $1 \times 1 \times 1$, and the padding of $1 \times 1 \times 1$, followed by the Rectified Linear Unit (ReLU) activation layer[13]. The first two pooling layers have filters with the size of $2 \times 2 \times 1$ and the stride of $2 \times 2 \times 1$, while the last two pooling layers use $2 \times 2 \times 2$ filters with the stride of $2 \times 2 \times 2$.

Decoder Network:

Our decoder network has four deconvolution layers and one convolution layer. The first two 3D deconvolution layers use filters with the size of $4 \times 4 \times 4$, the stride of $2 \times 2 \times 2$, and the padding of $1 \times 1 \times 1$. The last two 3D deconvolution layers have filters with the size of $4 \times 4 \times 1$, the stride of $2 \times 2 \times 1$, and the padding of $1 \times 1 \times 0$.

We enforce early supervision on all the hidden 3D deconvolution layers to improve the network parameter learning stage. To do so, we employ the deeply supervised learning scheme proposed in [15], which supervises both the hidden layers and the output layer to alleviate the problem of vanishing gradients during training.

The last 3D convolution layer serves as a predictor where we use the filter with the size of $3 \times 3 \times 3$, the stride of $1 \times 1 \times 1$, and the padding of $1 \times 1 \times 1$.

We define the loss function of DSRFCN3D as follows:

$$\mathcal{L}(\theta, w) = \ell_{predict}\left(\theta, w_{predict}\right) + \sum_{m=1}^{M} \ell_{deconv3D}\left(\theta, w_{deconv3D}^{(m)}\right), \tag{1}$$

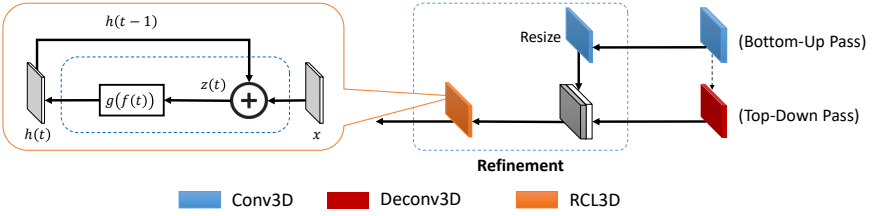


Figure 2: The detailed framework of our introduced refinement. The 3D convolution layer in the skip-connection uses the filter with the size of $3 \times 3 \times 3$ and the stride of $1 \times 1 \times 1$. The left block is the structure of the 3D Recurrent Convolution Layer (RCL3D) unit.

where M denotes the number of 3D deconvolution layers (in our case, $M = 4$), θ is the collection of all network layer parameters, and w denotes weights of the corresponding layer. ℓ is the binary cross-entropy loss, which can balance positive/negative classes between the predicted saliency voxel $\widehat{S} = (\widehat{S}_i)$ and its corresponding ground truth $S = (S_i)$:

$$\ell = -\frac{1}{N}\sum_{i=1}^{N}\left\{S_i \log \delta\left(\widehat{S}_i\right) + (1 - S_i)\log \delta\left(1 - \widehat{S}_i\right)\right\}, \tag{2}$$

where $N = H \times W \times L$ denotes the size of a video block. δ is the element-wise sigmoid function.

Refinement:

Our refinement has as its input a saliency voxel in a deconvolution layer and its corresponding convolution layer in the bottom-up pass. It integrates semantic information from coarser-level layers to refine the saliency prediction. Its detail is explained below.

2.2 3D Recurrent Convolution Layer based Refinement

Our refinement is a combination of the top-down skip connection[23] and our proposed 3D Recurrent Convolution Layer (RCL3D) (Fig. 2). It is capable of efficiently generating the high-fidelity prediction.

The skip-connection incorporates object appearance information in the bottom-up pass and semantic information in the top-down pass to generate a powerful feature map for saliency prediction. Namely, the feature map at a deconvolution layer in the top-down pass and the feature map in its corresponding convolution layer in the bottom-up pass are concatenated across channel dimension C to generate a stronger feature map. After this, the contextual information is enhanced and accumulated in the feature map through iterating the RCL3D unit. Note that before the concatenation, a 3D convolution layer with the filter of size of $3 \times 3 \times 3$, the stride of $1 \times 1 \times 1$, and the padding of $1 \times 1 \times 1$ is applied to the bottom-up feature map to reduce its size in our implementation (cf. Fig. 2).

3D Recurrent Convolution Layer (RCL3D):

We first define notations describing the RCL3D. Given a feature map $x \in \mathbb{R}^{H \times W \times L \times C}$ with the first two elements corresponding to two spatial coordinates (i.e., vertical and horizontal dimensions), the third denoting the temporal length (the number of frames), and the fourth being associated with the channel-index (or called filter-index), respectively, each x_{ijlc} with $1 \le i \le \mathrm{H}$, $1 \le j \le \mathrm{W}$, $1 \le l \le \mathrm{L}$, and $1 \le c \le \mathrm{C}$ is a 4D tensor.

Table 1: Number of videos used in our experiments.

Dataset	10-Clips[9]	SegTrack2[16]	DAVIS[22]	Total
Training	6	8	30	44
Testing	4	6	20	30

Table 2: Compared state-of-the-art methods and their classification.

target	hand-crafted method	deep learning
image		DCL[17], DHS[20], RFCN[31]
video	LGFOGR[33], RST[14], SAG[32], STS[36]	None

Our RCL3D is an extension of RCL[19] to the spatiotemporal domain so that it becomes a 3D filter. The left block in Fig. 2 depicts the structure of the RCL3D unit where $z(t)$ is the intermediate state at time t, and $g(f(t))$ is the dynamic behavior function at time t whose output is recurrently used. The RCL3D is iterated with T time steps. We set $T = 3$ as in [20][19].

The behavior of the RCL3D unit can be formally described as follows. Given a 4D tensor x_{ijlc} for the c-th channel of a unit located at position (i, j, l), x_{ijlc} and the state h_{ijlc} of the RCL3D unit at time $t-1$ (the recurrent input at time $t-1$) are combined to generate z_{ijlc} at time t:

$$z_{ijlc}(t) = \left(W_c^f\right)^\top x_{ijl} + \left(W_c^r\right)^\top h_{ijl}(t-1) + b_c, \tag{3}$$

where x_{ijl} denotes the feed-forward input from the layer in concern. W_c^f, W_c^r, and b_c denote the feed-forward weights, the recurrent weights and the bias, respectively. W_c^f, W_c^r, and b_c are implemented using two 3D convolution layers with $3 \times 3 \times 3$ filters.

The state of the RCL3D unit at time t is given by the dynamic behavior function:

$$h_{ijlc}(t) = g\left(f\left(z_{ijlc}(t)\right)\right), \tag{4}$$

where f is the ReLU activation function[13]: $f_{ijlc}(t) = f\left(z_{ijlc}(t)\right) = \max\left(z_{ijlc}(t), 0\right)$, and g is the local response normalization function[13] which prevents the states from exploding:

$$g\left(f_{ijlc}(t)\right) = \frac{f_{ijlc}(t)}{\left(1 + \frac{\alpha}{P} \sum_{c'=\max(0,c-P/2)}^{\min(C,c+P/2)} f_{ijlc'}^2(t)\right)^\beta}, \tag{5}$$

where P denotes the size of the local neighbor channels involved in the normalization. In our experiments, we set parameters $\alpha = 0.001$ and $\beta = 0.75$, similarly to [20].

3 Experimental Results

3.1 Benchmark Datasets and Evaluation Criteria

We evaluated the performance of our method on three public benchmark datasets: 10-Clips dataset[9], SegTrack2 dataset[16], and DAVIS dataset[22]. All the datasets contain manually annotated pixel-wise ground-truth for every frame.

We evaluated the performance using **F-measure**[1], and Mean Absolute Error (**MAE**). F-measure, which is computed based on the overlapping area between obtained results and

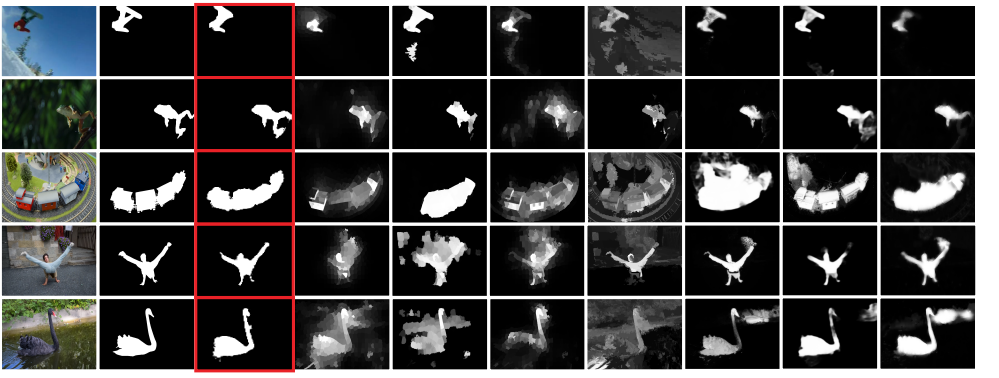


Figure 3: Visual comparison of our method against the state-of-the-art methods. From top-left to bottom-right, original image and ground-truth are followed by outputs obtained using our method, LGFOGR[33], RST[14], SAG[32], STS[36], DCL[17], DHS[20], and RFCN[31] in this order. Our method surrounded with red rectangles achieves the best results.

ground-truth, is a balanced measurement between precision and recall as follows: $F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 \times Precision + Recall}$. We remark that we set $\beta^2 = 0.3$ as used in [1] so that precision is more considered. MAE, on the other hand, is the average over the frame of pixel-wise absolute differences between the ground truth and obtained saliency voxels.

We compute precision and recall at each frame in a video and then compute the average over the video. Next, the mean of the averages over videos in a dataset is computed. F-measure is computed from the final precision and recall. MAEs are computed in the same way.

When binarizing results for the comparison with the ground truth, we used **F-Adap**[10], an adaptive threshold $\theta = \mu + \eta$ where μ and η are the mean value and the standard deviation of the saliency voxel. We also used **F-Max**[3], which describes the maximum of F-measure scores for different thresholds from 0 to 255.

3.2 Implementation and Training

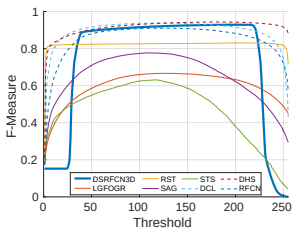
Implementation platform:

We implement our method with C/C++, using Caffe[11] implementation of the C3D network[29]. All experiments were conducted on a computer with a Core i7 3.6 GHz processor, 32 GB of RAM, and GTX 1080 GPU. The average time for processing a video with 100 frames was about 1.13 seconds.

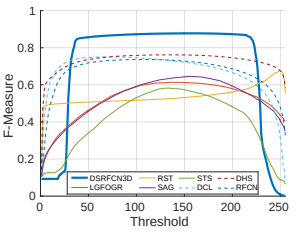
Training models:

We first built a pre-trained model on pseudo-videos to learn generic semantic information of plenty of categories and then fine-tuned it on video datasets to focus on dynamics.

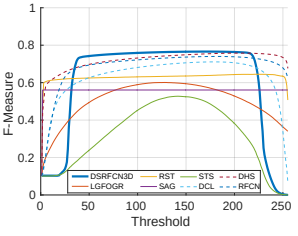
To generate pseudo-videos, we mixed four image datasets MSRA10k[5], ECSSD[27], HKU-IS[18], and THUR15k[5], resulting in 16,000 images in total, and then converted each image to a pseudo-video as follows. We used a window to crop an image and then randomly slide the window across the image in the diagonal/off-diagonal direction to have more 3 crops from the image. Next, we aggregated the 4 clops into a 4-frame long pseudo-video.



(a) 10-Clips Dataset



(b) SegTrack2 Dataset



(c) DAVIS Dataset

Figure 4: Quantitative comparison with state-of-the-art methods on three benchmark datasets, using F-measure with different thresholds. Our method is denoted by DSRFCN3D (thick blue).

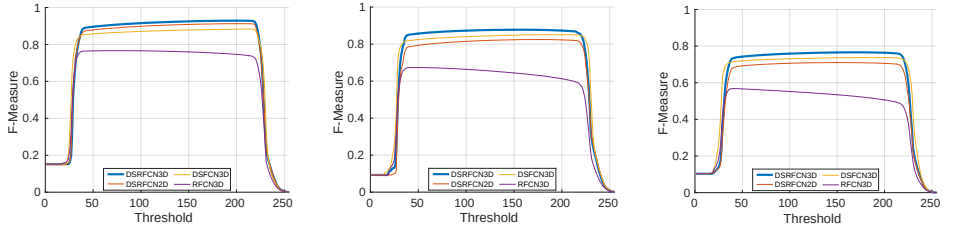
Table 3: Quantitative comparison with state-of-the-art methods on three datasets, using F-Adap, F-Max (higher is better), and MAE (smaller is better). The best three results are shown in **blue**, **green**, and **red**, respectively. DSRFCN3D is marked as **boldfaced**.

Dataset	10-Clips			SegTrack2			DAVIS		
Metric	F-Adap ⇑	F-Max ⇑	MAE ⇓	F-Adap ⇑	F-Max ⇑	MAE ⇓	F-Adap ⇑	F-Max ⇑	MAE ⇓
DSRFCN3D	**0.917**	**0.929**	**0.021**	**0.869**	**0.878**	**0.021**	**0.756**	**0.766**	**0.036**
LGFOGR[33]	0.629	0.667	0.207	0.500	0.614	0.117	0.537	0.601	0.102
RST[27]	0.827	0.831	0.055	0.510	0.677	0.125	0.627	0.645	0.077
SAG[25]	0.755	0.777	0.117	0.504	0.646	0.106	0.494	0.548	0.103
STS[34]	0.591	0.631	0.177	0.471	0.583	0.147	0.379	0.527	0.183
DCL[18]	**0.935**	**0.937**	**0.031**	**0.734**	**0.750**	**0.060**	0.664	0.711	**0.067**
DHS[20]	**0.923**	**0.947**	**0.022**	**0.733**	**0.762**	**0.050**	**0.715**	**0.758**	**0.048**
RFCN[31]	0.901	0.910	0.046	0.716	0.737	0.062	**0.710**	**0.740**	**0.067**

From each video dataset except for the DAVIS dataset, we chose randomly 60% videos and mixed them into a larger dataset for training while the remaining videos were used for testing each dataset (Table 1). This approach enables the trained model not to over-fit to a specific small dataset. We remark that for the DAVIS dataset, we used the training set and the testing set as in the DAVIS Benchmark[22]. We thus used 44 videos for training. We split each video in the 44 videos into overlapping 4-frame blocks with stride 1, producing in total about 3200 training samples. We note that we split each video in the test set into non-overlapping 4-frame blocks with stride 4.

During the training process, a simple data augmentation technique through mirroring and cropping was employed to avoid the problem of over-fitting. Similarly to [29], video blocks are first rescaled to resolution 128×171 and then cropped to 112×112 at a random position and horizontally flipped randomly.

To train our models, we used the Adam optimizer[12] with moments $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a weight decay of 0.0002. The size of each mini-batch is 1 due to our GPU memory constraints. We first constructed our pre-trained model from the scratch by training the pseudo-videos with 500k iterations. A base learning rate of 0.0005 with a 1/10th slow-down every 100k iterations was used. Then the pre-trained model was fine-tuned on the video dataset with 300k iterations. The learning rate was initially set to 10^{-5} and divided by 10 at every 100k iterations.



(a) 10-Clips Dataset (b) SegTrack2 Dataset (c) DAVIS Dataset

Figure 5: Comparison with the baseline methods using F-measure. Our (complete) method, denoted by DSRFCN3D, is marked in thick blue.

Table 4: Quantitative evaluation of components using F-Adap and MAE. The best results are shown in blue (higher is better for F-Adap and lower is better for MAE). Our (complete) method, denoted by DSRFCN3D, is marked as **boldfaced**.

setting description	3D filter	RCL	Deep supervision	10-Clips		SegTrack2		DAVIS	
				F-Adap ⇑	MAE ⇓	F-Adap ⇑	MAE ⇓	F-Adap ⇑	MAE ⇓
DSRFCN3D	∨	∨	∨	**0.917**	**0.021**	**0.869**	**0.021**	**0.756**	**0.036**
DSRFCN2D	×	∨	∨	0.900	0.028	0.811	0.037	0.711	0.044
DSFCN3D	∨	×	∨	0.872	0.030	0.834	0.025	0.729	0.040
RFCN3D	∨	∨	×	0.767	0.062	0.682	0.045	0.569	0.061

3.3 Comparison with the State-of-the-Arts

We compared the performance of our proposed method (denoted by DSRFCN3D) with recent state-of-the-art methods for SOD: LGFOGR[33], RST[14], SAG[32], STS[36], DCL[17], DHS[20], and RFCN[31] (cf. Table 2). We remark that we run original codes provided by the authors with recommended parameter settings. For the methods developed for the still image, we frame-wisely applied them to videos.

Figure 3 shows examples of obtained results. We qualitatively confirm that our method produces the best results on each dataset. Our method can handle complex foreground and background with different details, giving accurate and uniform saliency assignment.

Quantitative evaluations are shown in Fig.4 for F-measure and in Table 3 for F-Adap, F-Max, and MAE. Our method exhibits the best performance on all metrics on Segtrack2 and DAVIS datasets. Particularly, our method significantly outperforms the other methods on the Segtrack2 dataset. On the 10-Clips dataset, our method achieves the best performance in term of MAE and is the third best method in term of F-Measure, slightly lower than those of DCL[17] and DHS[20]. This can be understood that the performance on the 10-Clips dataset has been already saturated and that improving the performance is even more difficult.

3.4 Detailed Analysis of the Proposed Method

To demonstrate the effectiveness of using 3D deep feature (3D filter), RCL, and deep supervision, we compared DSRFCN3D with three baseline methods: deeply supervised 2D recurrent FCN (DSRFCN2D) totally based on 2D filters, deeply supervised 3D FCN (DSFCN3D) not using RCL3D, and recurrent 3D FCN (RFCN3D) without deeply supervised training. These baseline methods respectively evaluate effectiveness of our proposed 3D features, our designed network structure, and our designed training of the network. The baseline models were trained with the same settings as our DSRFCN3D.

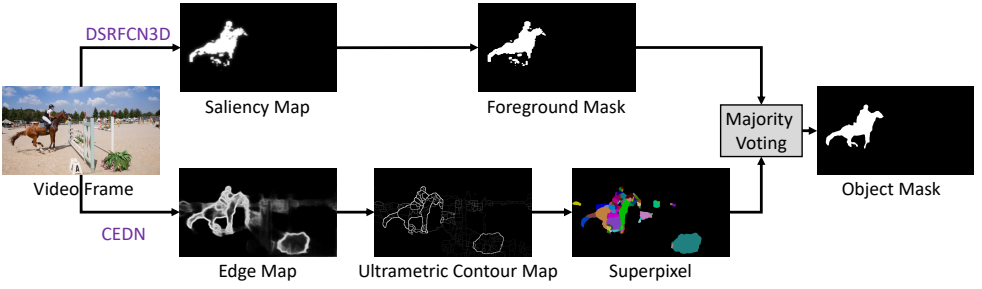


Figure 6: Boundary snapping[4] based video object segmentation framework using the saliency map.

Quantitative evaluation results are illustrated in Fig.5 and Table 4. They indicate that DSRFCN3D exhibits the best performance on all metrics on the three datasets. We see that (1) the 3D filter effectively works in videos than the 2D filter, that (2) the refinement based on RCL improves the performance of the network, and that (3) deep supervision effectively trains the network.

4 Application to Video Object Segmentation

Video object segmentation (VOS) is a binary labeling problem aiming to separate foreground objects from the background of a video[22], while salient object detection (SOD) aims to detect and segment salient objects in natural scenes. Although VOS and SOD are different tasks, SOD methods are beneficial for VOS when salient objects are main objects in scenes. In this section, we demonstrate the applicability of our proposed method to VOS.

Figure 6 illustrates the framework for VOS using the saliency map[4]. In the first branch, the output saliency map is binarized using the adaptive threshold mentioned in Section 3.1 to obtain the foreground mask. In the second branch, we implemented the object segmentation method based on boundary snapping[4]. We first detected contours of foreground objects using CEDN, the method by J.Yang *et al*. [34], and then applied the combinatorial grouping method[24] to compute the Ultrametric Contour Map (UCM)[24], which presents hierarchical segmentation. Superpixels were aligned by binarizing the UCM using threshold $\tau = 0.15$. From the foreground mask and superpixels, we performed majority voting to segment the object.

VOS methods are classified into two groups: one that is requiring the initial object mask at the first frame, and the other that is not. The former is called unsupervised while the latter is semi-supervised in the DAVIS Benchmark[22]. Since an initial object mask becomes a strong prior for accurately segmenting objects in subsequent frames, we chose most recent unsupervised methods for our comparison: CVOS[28], FST[21], and NLC[8]. Note that our method (denoted by DSRFCN3D*) does not require any initial object mask.

We tested all methods on the DAVIS dataset[22], the newest dataset for VOS, and evaluated results using measures in the 2017 DAVIS Challenge[25] (Region Similarity $\mathcal{J}$, Contour Accuracy $\mathcal{F}$, and Overall Performance $\mathcal{O}$). For a given error measure, we used three different statistics as done in [22]. They are the mean error, the object recall (measuring the fraction of sequences scoring higher than a threshold $\tau = 0.5$), and the decay (quantifying the performance loss (or gain) over time). Note that the results of the compared state-of-the-art VOS

Table 5: Quantitative comparison with state-of-the-art VOS techniques on DAVIS dataset. The best two results are shown in **blue**, **green**, respectively. Our methods is denoted by DSRFCN3D*.

Metric	Region Similarity ($\mathcal{J}$)			Contour Accuracy ($\mathcal{F}$)			Overall Performance ($\mathcal{O}$)
	Mean⇑	Recall⇑	Decay⇓	Mean⇑	Recall⇑	Decay⇓	Mean⇑
DSRFCN3D*	**0.651**	**0.782**	**0.000**	**0.609**	**0.726**	**0.027**	**0.630**
CVOS[28]	0.482	0.540	**0.105**	0.447	**0.526**	0.117	0.465
FST[21]	**0.558**	**0.649**	**0.000**	0.511	0.516	**0.029**	0.535
NLC[8]	0.551	0.558	0.126	**0.523**	0.519	0.114	**0.537**

Figure 7: Visual comparison of our method against the state-of-the-art VOS methods. From left to right, original video frame and ground-truth are followed by outputs obtained using our method, CVOS[28], FST[21], and NLC[8] in this order. Our method surrounded with red rectangles achieves the best results.

techniques are given in the DAVIS Benchmark[22].

Evaluation results are showed in Table 5, indicating that our proposed method outperforms the state-of-the-art methods. Our method exhibits the best performance on all metrics at all statistics. Figure 7 shows examples of obtained results.

5 Conclusion

We proposed an end-to-end 3D deep neural network DSRFCN3D for SOD in videos. DSRFCN3D works in the spatiotemporal domain to detect 3D deep features, and possesses the refinement mechanisms at each layer to strengthen features using those in the top-down pass, which is in turn trained with supervision for progressively improving details of intermediate saliency. Experimental results on standard benchmark datasets demonstrate that DSRFCN3D outperforms state-of-the-art methods.

We used 4-frames as the size of each video block in the experiments. This comes only from the available memory size of GPU in our used PC. Investigating the appropriate size of the video block with which DSRFCN3D performs best is left for future work. Exploiting the network structure further to reduce the number of parameters and to utilize different sizes of video blocks is also left for future work.

Acknowledgement

This work is in part supported by JST CREST (Grant No. JPMJCR14D1) and by Grant-in-Aid for Scientific Research (Grant No. 16H02851) of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1597–1604. IEEE, June 2009.

[2] Loris Bazzani, Hugo Larochelle, and Lorenzo Torresani. Recurrent mixture density network for spatiotemporal visual attention. In *International Conference on Learning Representations (ICLR)*, 2017.

[3] A. Borji, M. M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, Dec 2015.

[4] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] M. M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, March 2015. ISSN 0162-8828. doi: 10.1109/TPAMI.2014.2345401.

[6] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Salientshape: group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014. ISSN 1432-2315.

[7] Ali Diba, Ali Pazandeh, and Luc Van Gool. Efficient two-stream motion and appearance 3d cnns for video classification. In *Workshop on brave new ideas for motion representations in videos*, pages 1–4, Oct 2016.

[8] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. *The British Machine Vision Conference(BMVC)*, 2(7):8, 2014.

[9] Ken Fukuchi, K. Miyazato, A. Kimura, S. Takagi, and J. Yamato. Saliency-based video segmentation with graph cuts and sequentially updated priors. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 638–641. IEEE, June 2009.

[10] Yangqing Jia and Mei Han. Category-independent object-level saliency detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1761–1768. IEEE, Dec 2013.

[11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3063-3.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] Trung-Nghia Le and Akihiro Sugimoto. Contrast based hierarchical spatial-temporal saliency for video. In *Image and Video Technology - 7th Pacific-Rim Symposium, PSIVT 2015, Auckland, New Zealand, November 25-27, 2015, Revised Selected Papers*, volume 9431 of *Lecture Notes in Computer Science*, pages 734–748. Springer International Publishing Switzerland, 2015. ISBN 978-3-319-29451-3.

[15] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics (AISTATS)*, pages 562–570, 2015.

[16] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *2013 IEEE International Conference on Computer Vision*, pages 2192–2199, Dec 2013.

[17] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 478–487, June 2016.

[18] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 5455–5463, June 2015.

[19] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3367–3375, June 2015.

[20] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 678–686, June 2016.

[21] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1777–1784. IEEE, Dec 2013.

[22] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, June 2016.

[23] Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. *Learning to Refine Object Segments*, pages 75–91. Springer International Publishing, 2016. ISBN 978-3-319-46448-0.

[24] J. Pont-Tuset, P. ArbelÃąez, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):128–140, Jan 2017. ISSN 0162-8828.

[25] 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.

[26] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99): 1–1, 2016. ISSN 0162-8828.

[27] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, April 2016. ISSN 0162-8828.

[28] B. Taylor, V. Karasev, and S. Soattoc. Causal video object segmentation from persistence of occlusions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4268–4276, June 2015.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, Dec 2015.

[30] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Deep end2end voxel2voxel prediction. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 402–409, June 2016.

[31] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. *Saliency Detection with Recurrent Fully Convolutional Networks*, pages 825–841. Springer International Publishing, 2016. ISBN 978-3-319-46493-0.

[32] Wenguan Wang, Jianbing Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3395–3402, June 2015.

[33] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. *Image Processing, IEEE Transactions on*, 24(11):4185–4196, Nov 2015. ISSN 1057-7149.

[34] J. Yang, B. Price, S. Cohen, H. Lee, and M. H. Yang. Object contour detection with a fully convolutional encoder-decoder network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 193–202, June 2016.

[35] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4584–4593, June 2016.

[36] Feng Zhou, Sing Bing Kang, and M.F. Cohen. Time-mapping using space-time saliency. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3358–3365, June 2014.