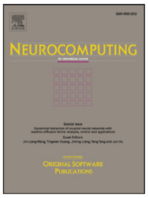
Neurocomputing 368 (2019) 180–187

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Constrained fixation point based segmentation via deep neural network

Gongyang Li[a,b], Zhi Liu[a,b,*], Ran Shi[c], Weijie Wei[a,b]

[a] Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China
[b] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
[c] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

ARTICLE INFO

Article history:
Received 29 April 2019
Revised 18 July 2019
Accepted 22 August 2019
Available online 28 August 2019

Communicated by Dr. Junwei Han

Keywords:
Human visual system
Object segmentation
Interactive image segmentation
Fixation point
Deep neural network

ABSTRACT

It is an explicit mode to use the clicking points by the mouse in the interactive image segmentation, while an implicit interaction mode is to use the fixation points from the eye-tracking device. Both modes can provide a series of points. Inspired by the similarity between these two interaction modes, we propose a novel human visual system (HVS) based neural network for transferring the constrained fixation point based segmentation to the clicking point based interactive segmentation. Briefly speaking, the sequence of information transmission and processing in our model is RGB image, VGG-16 backbone, LGN-like module (LGNL) and ConvLSTM block, which correspond to the pathway of stimulus transmission and processing, *i.e.* stimulus, retina, lateral geniculate nucleus (LGN) and visual cortex in the HVS. First, the RGB image is fed to the VGG-16 backbone to obtain the multiple-layer feature maps. Then the LGNL is adopted to effectively incorporate edge-aware features and semantic features from different layers of the VGG-16 backbone in multiple resolutions, so as to produce rich contextual features. Finally, with the guidance of the fixation density map transformed from the fixation points, the output feature maps of LGNL are utilized to generate the segmentation map via a stack of ConvLSTM blocks in a coarse-to-fine manner. Comprehensive experiments demonstrate that the proposed HVS based neural network achieves a higher segmentation performance and outperforms seven state-of-the-art methods, and prove that the transfer from constrained fixation points to clicking points is reasonable and valid.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Interactive image segmentation, which aims at segmenting the object according to the user's input, is a fundamental and challenging problem in the field of computer vision. It is well known that clicking points and drawing scribbles are two typical interaction modes, which require users to mark the image with the seed pixels of foreground and background. Numerous interactive segmentation methods based on clicking points and drawing scribbles have been proposed. Boykov and Jolly [1] proposed the interactive graph cuts for segmentation by solving a max-flow/min-cut based energy minimization problem. In [2], according to the connection between random walks on graphs and discrete potential theory, the probability that a random walker starting at each unlabeled pixel first reaches one of the pre-labeled pixels can be determined, and then the image segmentation is obtained by assigning each pixel to the label of the calculated greatest probability. In [3], Gulshan et al. proposed an interactive segmentation method by combining the geodesic distance information with the explicit edge information under the framework of graph cuts. Actually, different users have their own ways to click points or draw scribbles on the images. However, the segmentation methods [1–3] mentioned above are sensitive to the locations of inputs, so that they cannot accurately segment the same objects according to various inputs from different users.

Besides clicking points and drawing scribbles with the mouse, the fixation point is another interaction way. Both clicking points and fixation points provide a series of points. This means that the gap between fixation points and clicking points is tiny. In particular, Mishra et al. [4] reformulated the segmentation problem in conjunction with fixation points, and defined the segmentation problem as segmenting the region containing fixation points. In [4], the fixation points they used are marked artificially with a mouse, and all the fixation points are inside the object. Therefore, we can bridge the constrained fixation points that locate inside the objects with the clicking points, to derive the clicking point based interactive segmentation from the constrained fixation point based segmentation.

[*] Corresponding author at: School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China.
E-mail address: liuzhisjtu@163.com (Z. Liu).

https://doi.org/10.1016/j.neucom.2019.08.051
0925-2312/© 2019 Elsevier B.V. All rights reserved.

There are also some studies on the fixation point based segmentation/detection. Tian and Jung [5] proposed a fixation point based image segmentation method using the superpixel based random walk model. According to the analysis of gaze distribution, Shi et al. [6] proposed a gaze based object segmentation method. Some researchers also transformed the fixation points into a fixation map to study object segmentation/detection. In [7], Li et al. proposed a simple fixation map based model for salient object segmentation, in which a set of object candidates are ranked with a fixation map via a scoring function to produce the segmentation result. In [8], the fixation map is considered as a significant prior of salient objects. In [9], a multilayer graph-based two-stage saliency detection model by merging fixation map is proposed to detect salient objects in complex scenes. Wang et al. [10] proposed a unified neural network to infer salient objects from fixation map in a top-down manner.

We know that fixation points closely relate to the human visual system. In the previous studies, the fixation points can be used to serve as the seed points [5] or the object prior [8,9], to rank the object candidates [6,7], and to assist the segmentation process [10]. However, these methods [5–10] do not study the fixation point based segmentation from a physiological point of view. Thus, in this paper, we will the study fixation point based segmentation from the perspective of the human visual system, and transfer it to the clicking point based interactive segmentation via the constrained fixation points.

From the psychological aspect, the recent psychological studies [11–13] have shown that when viewing an identical scene, the visual systems of different individuals exhibit heterogeneous gaze patterns. In other words, when viewing the same object, the locations of fixation points of different individuals are different. Such a difference is related to age, race, gender and education of individual. Based on this study, Xu et al. [14] proposed a multi-task convolutional neural network (CNN) to perform the personalized saliency prediction, which is different from the universal saliency prediction models [15–20]. We adopt the personalized constrained fixation points to simulate the users' various inputs in the clicking point based interactive segmentation.

Recently, convolutional neural networks have made a significant progress in the field of computer vision, such as image recognition [21–23], semantic segmentation [24,25], video object segmentation [26] and saliency detection [27–29]. It is universally known that the computer vision system aims to replace the visual organs with various imaging systems, and the computer replaces the brain to complete the processing and interpretation of images and videos. The ultimate goal of computer vision is to enable computers to visually observe and understand the world like humans, and to adapt to the environment. Therefore, it is worth thinking about the process how brain processing visual information, and how to simulate this process with a computational model. Bell et al. [30] believed that the visual information processing in HVS can be divided into three parts: low-level, medium-level and high-level. The low-level processing extracts the physical characteristics of visual stimuli, such as brightness, boundary, color and motion. The medium-level processing involves the process of combining the information of these physical characteristics. The high-level processing involves segmenting, recognizing, classifying and understanding the objects. Considering that the pathway of visual information transmission and processing in HVS is stimulus, retina, lateral geniculate nucleus (LGN) and visual cortex in turn, we can simply distinguish that the retina is responsible for low-level processing, the LGN is in charge of medium-level processing as the relay station between retina and visual cortex, and the visual cortex manages high-level processing.

Inspired by the structure of HVS and the outstanding performance of CNN, we propose a novel deep neural network imitating the information processing pathway of HVS to segment objects with constrained fixation points. The pipeline of information transmission and processing in our network is RGB image, VGG-16 backbone [21], LGN-like module and ConvLSTM blocks in turn. The VGG-16 backbone processes the RGB image to generate the image-level features, which are rich in edge-aware information and semantic information. This backbone is treated as the low-level processing, and the medium-level processing is in the proposed LGN-like module. The LGN-like module receives feature maps of different layers from the backbone, and effectively incorporates these feature maps through multiple convolutional layers with different dilation rates to generate the contextual features. In the high-level processing, the integrated rich contextual features are concatenated with the fixation density map, which is transformed from the fixation points and used as the guidance, to segment and refine the objects in the ConvLSTM blocks. In this way, our network achieves a higher consistency with HVS. And the HVS based neural network can be transferred to the clicking point based interactive segmentation.

Overall, the main contributions of this paper are summarized as follows:

(1) We study the clicking point based interactive segmentation from a new perspective, and we transfer the constrained fixation point based segmentation to it.
(2) We propose a novel HVS based neural network to segment objects with constrained fixation points. Our model simulates the pathway of visual information transmission and processing in HVS, and experimental results demonstrate the superiority and effectiveness of our model.
(3) We propose a LGN-like module in our network for the aggregation and fusion of hierarchical features. The proposed module effectively incorporates edge-aware features in low layers and semantic features in high layers. Meanwhile, the integrated rich contextual features in the LGN-like module are complementary and robust for segmenting and refining the objects.

The remainder of this paper is organized as follows. The proposed HVS based model is detailed in Section 2. Experimental results and analysis are shown in Section 3, and the conclusion is drawn in Section 4.

2. The proposed model

In this section, we first describe the overall network architecture of our proposed model in Section 2.1. Then we present the fixation points transformation in Section 2.2. We give the detailed formulas of LGN-like module in Section 2.3. In the end, we describe the ConvLSTM based segmentation and refinement in Section 2.4.

2.1. Network architecture

In this paper, we propose a HVS based model to address constrained fixation point based segmentation, and transfer it to the clicking point based interactive segmentation. The overall architecture is shown in Fig. 1. Our network consists of three components: VGG-16 backbone [21], LGN-like module and ConvLSTM block. Corresponding to the HVS, the information processing in our model can also be divided into three levels: low-level, medium-level and high-level. Specifically, the VGG-16 backbone is in charge of low-level processing, the LGN-like module is responsible for medium-level processing, and the ConvLSTM block manages high-level processing. We first input the RGB image into the VGG-16 backbone to produce multi-layer feature maps, in which the feature maps in low layers are rich in edge-aware information and the feature maps in high layers capture abundant

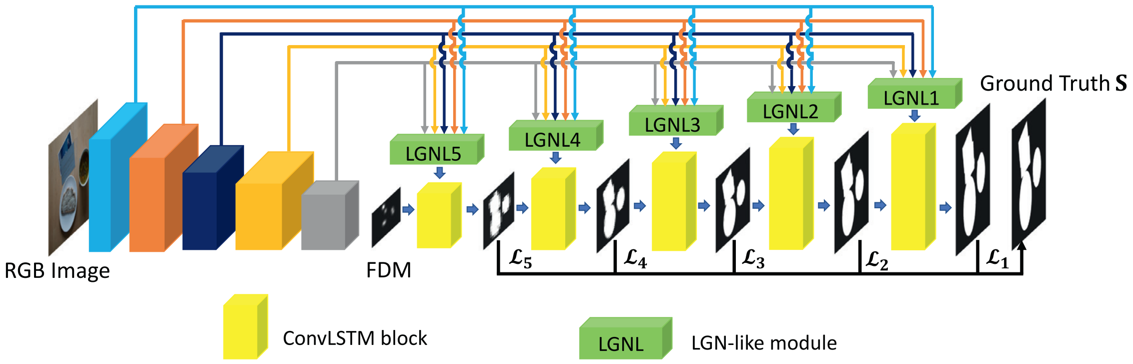


Fig. 1. The overall framework of our proposed model. The arrows between cuboids indicate the information stream. Given a RGB image (288 × 288 × 3), multi-level features are first extracted by the VGG-16 backbone [13], which is like the retina. Then hierarchical features aggregation and fusion is performed by LGNL. After that, the integrated rich contextual features are concatenated with fixation density map **FDM** (18 × 18 × 1) to generate the coarse segmentation map in the ConvLSTM block. Finally, the deep supervision is applied to improve the pixel-wise accuracy of segmentation. The final segmentation map is obtained by gradual refinements via a stack of ConvLSTM blocks. Please zoom-in for details.

semantic information. Then we propose the LGN-like module to aggregate and fuse the feature maps of all the five layers at five different resolutions. Just like the name of the LGN-like module, its function is to simply simulate the visual information processing in LGN. The LGN-like module is made up of convolutional layers with various fields of view so that it can capture multi-layer and multi-scale contextual information. In order to make full use of the fixation points, we transform the fixation points into a fixation density map by blurring with a Gaussian filter to obtain a good prior of the objects. And with the guidance of fixation density map, the first integrated rich contextual feature maps, *i.e.* the output of *LGNL*5, are fed into the ConvLSTM block to segment objects roughly. With the multi-resolution contextual feature maps, which are the output of other LGNLs, the coarse segmentation map is refined via a stack of ConvLSTM blocks in a coarse-to-fine manner to generate the final segmentation map. Notably, our network can be learned in an end-to-end way with deep supervision [31].

2.2. Fixation points transformation

Due to the special structure of human eyes, the HVS has a well-defined contrast sensitivity. Specifically, when we look at a point in an object, the resolution is higher as the distance from the fixation point is closer. It results in a concentric circle with the fixation point, so the center is clear, and the surroundings are dim. And thus, we can only clearly recognize the object in the center of the field of view. Therefore, in order to expand the receptive field of fixation points and accurately locate the position of the object, the fixation points are blurred with a Gaussian filter to produce a fixation density map.

The fixation points map **FP** can be generated based on the fixation points recorded by an eye tracker. According to **FP**, we compute fixation density map **FDM** as follows:

$$\mathbf{FDM} = Nor[\mathbf{FP} * \mathbf{G}(\sigma)], \tag{1}$$

where $Nor[\cdot]$ represents the min–max normalization. The notation '*' refers to the convolutional operator. $\mathbf{G}(\cdot)$ is a Gaussian filter, and its parameter σ determines the filter size and the standard deviation of it. The parameter σ is the visual angle in pixels, which is determined by several parameters when collecting the eye-tracking data, and it is defined as follows:

$$\sigma = \left[\frac{S_{img}}{2arctan\frac{S_{monitor}}{2D}} \right], \tag{2}$$

where $[\cdot]$ is the rounding operation. S_{img} is the diagonal length of the image, in pixels. $S_{monitor}$ is the diagonal length of the monitor, in inches, and D refers to the distance between the monitor and the eyes, also in inches.

2.3. LGN-like module

Contextual information is quite vital to assist object segmentation. Specifically, there are five-stage feature maps in VGG_16 backbone, *i.e.* conv1_2, conv2_2, conv3_3, conv4_3 and conv5_3. However, the existing image recognition methods [21,22] just use the feature map of the last layer (*i.e.* conv5_3) of the backbone network. Although the feature map of the last layer is rich in semantic information, it cannot capture the contextual information. A recent work [26], which aims to video object segmentation, proposes to up-sample the feature maps of the last four layers (conv2_2, onv3_3, conv4_3 and conv5_3) to the same size as that of the input frame, and then concatenates them to generate the segmentation result. In this fusion manner, the features from low layers and high layers combined to achieve the better segmentation results. Meanwhile, extracting the contextual features is a medium-level processing, which further processes the features of backbone, just like the function of LGN in the HVS. Inspired by Caelles et al. [26] and LGN, we propose a LGN-like module, which performs concatenation, dilated convolution and concatenation in turn, as shown in Fig. 2, to incorporate hierarchical features and learn multi-scale contextual information of objects in the image.

For the input image **I** with size $W \times H$, we first adopt the VGG-16 backbone to extract feature maps at five stages, *i.e.* conv1_2, conv2_2, conv3_3, conv4_3 and conv5_3, which are represented as $\mathbf{FM} = \{\mathbf{fm}_i, i = 1, \ldots\ldots, 5\}$ with resolution $\tau_i = [\frac{W}{2^{i-1}}, \frac{H}{2^{i-1}}]$, *i.e.* $[w_i, h_i]$. For the sake of concatenating these feature maps at the same resolution as that of $\mathbf{fm}_i$, we use the convolutional operation to shrink feature maps with a resolution larger than $\mathbf{fm}_i$, and the deconvolutional operation to expand feature maps with a resolution smaller than $\mathbf{fm}_i$, which is formulated as:

$$\mathbf{f}^{\tau_i} = \mathbf{Cat}(Conv(\mathbf{fm}_1; \psi_1), \ldots\ldots, Conv(\mathbf{fm}_i; \psi_i), \ldots\ldots, DeConv(\mathbf{fm}_5; \psi_5)), \tag{3}$$

where $Conv(\cdot)$ and $DeConv(\cdot)$ represent the convolutional operation and deconvolutional operation, respectively. The notation ψ_l is the kernel size of convolutional or deconvolutional operation, and $\psi_l = 2^{|i-l|} \times 2^{|i-l|}$, $l = 1, \ldots\ldots, 5$. **Cat** is the cross-channel concatenation. Whether it is the convolutional or the deconvolutional operation, the size of each output feature map is $w_i \times h_i \times 64$. And thus, the size of $\mathbf{f}^{\tau_i}$ is $w_i \times h_i \times 320$.

In order to simply simulate the processing of information by different cells in LGN, we use three convolutional layers with

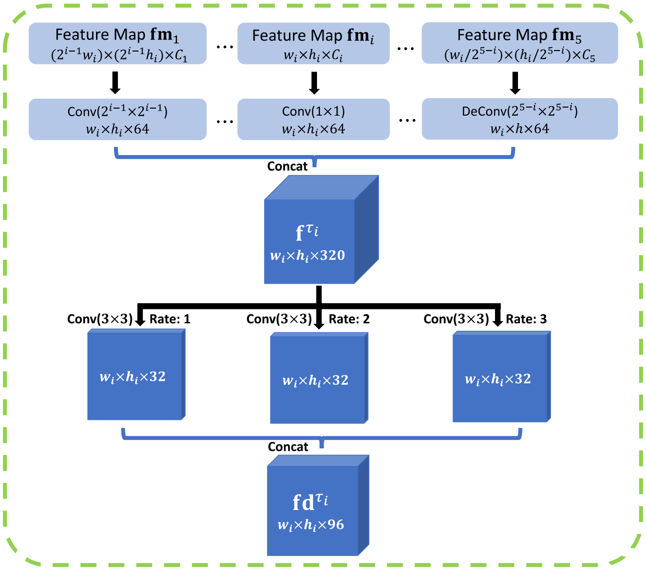


Fig. 2. Illustration of LGN-like module (LGNL). The LGNL first takes the five-layer feature maps with different resolutions and channels as the input. Then these feature maps will shrink or expand to the same resolution as that of the feature map $\mathbf{fm}_i$ and equal channels through convolutional or deconvolutional operation. The cross-channel concatenation is used to aggregate the five stacks of feature maps, namely $\mathbf{f}^{\tau_i}$. And the three convolutional layers with different dilation rates are adopted to capture multi-layer and multi-scale contextual information. Finally, the multiple contextual feature maps are concatenated in a cross-channel manner, called $\mathbf{fd}^{\tau_i}$. Please zoom-in for details.

different dilation rates to process $\mathbf{f}^{\tau_i}$. These three dilated convolutional layers [32] can enlarge the fields of view without the loss of resolution and the increase of computation. In particular, these three dilated convolutional layers have the same convolutional kernel size 3×3 with different dilation rates, which are set to 1, 2 and 3, respectively, to capture multi-layer and multi-scale contextual information. For computational efficiency, the number of the convolutional filters of each dilated convolutional layer is 32. Therefore, the size of the output feature map from each dilated convolutional layer is $w_i\times h_i\times32$. Finally, we combine the three feature maps by cross-channel concatenation to generate the multi-layer and multi-scale contextual feature map $\mathbf{fd}^{\tau_i}$, with a size of $w_i\times h_i\times96$.

2.4. ConvLSTM based segmentation and refinement

From the multi-resolution contextual features $\mathbf{FD}=\{\mathbf{fd}^{\tau_i}, i=1,\ldots\ldots,5\}$ and the fixation density map **FDM,** we expect our network to be able to infer precise segmentation map. Because ConvLSTM [33] has exhibited the ability to gradually refine the details of objects in salient object detection [10,34], we design a ConvLSTM block to infer the precise segmentation map. ConvLSTM is a variant of traditional fully connected LSTM [35] and introduces convolutional operation into input-to-state and state-to-state transitions. Therefore, ConvLSTM can preserve the spatial information of convolutional feature maps and optimize the details of the object well. Our ConvLSTM block consists of three ConvLSTM layers, and each convolutional operation in ConvLSTM layer with 32 filters with kernel size 3×3. Its function is to process the fine information similar to the visual cortex in HVS.

As aforementioned, we use the **FDM** to guide $\mathbf{fd}^{\tau_5}$ for segmenting objects coarsely via our ConvLSTM block. Since the resolution of **FDM** is 288×288, which is unmatched with that of $\mathbf{fd}^{\tau_5}$ (the resolution is 18×18), we down-sample the **FDM** to 18×18 by max pooling. The **FDM** and $\mathbf{fd}^{\tau_5}$ are concatenated to form the feature map **X** which is the input to the three ConvLSTM layers recurrently. We apply convolutional filter with kernel size 1×1 with Sigmoid and deconvolutional filter with kernel size 2×2 to the output of ConvLSTM – 3, *i.e.* $\mathbf{H}_3$, to get $\mathbf{sm}^5$ (the resolution is 36×36) as shown in Fig. 3. In particular, the channel number of **FDM** is 1. Then the coarse segmentation map $\mathbf{sm}^5$ will be refined progressively with the multi-resolution contextual features, which are the output of other LGN-like modules. Meanwhile, the resolution of segmentation map will increase gradually to the original size of the input image. The segmentation map $\mathbf{sm}^i$ $(i=4,\ldots\ldots,1)$ is generated with the same way as $\mathbf{sm}^5$, where the **FDM** is replaced by $\mathbf{sm}^{i+1}$. In particular, the deconvolutional filter is replaced by a convolutional filter with kernel size 1×1 for generating the final segmentation map $\mathbf{sm}^1$. In summary, the process of inferring the segmentation map $\mathbf{sm}^i$ is defined as follows:

$$\mathbf{sm}^i=\begin{cases}DeConv(Conv(CB(\mathbf{FDM},\mathbf{fd}^{\tau_i});\psi_6);\psi_7), i=5\\ DeConv\Big(Conv\Big(CB\Big(\mathbf{sm}^{i+1},\mathbf{fd}^{\tau_i}\Big);\psi_6\Big);\psi_7\Big), i=4,3,2,\\ Conv\Big(Conv\Big(CB\Big(\mathbf{sm}^{i+1},\mathbf{fd}^{\tau_i}\Big);\psi_6\Big);\psi_6\Big), i=1\end{cases} \quad (4)$$

where $CB(\cdot)$ is the operation of ConvLSTM block, $Conv(*;\ \psi_6)$ is the convolutional layer with kernel size ψ_6 *i.e.* 1×1 and $DeConv(*;\ \psi_7)$ is the deconvolutional layer with kernel size ψ_7 *i.e.* 2×2. Here, the sigmoid activation function is omitted for simplicity.

In order to improve the quality of the intermediate segmentation maps, *i.e.* $\mathbf{sm}^i$ $(i=5,\ldots\ldots,2)$, we adopt the deeply supervised learning mechanism [31]. The pixel-wise supervision information from the ground truth **S** will guide the segmentation at each level to make the details of $\mathbf{sm}^i$ much finer. Specifically, the intermediate segmentation map $\mathbf{sm}^i$ need to be resized to the same resolution as the ground truth **S**. And we adopt the classical cross-entropy loss function in this case, for each resized segmentation map $\mathbf{sm}^i$, the corresponding loss function is defined as follows:

$$\mathcal{L}_i\Big(\mathbf{S},\mathbf{sm}^i\Big)=-\sum_{x,y}\Big[S_{x,y}\log\Big(sm^i_{x,y}\Big)+(1-S_{x,y})\log\Big(1-sm^i_{x,y}\Big)\Big], \quad (5)$$

where $S_{x,\,y}\in\{0,1\}$ is the label of the pixel $(x,\,y)$ in the ground truth **S**, and $sm^i_{x,y}\in[0,\ 1]$ is the confidence score of the pixel $(x,\,y)$ belonging to the object. Thus, the final loss function $\mathcal{L}$ is defined as follows:

$$\mathcal{L}=\sum_{i=1}^{5}\mathcal{L}_i\Big(\mathbf{S},\mathbf{sm}^i\Big). \quad (6)$$

3. Experimental results

3.1. Experimental setup

Dataset: Currently, there are no datasets for fixation point based segmentation. So, we collect suitable data for the fixation point based segmentation from OSIE dataset [36], which is designed for fixation prediction. The OSIE dataset contains eye-tracking data from 15 participants for a full set of 700 observation images with a resolution of 800×600, so each image has 15 personalized fixation point maps. In addition, each observation image is manually segmented into a collection of objects on which semantic attributes are manually annotated. In order to obtain the constrained fixation points, which fall inside the objects, we exploit two criteria to screen 700 observation images, their corresponding 10,500 fixation point maps and semantic ground truths as follows:

First, on the basis of images, personalized fixation point maps and the corresponding semantic ground truths, we choose the fixation point maps and their semantic ground truths if all the fixation points locate in the objects.

Second, we transform the selected semantic ground truths into the ground truths of binary object segmentation, *i.e.* the selected

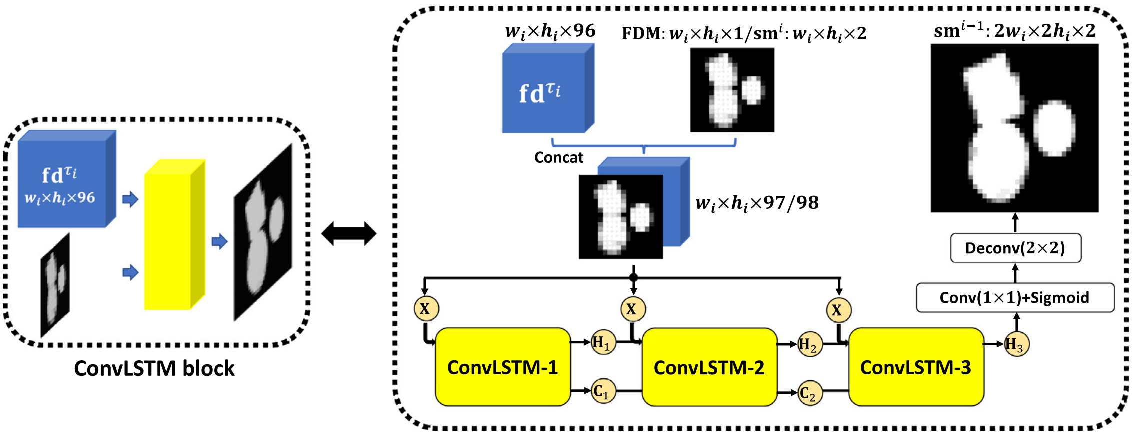


Fig. 3. Illustration of ConvLSTM block. The ConvLSTM block receives the $\mathbf{fd}^{\tau_i}$ and $\mathbf{FDM}/\mathbf{sm}^i$, and then the concatenated features **X** are transmitted to three ConvLSTM layers to generate the segmentation map $\mathbf{sm}^{i-1}$, which is more precise than $\mathbf{sm}^i$ and has a better segmentation quality. Please zoom-in for details.

semantic labels are set to the object label '1', and the background labels are set to '0'.

According to the two criteria, we finally select 3683 constrained fixation point maps and the corresponding ground truths of binary object segmentation to build a new dataset, OSIE-CFPS, for constrained fixation point based segmentation. According to the serial numbers of the observation images, we specify the fixation point maps and the binary ground truths corresponding to the first 600 observation images as the training set with 3075 samples, and those of the last 100 observation images as the testing set with 608 samples.

Following the parameters S_{img}, $S_{monitor}$ and D described in OSIE [36], we use Eq. (2) to calculate the visual angle σ, which is 24 pixels. Thus, we can transform the 3683 fixation point maps into the fixation density maps. So, one observation image, one fixation density map and its corresponding binary ground truth consist of a training triplet. To improve the varieties, we simply augment this dataset by mirror reflection and rotation (0°, 90°, 180° and 270°), producing 15,375 training triplets totally.

In particular, in order to evaluate the reasonableness and validity of the transfer from the constrained fixation point based segmentation to the clicking point based interactive segmentation using our model, we also measure our HVS based neural network on the GrabCut dataset [37], which is a common benchmark dataset for interactive image segmentation methods and consists of 50 natural images with ground truths.

Evaluation metrics: To evaluate the segmentation performance, we adopt the commonly used region similarity in terms of intersection-over-union (IoU) between the predicted object mask **M** (*i.e.* $\mathbf{sm}^1$ in the proposed model) and the binary ground truth **S**,

$$IoU = \frac{|\mathbf{M} \cap \mathbf{S}|}{|\mathbf{M} \cup \mathbf{S}|}, \tag{7}$$

we compute the mean of the *IoU* across all images in the test set and thus also refer to this metric as *mIoU*.

Implementation details: We implement our model based on MATLAB R2014a platform with the Caffe [38] framework, and train our model in an end-to-end manner. We run our model in a PC with an i7-6700K CPU (16 GB memory) and a NVIDIA Titan X GPU (12 GB memory). The parameters of our backbone are initialized from the VGG-16 model [21]. For other convolutional layers, we initialize the weights by the "xavier" method [39]. In the training phase, we use the standard stochastic gradient descent (SGD) [40] method with batch size 8, momentum 0.9 and weight decay 0.0001. The learning rate is set to 1e−7 and decreased by 10% after 20,000 iterations. The model needs about 40,000 training iterations for convergence, which takes nearly 37.5 h. When testing, the proposed model takes 0.11 s to generate the segmentation map with a resolution of 288×288, which is then resized to the same size as the input image.

Table 1
Ablation study of the proposed model on OSIE-CFPS testing set.

Aspects	Methods	mIoU [%]
	Ours ($\mathbf{sm}^1$)	**78.5**
LGN-like module Variants	LGNL-w/o dilation	73.0
	LGNL-one convolution	76.7
	LGNL-one layer	76.1
	w/o ConvLSTM block	75.3
	w/o deep supervision	70.0
Architecture	$\mathbf{sm}^2$	78.2
	$\mathbf{sm}^3$	77.6
	$\mathbf{sm}^4$	74.7
	$\mathbf{sm}^5$	69.7

3.2. Ablation study

We now conduct a more detailed examination of our model on OSIE-CFPS dataset, and we change one component each time to assess individual contributions. All the experiments in this section are retrained with the same hyperparameters as described in Section 3.1.

LGN-like module: In our model, the function of LGN-like module is to simply simulate visual information processing in LGN. However, due to the complexity of the nervous system in LGN, we only use three dilated convolutional layers with different dilation rates to simply imitate different cells in LGN. In order to verify the effectiveness of simulation in LGN-like module, we directly remove the three dilated convolutional layers, and thus $\mathbf{f}^{\tau_i}$ is straightforwardly fed to the ConvLSTM block instead of $\mathbf{fd}^{\tau_i}$, named *LGNL-w/o dilation*. From Table 1, we find that the performance of *LGNL-w/o dilation* is 73.0%, which is 5.5% lower than the original model (*i.e. Ours*). When we add a convolutional layer with 32 filters with kernel size 3×3 to *LGNL-w/o dilation*, we get *LGNL-one convolution* which is 3.7% higher than *LGNL-w/o dilation* but still 1.8% lower than *Ours*. This demonstrates that the dilated convolutional layers in LGNL aggregate and integrate the features from VGG-16 backbone well, and the dilated convolutional layers with three different dilation rates can better simulate the information processing in different cells, *i.e.* processing information with different receptive fields reflects the multi-scale information fusion.

Besides, we replace the input of LGNL, which contains all the feature maps at five different resolutions, with only the feature

Table 2
Quantitative comparison of different methods on the OSIE-CFPS testing set and the GrabCut dataset. The red is the best, and the blue is the second best.

Aspects	Methods	mIoU [%] ↑	
		OSIE-CFPS dataset	GrabCut dataset
Clicking point based interactive segmentation	GraphCut [1]	50.1	65.9
	RandomWalk [2]	50.3	45.9
	GSC [3]	52.4	70.0
Fixation point based segmentation	GBOS [6]	41.8	64.9
	SOS [7]	42.2	64.9
	AVS [4]	47.8	58.8
	SegNet [25]	**59.4**	**72.7**
	Ours	**78.5**	**76.2**

map at the corresponding resolution from VGG_16 backbone, *e.g.* the input of *LGNLi* is $\mathbf{fm}_i$ only. We call it *LGNL-one layer* which reaches 76.1%, but 2.4% lower than *Ours*. This shows that the multi-layer features provide much richer contextual information than only one-layer features. To sum up, our LGN-like module can capture rich multi-layer and multi-scale contextual information for object segmentation.

ConvLSTM block: Compared with the classic convolutional layers, the ConvLSTM block recurrently processes the contextual features and iteratively optimizes the segmentation map. Hence, to study the contribution of the ConvLSTM block, we replace the ConvLSTM block with three convolutional layers, which have 32 filters with kernel size 3 × 3, named *w/o ConvLSTM block*. From Table 1, we observe a drop in mIoU of *w/o ConvLSTM block*, which demonstrates the effectiveness of the proposed ConvLSTM block.

Deep supervision: In the training phase, we adopt deeply supervised learning mechanism to provide pixel-level supervision of the intermediate results, *i.e.* $\mathbf{sm}^5$, $\mathbf{sm}^4$, $\mathbf{sm}^3$, and $\mathbf{sm}^2$, and the intermediate results are of high quality. To study the effect of deep supervision, we remove the $\mathcal{L}_5$, $\mathcal{L}_4$, $\mathcal{L}_3$ and $\mathcal{L}_2$ in Eq. (6), and the final loss function $\mathcal{L}$ is only $\mathcal{L}_1$. Then we retrain our model, and the variant *w/o deep supervision* achieves a mIoU of 70.0%. This means that the deep supervision plays an important role in our model with 8.5% improvement.

Segmentation refinement: We use the ConvLSTM block to refine our coarse segmentation map in a coarse-to-fine manner. Thus, we test 4 baselines: $\mathbf{sm}^5$, $\mathbf{sm}^4$, $\mathbf{sm}^3$, and $\mathbf{sm}^2$, and the corresponding mIoU values are 69.7%, 74.7%, 77.6% and 78.2%, respectively. We find that the segmentation maps are gradually optimized by ConvLSTM block, and the coarse-to-fine refinement is valid. Finally, the final segmentation maps, *i.e. Ours*, achieves a mIoU of 78.5%.

3.3. Performance comparison with the state-of-the-arts

We compare the proposed HVS based model with four state-of-the-art fixation point based segmentation methods including GBOS [6], SOS [7], AVS [4] and SegNet [25]. In addition, we also compare our model with three clicking point based interactive segmentation methods, which are GraphCut [1], RandomWalk [2] and GSC [3]. The segmentation results of all the other methods are obtained by running the publicly available codes provided by the authors.

For a fair comparison on OSIE-CFPS testing set, we make an appropriate transformation of the fixation point map in order to simulate the interaction of clicking point. Given a fixation point map, the fixation points are treated as the foreground/positive seeds. If an interactive segmentation method needs background/negative seeds, we randomly sample five background pixels in the binary ground truth as background/negative seeds. Then we morphologically dilate all points including fixation points and background points by two pixels to get the label map, which is used to segment objects in interactive segmentation methods.

We also compare our model with a semantic segmentation approach, *i.e.* SegNet [25]. To make it comparable, the connected component of a given semantic label that contains the fixation point is selected as the object and the remaining area is treated as background. As for AVS [4], we only take one fixation point once for segmentation, and then merge all the segmentation maps to obtain the final segmentation result.

Quantitative comparison: For a quantitative comparison, Table 2 shows that our model can achieve the better segmentation performance than either the clicking point based interactive segmentation methods or the fixation point based segmentation methods on OSIE-CFPS testing set. The mIoU values of other methods are mostly around 50.0% on OSIE-CFPS testing set, while our method reaches 78.5%, which is 26.1% better than the clicking point based interactive segmentation method GSC [3] and 19.1% better than the best fixation point based segmentation method [25]. The reason behind this is that our model, based on physiological structure, simulates the three processes of low-level, medium-level and high-level processing in the human visual system. And the experimental results demonstrate that the strategy of simulating HVS in our network is effective and superior for fixation point based segmentation.

Qualitative comparison: Fig. 4 illustrates some segmentation maps generated by our model as well as other seven state-of-the-art methods on OSIE-CFPS testing set (the 1–8 rows). It can be seen that our method is well applicable to various complex scenes. For images with multiple objects and complex background, our method can segment the entire objects with fine details. However, other fixation point based methods can only segment partial or incomplete objects, and clicking point based interactive segmentation methods can only segment the region where objects are located and cannot segment objects in detail.

3.4. Transferring to the clicking point based interactive segmentation

In particular, we also test our model on the GrabCut dataset to evaluate the reasonableness and validity of the transfer from the constrained fixation points to the clicking points. For each image in the GrabCut dataset, we randomly sample four points in the foreground as fixation points/positive seeds, and also randomly take four points in the background as negative seeds for the clicking point based interactive segmentation methods [1–3]. Notably, when our HVS based neural network performs segmentation with clicking points, the clicking points are treated as the fixation points. Similarly as the fixation points, the clicking points are also blurred by a Gaussian filter to produce a clicking density map. The clicking density map enables our neural network to understand which object the user needs to segment. In order to simulate various inputs from different users in the interactive segmentation, we conduct three sampling processes on each test image, and report the corresponding mean value in Table 2.

In Table 2, we can find that the performances of the fixation point based segmentation methods are generally better than those of the clicking point based interactive segmentation methods on

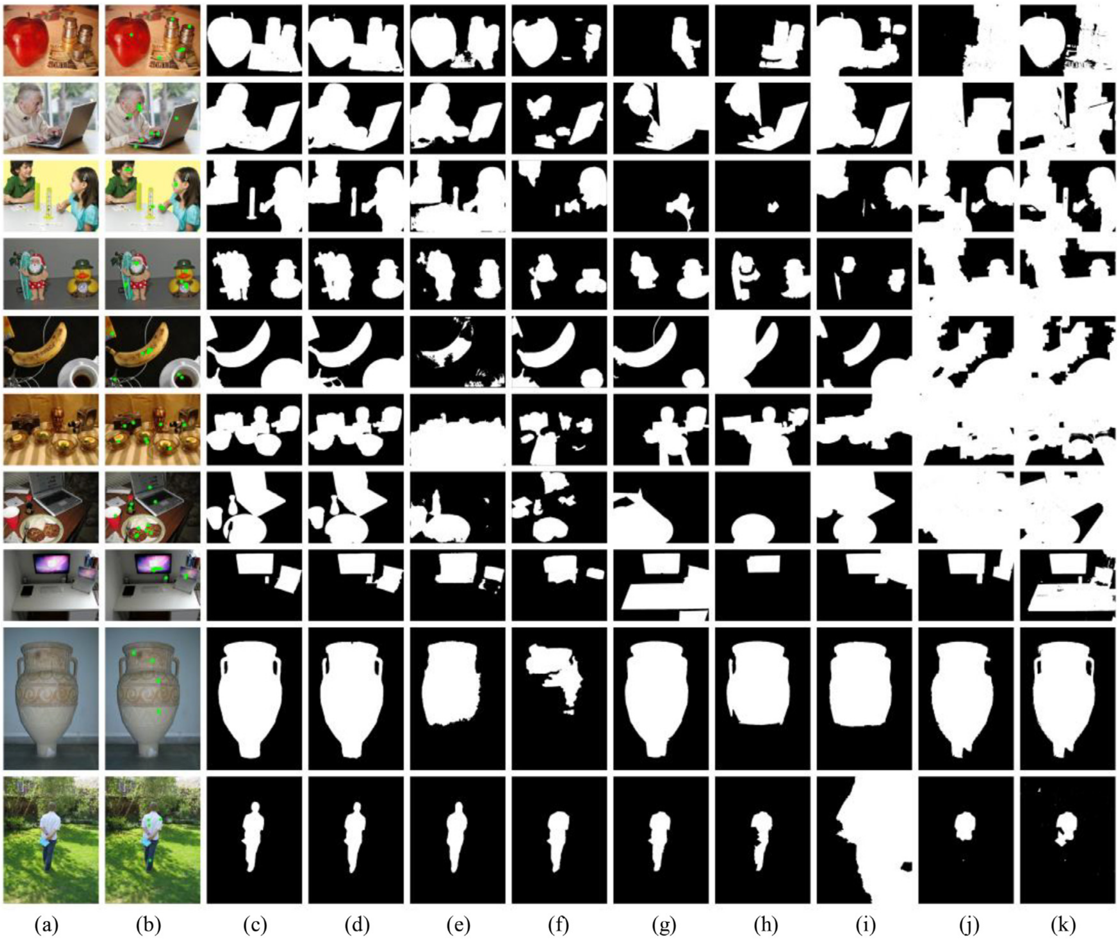


Fig. 4. Qualitative comparisons of our model and seven state-of-the-art methods on OSIE-CFPS testing set and GrabCut dataset. (a) Images; (b) Images with fixation/clicking points (*i.e.* green dots); (c) ground truths; segmentation results obtained using (d) Ours; (e) SegNet [25]; (f) AVS [4]; (g) SOS [7]; (h) GBOS [6]; (i) RandomWalk [2]; (j) GSC [3] and (k) GraphCut [1]. Please zoom-in for details, especially (b).

the GrabCut dataset. Concretely, we calculate the mean mIoU values of the fixation point based segmentation methods and the clicking point based interactive segmentation methods, which are 67.5% and 60.6%, respectively. This demonstrates that the transfer from the constrained fixation points to the clicking points is valid and successful. Notably, our method achieves the best performance 76.2% on the GrabCut dataset among all the eight methods, and further demonstrates that the transfer of our HVS based neural network from the constrained fixation points to the clicking points is more efficient. This reveals that the information processing structure of simulating HVS in our model is also valid and suitable.

Qualitative results over example images from the GrabCut dataset are depicted in the bottom two rows of Fig. 4. Concretely, our method can segment the objects completely with only four positive points (green dots in Fig. 4), which are regarded as fixation points, while the clicking point based interactive segmentation methods [1–3] cannot segment the objects well with four positive points and four negative points. For clarity, note that the negative points are not marked in Fig. 4.

Both qualitative and quantitative results on the GrabCut dataset demonstrate that the transfer from the constrained fixation points to the clicking points is successful, and our HVS based neural network works well for the clicking point based interactive segmentation. It benefits from not only the similarity between the constrained fixation points and the clicking points, but also the simulation of the visual information transmission and processing in HVS. Especially, we use three dilated convolutional layers with different dilation rates to imitate different cells in LGN, and thus the LGN-like module can capture rich multi-layer and multi-scale contextual information for object segmentation.

4. Conclusion

In this paper, we propose a HVS based model for transferring the constrained fixation point based segmentation to the clicking point based interactive segmentation. We first obtain the multiple-layer feature maps by feeding the RGB image to the VGG-16 backbone. Then we propose a LGN-like module to aggregate and fuse multiple-layer feature maps at different resolutions. The integrated contextual features and the fixation density map are fed into the proposed ConvLSTM blocks to segment the gazed objects in a coarse-to-fine manner. Experimental results confirm that the simulation of HVS in our model is superior and effective, and demonstrate that the transfer from the constrained fixation points to the clicking points is reasonable and valid.

Conflict of interest

None.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61771301 and 61801219.

References

[1] Y. Boykov, M. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2001, pp. 105–112.

[2] L. Grady, Random walks for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1768–1783.

[3] V. Gulshan, C. Rother, A. Criminisi, A. Blake, A. Zisserman, Geodesic star convexity for interactive image segmentation, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3129–3136.

[4] A.K. Mishra, Y. Aloimonos, C.L. Fah, Active segmentation with fixation, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2009, pp. 468–475.

[5] X. Tian, C. Jung, Point-cut: fixation point-based image segmentation using random walk model, in: Proceedings of the International Conference on Image Processing (ICIP), IEEE, 2015, pp. 2125–2129.

[6] R. Shi, N.K. Ngan, H. Li, Gaze-based object segmentation, IEEE Signal Process. Lett. 24 (10) (2017) 1493–1497.

[7] Y. Li, X. Hou, C. Koch, J.M. Rehg, A.L. Yuille, The secrets of salient object segmentation, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 280–287.

[8] H. Wang, H. Lv, Salient object detection with fixation priori, in: Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC), IEEE, 2016, pp. 285–289.

[9] S. Li, C. Zeng, S. Liu, Y. Fu, Merging fixation for saliency detection in a multilayer graph, Neurocomputing 230 (2017) 173–183.

[10] W. Wang, J. Shen, X. Dong, A. Borji, Salient object detection driven by fixation prediction, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2018, pp. 1711–1720.

[11] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, L. Van Gool, The interestingness of images, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2013, pp. 1633–1640.

[12] K. Rayner, Eye movements and attention in reading, scene perception, and visual search, Q. J. Exp. Psychol. 62 (8) (2009) 1457–1506.

[13] J.E. Hoffman, B. Subramaniam, The role of visual attention in saccadic eye movements, Atten. Percept. Psychophys. 57 (6) (1995) 787–795.

[14] Y. Xu, N. Li, J. Wu, J. Yu, S. Gao, Beyond universal saliency: personalized saliency prediction with multi-task CNN, in: Proceedings of the International Joint Conference on Artificial Intelligent (IJCAI), 2017, pp. 3887–3893.

[15] R. Quan, J. Han, D. Zhang, F. Nie, X. Qian, X. Li, Unsupervised salient object detection via inferring from imperfect saliency models, IEEE Trans. Multimed. 5 (20) (2018) 1101–1112.

[16] H. Tang, C. Chen, X. Pei, Saliency detection from one time sampling for eye fixation prediction, Multimed. Tools Appl. 77 (1) (2018) 165–184.

[17] Y. Fang, Z. Chen, W. Lin, C. Lin, Saliency detection in the compressed domain for adaptive image retargeting, IEEE Trans. Image Process. 9 (21) (2012) 3888–3901.

[18] Y. Fang, W. Lin, Z. Chen, C. Tsai, C. Lin, A video saliency detection model in compressed domain, IEEE Trans. Circuits Syst. Video Technol. 1 (24) (2014) 27–38.

[19] Y. Fang, C. Zhang, J. Li, J. Lei, M.P. Silva, P.L. Callet, Visual attention modeling for stereoscopic video: a benchmark and computational model, IEEE Trans. Image Process. 26 (10) (2017) 4684–4696.

[20] Z. Liu, W. Zou, O. Le Meur, Saliency tree: A novel saliency detection framework, IEEE Trans. Image Process. 23 (5) (2014) 1937–1952.

[21] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations (ICLR), 2015, pp. 1–14.

[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778.

[23] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the Neural Information Processing Systems (NIPS), 2012, pp. 1097–1105.

[24] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 3431–3440.

[25] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.

[26] S. Caelles, K.K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, L. van Gool, One-shot video object segmentation, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 5320–5329.

[27] D. Fan, W. Wang, M. Cheng, J. Shen, Shifting more attention to video salient object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019.

[28] D. Zhang, J. Han, Y. Zhang, D. Xu, Synthesizing supervision for learning deep saliency network without human annotation, IEEE Trans. Pattern Anal. Mach. Intell. (2019) 1, doi:10.1109/TPAMI.2019.2900649.

[29] L. Theis, I. Korshunova, A. Tejani, F. Huszár, Faster Gaze Prediction with Dense Networks and Fisher Pruning, arXiv 1801.05787, 2018.

[30] A.H. Bell, L. Pessoa, R.B.H. Tootell, L.G. Ungerleider, Visual perception of objects, Fundam. Neurosci. (2012) 947–968.

[31] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2015, pp. 1395–1403.

[32] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: Proceedings of the International Conference on Learning Representations (ICLR), 2016.

[33] X. Shi, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, W.C. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, in: Proceedings of the Neural Information Processing Systems (NIPS), 2015, pp. 802–810.

[34] H. Song, W. Wang, S. Zhao, J. Shen, L. Lam, Pyramid dilated deeper ConvLSTM for video salient object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2018, pp. 744–760.

[35] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[36] J. Xu, M. Jiang, S. Wang, M.S. Kankanhalli, Q. Zhao, Predicting human gaze beyond pixels, J. Vis. 14 (1) (2018) 1–20.

[37] C. Rother, V. Kolmogorov, A. Blake, “GrabCut”: interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (3) (2004) 309–314.

[38] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceeding of the International Conference on Multimedia, ACM, 2014, pp. 675–678.

[39] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), 2010, pp. 249–256.

[40] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of the International Conference on Computational Statistics (COMPSTAT), 2010, pp. 177–186.

Gongyang Li received the B.E. degree from Shanghai Normal University, Shanghai, China, in 2016. He is currently pursuing the Ph.D. degree at the School of Communication and Information Engineering, Shanghai University, Shanghai, China. His research interests include image/video object segmentation and saliency detection.

Zhi Liu received the B.E. and M.E. degrees from Tianjin University, Tianjin, China, and the Ph.D. degree from Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai, China, in 1999, 2002, and 2005, respectively. He is currently a Professor with the School of Communication and Information Engineering, Shanghai University, Shanghai, China. From Aug. 2012 to Aug. 2014, he was a Visiting Researcher with the SIROCCO Team, IRISA/INRIA-Rennes, France, with the support by EU FP7 Marie Curie Actions. He has published more than 170 refereed technical papers in international journals and conferences. His research interests include image/video processing, machine learning, computer vision and multimedia communication. He was a TPC member/session chair in ICIP 2017, PCM 2016, VCIP 2016, ICME 2014, WIAMIS 2013, *etc.* He co-organized special sessions on visual attention, saliency models, and applications at WIAMIS 2013 and ICME 2014. He is an area editor of *Signal Processing: Image Communication* and served as a guest editor for the special issue on *Recent Advances in Saliency Models, Applications and Evaluations* in *Signal Processing: Image Communication*. He is a senior member of IEEE.

Ran Shi received his B.S. degree in Electronic Science and Technology from Changshu Institute of Technology and M.S. degree in Signal and Information Processing from Shanghai University in 2009 and 2012. He joined The Chinese University of Hong Kong (CUHK) as a Research Assistant in 2012, and obtained his Ph.D. in Electronic Engineering (CUHK) in 2017. Currently, he is an assistant professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include object segmentation, visual quality evaluation, interactive segmentation and salient object detection.

Weijie Wei received the B.E. degree from Shanghai University, Shanghai, China, in 2018. He is currently pursuing the M.E. degree at the School of Communication and Information Engineering, Shanghai University, Shanghai, China. His research interests include deep learning and saliency prediction.