



DeepVS: A Deep Learning Based Video Saliency Prediction Approach

Lai Jiang¹, Mai Xu², Tie Liu, Minglang Qiao, and Zulin Wang

Beihang University, Beijing, China

{jianglai.china,maixu,liutie,minglangqiao,wzulin}@buaa.edu.cn

Abstract. In this paper, we propose a novel deep learning based video saliency prediction method, named DeepVS. Specifically, we establish a large-scale eye-tracking database of videos (LEDOV), which includes 32 subjects' fixations on 538 videos. We find from LEDOV that human attention is more likely to be attracted by objects, particularly the moving objects or the moving parts of objects. Hence, an object-to-motion convolutional neural network (OM-CNN) is developed to predict the intra-frame saliency for DeepVS, which is composed of the objectness and motion subnets. In OM-CNN, cross-net mask and hierarchical feature normalization are proposed to combine the spatial features of the objectness subnet and the temporal features of the motion subnet. We further find from our database that there exists a temporal correlation of human attention with a smooth saliency transition across video frames. We thus propose saliency-structured convolutional long short-term memory (SS-ConvLSTM) network, using the extracted features from OM-CNN as the input. Consequently, the inter-frame saliency maps of a video can be generated, which consider both structured output with center-bias and cross-frame transitions of human attention maps. Finally, the experimental results show that DeepVS advances the state-of-the-art in video saliency prediction.

Keywords: Saliency prediction · Convolutional LSTM
Eye-tracking database

1 Introduction

The foveation mechanism in the human visual system (HVS) indicates that only a small fovea region captures most visual attention at high resolution, while other peripheral regions receive little attention at low resolution. To predict human attention, saliency prediction has been widely studied in recent years, with multiple applications [5, 21, 22, 38] in object recognition, object segmentation, action recognition, image caption, and image/video compression, among others. In this

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01264-9_37) contains supplementary material, which is available to authorized users.

paper, we focus on predicting video saliency at the pixel level, which models attention on each video frame.

The traditional video saliency prediction methods mainly focus on the feature integration theory [16, 19, 20, 26], in which some spatial and temporal features were developed for video saliency prediction. Differing from the integration theory, the deep learning (DL) based methods [13, 18, 28, 29, 32] have been recently proposed to learn human attention in an end-to-end manner, significantly improving the accuracy of image saliency prediction. However, only a few works have managed to apply DL in video saliency prediction [1, 2, 23, 27]. Specifically, Cagdas *et al.* [1] applied a two-stream CNN structure taking both RGB frames and motion maps as the inputs for video saliency prediction. Bazzani *et al.* [2] leveraged a deep convolutional 3D (C3D) network to learn the representations of human attention on 16 consecutive frames, and then a long short-term memory (LSTM) network connected to a mixture density network was learned to generate saliency maps in a Gaussian mixture distribution.

For training the DL networks, we establish a large-scale eye-tracking database of videos (LEDOV) that contains the free-view fixation data of 32 subjects viewing 538 diverse-content videos. We validate that 32 subjects are enough through consistency analysis among subjects, when establishing our LEDOV database. The previous databases [24, 33] do not investigate the sufficient number of subjects in the eye-tracking experiments. For example, although Hollywood [24] contains 1857 videos, it only has 19 subjects and does not show whether the subjects are sufficient. More importantly, Hollywood focuses on task-driven attention, rather than free-view saliency prediction.

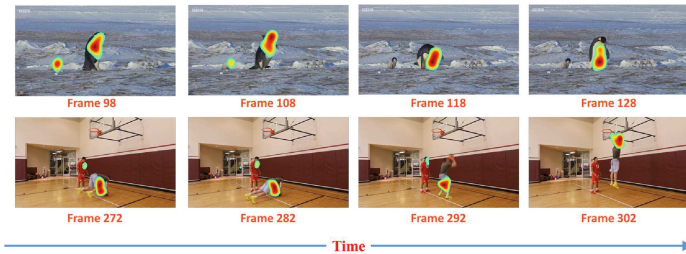


Fig. 1. Attention heat maps of some frames selected from two videos. The heat maps show that: (1) the regions with object can draw a majority of human attention, (2) the moving objects or the moving parts of objects attract more human attention, and (3) a dynamic pixel-wise transition of human attention occurs across video frames.

In this paper, we propose a new DL based video saliency prediction (DeepVS) method. We find from Fig. 1 that people tend to be attracted by the moving objects or the moving parts of objects, and this finding is also verified in the analysis of our LEDOV database. However, all above DL based methods do not explore the motion of objects in predicting video saliency. In DeepVS, a

novel object-to-motion convolutional neural network (OM-CNN) is constructed to learn the features of object motion, in which the cross-net mask and hierarchical feature normalization (FN) are proposed to combine the subnets of objectness and motion. As such, the moving objects at different scales can be located as salient regions.

Both Fig. 1 and the analysis of our database show that the saliency maps are smoothly transitioned across video frames. Accordingly, a saliency-structured convolutional long short-term memory (SS-ConvLSTM) network is developed to predict the pixel-wise transition of video saliency across frames, with the output features of OM-CNN as the input. The traditional LSTM networks for video saliency prediction [2, 23] assume that human attention follows the Gaussian mixture distribution, since these LSTM networks cannot generate structured output. In contrast, our SS-ConvLSTM network is capable of retaining spatial information of attention distribution with structured output through the convolutional connections. Furthermore, since the center-bias (CB) exists in the saliency maps as shown in Fig. 1, a CB dropout is proposed in the SS-ConvLSTM network. As such, the structured output of saliency considers the CB prior. Consequently, the dense saliency prediction of each video frame can be obtained in DeepVS in an end-to-end manner. The experimental results show that our DeepVS method advances the state-of-the-art of video saliency prediction in our database and other 2 eye-tracking databases. Both the DeepVS code and the LEDOV database are available online.

2 Related Work

Feature Integration Methods. Most early saliency prediction methods [16, 20, 26, 34] relied on the feature integration theory, which is composed of two main steps: feature extraction and feature fusion. In the image saliency prediction task, many effective spatial features were extracted to predict human attention with either a top-down [17] or bottom-up [4] strategy. Compared to image, video saliency prediction is more challenging because temporal features also play an important role in drawing human attention. To achieve this, a countable amount of motion-based features [11, 42] were designed as additional temporal information for video saliency prediction. Besides, some methods [16, 40] focused on calculating a variety of temporal differences across video frames, which are effective in video saliency prediction. Taking advantage of sophisticated video coding standards, the methods of [7, 37] explored the spatio-temporal features in compressed domain for predicting video saliency. In addition to feature extraction, many works have focused on the fusion strategy to generate video saliency maps. Specifically, a set of probability models [15, 31, 40] were constructed to integrate different kinds of features in predicting video saliency. Moreover, other machine learning algorithms, such as support vector machine and neural network, were also applied to linearly [26] or non-linearly [20] combine the saliency-related features. Other advanced methods [9, 19, 41] applied phase spectrum analysis in the fusion model to bridge the gap between features and video saliency. For instance,

Guo *et al.* [9] exploited phase spectrum of quaternion Fourier transform (PQFT) on four feature channels to predict video saliency.

DL Based Methods. Most recently, DL has been successfully incorporated to automatically learn spatial features for predicting the saliency of images [13, 18, 28, 29, 32]. However, only a few works have managed to apply DL in video saliency prediction [1–3, 23, 27, 33, 35]. In these works, the dynamic characteristics were explored in two ways: adding temporal information to CNN structures [1, 3, 27, 35] or developing a dynamic structure with LSTM [2, 23]. For adding temporal information, a four-layer CNN in [3] and a two-stream CNN in [1] were trained with both RGB frames and motion maps as the inputs. Similarly, in [35], the pair of consecutive frames concatenated with a static saliency map (generated by the static CNN) are fed into the dynamic CNN for video saliency prediction, allowing the CNN to generalize more temporal features. In our work, the OM-CNN structure of DeepVS includes the subnets of objectness and motion, since human attention is more likely to be attracted by the moving objects or the moving parts of objects. For developing the dynamic structure, Bazzani *et al.* [2] and Liu *et al.* [23] applied LSTM networks to predict video saliency maps, relying on both short- and long-term memory of attention distribution. However, the fully connected layers in LSTM limit the dimensions of both the input and output; thus, it is unable to obtain the end-to-end saliency map and the strong prior knowledge needs to be assumed for the distribution of saliency in [2, 23]. In our work, DeepVS explores SS-ConvLSTM to directly predict saliency maps in an end-to-end manner. This allows learning the more complex distribution of human attention, rather than a pre-assumed distribution of saliency.

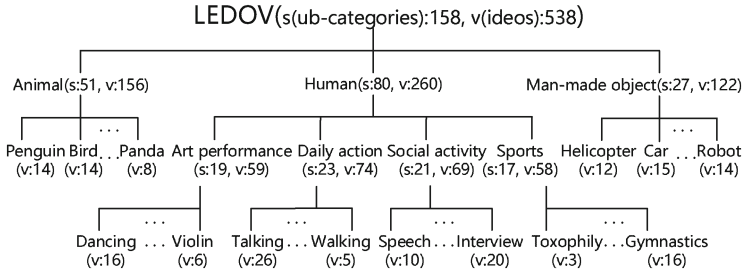


Fig. 2. Category tree of videos in LEDOV according to the content. The numbers of categories/sub-categories are shown in the brackets. Besides, the number of videos for each category/sub-category is also shown in the brackets.

3 LEDOV Database

For training the DNN models of DeepVS, we establish the LEDOV database. Some details of establishing LEDOV database are as follows.

Stimuli. In order to make the content of LEDOV diverse, we constructed a hierarchical tree of key words for video categories as shown in Fig. 2. There were three main categories, i.e., animal, human and man-made object. Note that the natural scene videos were not included, as they are scarce in comparison with other categories. The category of animal had 51 sub-categories. Similarly, the category of man-made objects was composed of 27 sub-categories. The category of human had the sub-categories of daily action, sports, social activity and art performance. These sub-categories of human were further classified as can be seen in Fig. 2. Consequently, we obtained 158 sub-categories in total, and then collected 538 videos belonging to these 158 sub-categories from YouTube. The number of videos for each category/sub-category can be found in Fig. 2. Some examples of the collected videos are provided in the supplementary material. It is worth mentioning that LEDOV contains the videos with a total of 179,336 frames and 6,431 seconds, and that all videos are at least 720p resolution and 24 Hz frame rate.

Procedure. For monitoring the binocular eye movements, a Tobii TX300 eye tracker [14] was used in our experiment. During the experiment, the distance between subjects and the monitor was fixed at 65 cm. Before viewing videos, each subject was required to perform a 9-point calibration for the eye tracker. Afterwards, the subjects were asked to free-view videos displayed at a random order. Meanwhile, the fixations of the subjects were recorded by the eye tracker.

Subjects. A new scheme was introduced for determining the sufficient number of participants. We stopped recruiting subjects for eye-tracking experiments once recorded fixations converged. Specifically, the subjects (with even numbers), who finished the eye-tracking experiment, were randomly divided into 2 equal groups by 5 times. Then, we measured the linear correlation coefficient (CC) of the fixation maps from two groups, and the CC values are averaged over the 5-time division. Figure 3 shows the averaged CC values of two groups, when the number of subjects increases. As seen in this figure, the CC value converges when the subject number reaches 32. Thus, we stopped recruiting subjects, when we collected the fixations of 32 subjects. Finally, 5,058,178 fixations of all 32 subjects on 538 videos were collected for our eye-tracking database.

Findings. We mine our database to analyze human attention on videos. Specifically, we have the following 3 findings, the analysis of which is presented in the supplemental material. *Finding 1:* High correlation exists between objectness and human attention. *Finding 2:* Human attention is more likely to be attracted by the moving objects or the moving parts of objects. *Finding 3:* There exists a temporal correlation of human attention with a smooth saliency transition across video frames.

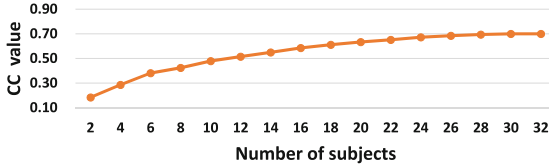


Fig. 3. The consistency (CC value) for different numbers of subjects over all videos in LEDOV.

4 Proposed Method

4.1 Framework

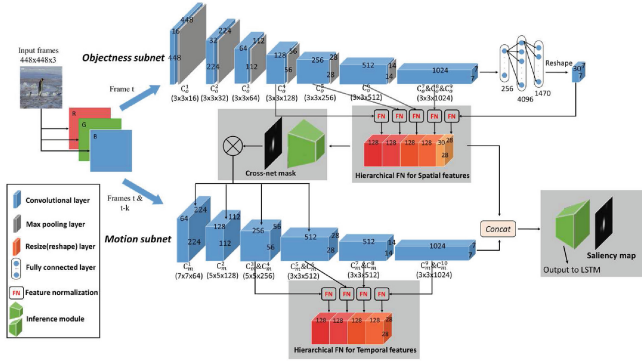
For video saliency prediction, we develop a new DNN architecture that combines OM-CNN and SS-ConvLSTM. According to *Findings 1 and 2*, human attention is highly correlated to objectness and object motion. As such, OM-CNN integrates both regions and motion of objects to predict video saliency through two subnets, i.e., the subnets of objectness and motion. In OM-CNN, the objectness subnet yields a cross-net mask on the features of the *convolutional layers* in the motion subnet. Then, the spatial features from the objectness subnet and the temporal features from the motion subnet are concatenated by the proposed hierarchical feature normalization to generate the spatio-temporal features of OM-CNN. The architecture of OM-CNN is shown in Fig. 4. Besides, SS-ConvLSTM with the CB dropout is developed to learn the dynamic saliency of video clips, in which the spatio-temporal features of OM-CNN serve as the input. Finally, the saliency map of each frame is generated from 2 *deconvolutional layers* of SS-ConvLSTM. The architecture of SS-ConvLSTM is shown in Fig. 5.

4.2 Objectness and Motion Subnets in OM-CNN

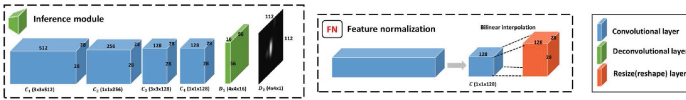
In OM-CNN, an objectness subnet is designed for extracting multi-scale spatial features related to objectness information, which is based on a pre-trained YOLO [30]. To avoid over-fitting, a pruned structure of YOLO is applied as the objectness subnet, including 9 *convolutional layers*, 5 *pooling layers* and 2 *fully connected layers (FC)*. To further avoid over-fitting, an additional *batch-normalization layer* is added to each *convolutional layer*. Assuming that $BN(\cdot)$, $P(\cdot)$ and $*$ are the batch-normalization, max pooling and convolution operations, the output of the k -th *convolutional layer* \mathbf{C}_o^k in the objectness subnet can be computed as

$$\mathbf{C}_o^k = L_{0.1}(BN(P(\mathbf{C}_o^{k-1}) * \mathbf{W}_o^{k-1} + \mathbf{B}_o^{k-1})), \quad (1)$$

where \mathbf{W}_o^{k-1} and \mathbf{B}_o^{k-1} indicate the kernel parameters of weight and bias at the $(k-1)$ -th *convolutional layer*, respectively. Additionally, $L_{0.1}(\cdot)$ is a leaky ReLU activation with leakage coefficient of 0.1. In addition to the objectness subnet, a motion subnet is also incorporated in OM-CNN to extract multi-scale temporal



(a) The overall architecture of OM-CNN



(b) The details for sub-modules of inference module and feature normalization

Fig. 4. Overall architecture of our OM-CNN for predicting video saliency of intra-frame. The sizes of convolutional kernels are shown in the figure. For instance, $3 \times 3 \times 16$ means 16 convolutional kernels with size of 3×3 . Note that the 7 – 9th *convolutional layers* (C_7^9 , C_8^9 & C_9^9) in the objectness subnet have the same size of convolutional kernels, thus sharing the same cube in (a) but not sharing the parameters. Similarly, each of the last four cubes in the motion subnet represents 2 *convolutional layers* with same kernel size. The details of the inference and feature normalization modules are shown in (b). Note that the proposed cross-net mask, hierarchical feature normalization and saliency inference module are highlighted with gray background.

features from the pair of neighboring frames. Similar to the objectness subnet, a pruned structure of FlowNet [6] with 10 *convolutional layers* is applied as the motion subnet. For details about objectness and motion subnets, please refer to Fig. 4(a). In the following, we propose combining the subnets of objectness and motion.

4.3 Combination of Objectness and Motion Subnets

In OM-CNN, we propose the hierarchical FN and cross-net mask to combine the multi-scale features of both objectness and motion subnets for predicting saliency. In particular, the cross-net mask can be used to encode objectness information when generating temporal features. Moreover, the inference module is developed to generate the cross-net mask or saliency map, based on the learned features.

Hierarchical FN. For leveraging the multi-scale information with various receptive fields, the output features are extracted from different *convolutional*

layers of the objectness and motion subnets. Here, a hierarchical FN is introduced to concatenate the multi-scale features, which have different resolutions and channel numbers. Specifically, we take hierarchical FN for spatial features as an example. First, the features of the 4-th, 5-th, 6-th and last *convolutional layer* in the objectness subnet are normalized through the FN module to obtain 4 sets of spatial features $\{\mathbf{FS}_i\}_{i=1}^4$. As shown in Fig. 4(b), each FN module is composed of a 1×1 *convolutional layer* and a *bilinear layer* to normalize the input features into 128 channels at a resolution of 28×28 . All spatial features¹ $\{\mathbf{FS}_i\}_{i=1}^5$ are concatenated in a hierarchy to obtain a total size of $28 \times 28 \times 542$, as the output of hierarchical FN. Similarly, the features of the 4-th, 6-th, 8-th and 10-th *convolutional layers* of the motion subnet are concatenated by hierarchical FN, such that the temporal features $\{\mathbf{FT}_i\}_{i=1}^4$ with a total size of $28 \times 28 \times 512$ are obtained.

Inference Module. Then, given the extracted spatial features $\{\mathbf{FS}_i\}_{i=1}^5$ and temporal features $\{\mathbf{FT}_i\}_{i=1}^4$ from the two subnets of OM-CNN, an inference module I_f is constructed to generate the saliency map \mathbf{S}_f , which models the intra-frame saliency of a video frame. Mathematically, \mathbf{S}_f can be computed as

$$\mathbf{S}_f = I_f(\{\mathbf{FS}_i\}_{i=1}^5, \{\mathbf{FT}_i\}_{i=1}^4). \quad (2)$$

The inference module I_f is a CNN structure that consists of 4 *convolutional layers* and 2 *deconvolutional layers* with a stride of 2. The detailed architecture of I_f is shown in Fig. 4(b). Consequently, \mathbf{S}_f is used to train the OM-CNN model, as discussed in Sect. 4.5. Additionally, the output of *convolutional layer* C_4 with a size of $28 \times 28 \times 128$ is viewed as the final spatio-temporal features, denoted as \mathbf{FO} . Afterwards, \mathbf{FO} is fed into SS-ConvLSTM for predicting intra-frame saliency.

Cross-Net Mask. *Finding 2* shows that attention is more likely to be attracted by the moving objects or the moving parts of objects. However, the motion subnet can only locate the moving parts of a whole video frame without any object information. Therefore, the cross-net mask is proposed to impose a mask on the *convolutional layers* of the motion subnet, for locating the moving objects and the moving parts of objects. The cross-net mask \mathbf{S}_c can be obtained upon the multi-scale features of the objectness subnet. Specifically, given spatial features $\{\mathbf{FS}_i\}_{i=1}^5$ of the objectness subnet, \mathbf{S}_c can be generated by another inference module I_c as follows,

$$\mathbf{S}_c = I_c(\{\mathbf{FS}_i\}_{i=1}^5). \quad (3)$$

Note that the architecture of I_c is same as that of I_f as shown in Fig. 4(b), but not sharing the parameters. Consequently, the cross-net mask \mathbf{S}_c can be obtained to encode the objectness information, roughly related to salient regions. Then, the cross-net mask \mathbf{S}_c is used to mask the outputs of the first 6 *convolutional*

¹ \mathbf{FS}_5 is generated by the output of the last *FC layer* in the objectness subnet, encoding the high level information of the sizes, class and confidence probabilities of candidate objects in each grid.

layers of the motion subnet. Accordingly, the output of the k -th convolutional layer \mathbf{C}_m^k in the motion subnet can be computed as

$$\mathbf{C}_m^k = L_{0.1}(M(\mathbf{C}_m^{k-1}, \mathbf{S}_c) * \mathbf{W}_m^{k-1} + \mathbf{B}_m^{k-1}),$$

where $M(\mathbf{C}_m^{k-1}, \mathbf{S}_c) = \mathbf{C}_m^{k-1} \cdot (\mathbf{S}_c \cdot (1 - \gamma) + \mathbf{1} \cdot \gamma).$ (4)

In (4), \mathbf{W}_m^{k-1} and \mathbf{B}_m^{k-1} indicate the kernel parameters of weight and bias at the $(k - 1)$ -th convolutional layer in the motion subnet, respectively; γ ($0 \leq \gamma \leq 1$) is an adjustable hyper-parameter for controlling the mask degree, mapping the range of \mathbf{S}_c from $[0, 1]$ to $[\gamma, 1]$. Note that the last 4 convolutional layers are not masked with the cross-net mask for considering the motion of the non-object region in saliency prediction.

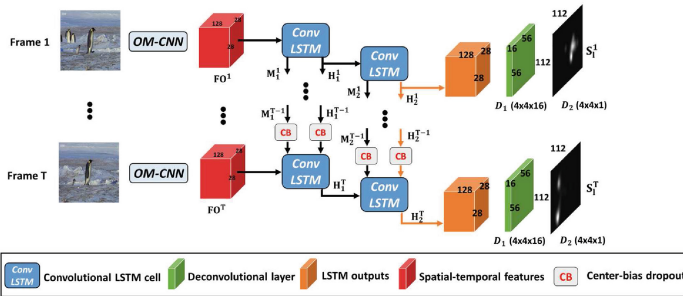


Fig. 5. Architecture of our SS-ConvLSTM for predicting saliency transition across inter-frame, following the OM-CNN. Note that the training process is not annotated in the figure.

4.4 SS-ConvLSTM

According to *Finding 3*, we develop the SS-ConvLSTM network for learning to predict the dynamic saliency of a video clip. At frame t , taking the OM-CNN features \mathbf{FO} as the input (denoted as \mathbf{FO}^t), SS-ConvLSTM leverages both long- and short-term correlations of the input features through the memory cells ($\mathbf{M}_1^{t-1}, \mathbf{M}_2^{t-1}$) and hidden states ($\mathbf{H}_1^{t-1}, \mathbf{H}_2^{t-1}$) of the 1-st and 2-nd LSTM layers at last frame. Then, the hidden states of the 2-nd LSTM layer \mathbf{H}_2^t are fed into 2 deconvolutional layers to generate final saliency map \mathbf{S}_t^t at frame t . The architecture of SS-ConvLSTM is shown in Fig. 5.

We propose a CB dropout for SS-ConvLSTM, which improves the generalization capability of saliency prediction via incorporating the prior of CB. It is because the effectiveness of the CB prior in saliency prediction has been verified [37]. Specifically, the CB dropout is inspired by the Bayesian dropout [8]. Given an input dropout rate p_b , the CB dropout operator $\mathbf{Z}(p_b)$ is defined based on an

L -time Monte Carlo integration:

$$\begin{aligned} \mathbf{Z}(p_b) &= \text{Bino}(L, p_b \cdot \mathbf{S}_{\text{CB}}) / (L \cdot \text{Mean}(\mathbf{S}_{\text{CB}})), \\ \text{where } \mathbf{S}_{\text{CB}}(i, j) &= 1 - \frac{\sqrt{(i - W/2)^2 + (j - H/2)^2}}{\sqrt{(W/2)^2 + (H/2)^2}}. \end{aligned} \quad (5)$$

$\text{Bino}(L, \mathbf{P})$ is a randomly generated mask, in which each pixel (i, j) is subject to a L -trial Binomial distribution according to probability $\mathbf{P}(i, j)$. Here, the probability matrix \mathbf{P} is modeled by CB map \mathbf{S}_{CB} , which is obtained upon the distance from pixel (i, j) to the center $(W/2, H/2)$. Consequently, the dropout operator takes the CB prior into account, the dropout rate of which is based on p_b .

Next, similar to [36], we extend the traditional LSTM by replacing the Hadamard product (denoted as \circ) by the convolutional operator (denoted as $*$), to consider the spatial correlation of input OM-CNN features in the dynamic model. Taking the first layer of SS-ConvLSTM as an example, a single LSTM cell at frame t can be written as

$$\begin{aligned} \mathbf{I}_1^t &= \sigma((\mathbf{H}_1^{t-1} \circ \mathbf{Z}_i^h) * \mathbf{W}_i^h + (\mathbf{F}^t \circ \mathbf{Z}_i^f) * \mathbf{W}_i^f + \mathbf{B}_i), \\ \mathbf{A}_1^t &= \sigma((\mathbf{H}_1^{t-1} \circ \mathbf{Z}_a^h) * \mathbf{W}_a^h + (\mathbf{F}^t \circ \mathbf{Z}_a^f) * \mathbf{W}_a^f + \mathbf{B}_a), \\ \mathbf{O}_1^t &= \sigma((\mathbf{H}_1^{t-1} \circ \mathbf{Z}_o^h) * \mathbf{W}_o^h + (\mathbf{F}^t \circ \mathbf{Z}_o^f) * \mathbf{W}_o^f + \mathbf{B}_o), \\ \mathbf{G}_1^t &= \tanh((\mathbf{H}_1^{t-1} \circ \mathbf{Z}_g^h) * \mathbf{W}_g^h + (\mathbf{F}^t \circ \mathbf{Z}_g^f) * \mathbf{W}_g^f + \mathbf{B}_g), \\ \mathbf{M}_1^t &= \mathbf{A}_1^t \circ \mathbf{M}_1^{t-1} + \mathbf{I}_1^t \circ \mathbf{G}_1^t, \quad \mathbf{H}_1^t = \mathbf{O}_1^t \circ \tanh(\mathbf{M}_1^t), \end{aligned} \quad (6)$$

where σ and \tanh are the activation functions of sigmoid and hyperbolic tangent, respectively. In (6), $\{\mathbf{W}_i^h, \mathbf{W}_a^h, \mathbf{W}_o^h, \mathbf{W}_g^h, \mathbf{W}_i^f, \mathbf{W}_a^f, \mathbf{W}_o^f, \mathbf{W}_g^f\}$ and $\{\mathbf{B}_i, \mathbf{B}_a, \mathbf{B}_o, \mathbf{B}_g\}$ denote the kernel parameters of weight and bias at each *convolutional layer*; $\mathbf{I}_1^t, \mathbf{A}_1^t$ and \mathbf{O}_1^t are the gates of input (i), forget (a) and output (o) for frame t ; $\mathbf{G}_1^t, \mathbf{M}_1^t$ and \mathbf{H}_1^t are the input modulation (g), memory cells and hidden states (h). They are all represented by 3-D tensors with a size of $28 \times 28 \times 128$. Besides, $\{\mathbf{Z}_i^h, \mathbf{Z}_a^h, \mathbf{Z}_o^h, \mathbf{Z}_g^h\}$ are four sets of randomly generated CB dropout masks ($28 \times 28 \times 128$) through $\mathbf{Z}(p_h)$ in (5) with a hidden dropout rate of p_h . They are used to mask on the hidden states \mathbf{H}_1^t , when computing different gates or modulation $\{\mathbf{I}_1^t, \mathbf{A}_1^t, \mathbf{O}_1^t, \mathbf{G}_1^t\}$. Similarly, given feature dropout rate p_f , $\{\mathbf{Z}_i^f, \mathbf{Z}_a^f, \mathbf{Z}_o^f, \mathbf{Z}_g^f\}$ are four randomly generated CB dropout masks from $\mathbf{Z}(p_f)$ for the input features \mathbf{F}^t . Finally, saliency map \mathbf{S}_1^t is obtained upon the hidden states of the 2-nd LSTM layer \mathbf{H}_2^t for each frame t .

4.5 Training Process

For training OM-CNN, we utilize the Kullback-Leibler (KL) divergence-based loss function to update the parameters. This function is chosen because [13] has proven that the KL divergence is more effective than other metrics in training DNNs to predict saliency. Regarding the saliency map as a probability distribution of attention, we can measure the KL divergence D_{KL} between the saliency

map \mathbf{S}_f of OM-CNN and the ground-truth distribution \mathbf{G} of human fixations as follows:

$$D_{\text{KL}}(\mathbf{G}, \mathbf{S}_f) = (1/WH) \sum_{i=1}^W \sum_{j=1}^H G_{ij} \log(G_{ij}/S_f^{ij}), \quad (7)$$

where G_{ij} and S_f^{ij} refer to the values of location (i, j) in \mathbf{G} and \mathbf{S}_f (resolution: $W \times H$). In (7), a smaller KL divergence indicates higher accuracy in saliency prediction. Furthermore, the KL divergence between the cross-net mask \mathbf{S}_c of OM-CNN and the ground-truth \mathbf{G} is also used as an auxiliary function to train OM-CNN. This is based on the assumption that the object regions are also correlated with salient regions. Then, the OM-CNN model is trained by minimizing the following loss function:

$$L_{\text{OM-CNN}} = \frac{1}{1+\lambda} D_{\text{KL}}(\mathbf{G}, \mathbf{S}_f) + \frac{\lambda}{1+\lambda} D_{\text{KL}}(\mathbf{G}, \mathbf{S}_c). \quad (8)$$

In (8), λ is a hyper-parameter for controlling the weights of two KL divergences. Note that OM-CNN is pre-trained on YOLO and FlowNet, and the remaining parameters of OM-CNN are initialized by the Xavier initializer. We found from our experimental results that the auxiliary function can decrease KL divergence by 0.24.

To train SS-ConvLSTM, the training videos are cut into clips with the same length T . In addition, when training SS-ConvLSTM, the parameters of OM-CNN are fixed to extract the spatio-temporal features of each T -frame video clip. Then, the loss function of SS-ConvLSTM is defined as the average KL divergence over T frames:

$$L_{\text{SS-ConvLSTM}} = \frac{1}{T} \sum_{i=1}^T D_{\text{KL}}(\mathbf{S}_i^i, \mathbf{G}_i). \quad (9)$$

In (9), $\{\mathbf{S}_i^i\}_{i=1}^T$ are the final saliency maps of T frames generated by SS-ConvLSTM, and $\{\mathbf{G}_i\}_{i=1}^T$ are their ground-truth attention maps. For each LSTM cell, the kernel parameters are initialized by the Xavier initializer, while the memory cells and hidden states are initialized by zeros.

5 Experimental Results

5.1 Settings

In our experiment, the 538 videos in our eye-tracking database are randomly divided into training (456 videos), validation (41 videos) and test (41 videos) sets. Specifically, to learn SS-ConvLSTM of DeepVS, we temporally segment 456 training videos into 24,685 clips, all of which contain T ($=16$) frames. An overlap of 10 frames is allowed in cutting the video clips, for the purpose of data augmentation. Before inputting to OM-CNN of DeepVS, the RGB channels of each frame are resized to 448×448 , with their mean values being removed. In training

OM-CNN and SS-ConvLSTM, we learn the parameters using the stochastic gradient descent algorithm with the Adam optimizer. Here, the hyper-parameters of OM-CNN and SS-ConvLSTM are tuned to minimize the KL divergence of saliency prediction over the validation set. The tuned values of some key hyper-parameters are listed in Table 1. Given the trained models of OM-CNN and SS-ConvLSTM, all 41 test videos in our eye-tracking database are used to evaluate the performance of our method, in comparison with 8 other state-of-the-art methods. All experiments are conducted on a single Nvidia GTX 1080 GPU. Benefiting from that, our method is able to make real-time prediction for video saliency at a speed of 30 Hz.

Table 1. The values of hyper-parameters in OM-CNN and SS-ConvLSTM.

OM-CNN	Objectness mask parameter γ in (4)	0.5
	KL divergences weight λ in (8)	0.5
	Stride k between input frames in motion subnet	5
	Initial learning rate	1×10^{-5}
	Training epochs (iterations)	$12(\sim 1.5 \times 10^5)$
	Batch size	12
	Weight decay	5×10^{-6}
SS-ConvLSTM	Bayesian dropout rates p_h and p_f	0.75 & 0.75
	Times of Monte Carlo integration L	100
	Initial learning rate	1×10^{-4}
	Training epochs (iterations)	$15(\sim 2 \times 10^5)$
	Weight decay	5×10^{-6}

5.2 Evaluation on Our Database

In this section, we compare the video saliency prediction accuracy of our DeepVS method and to other state-of-the-art methods, including GBVS [11], PQFT [9], Rudoy [31], OBDL [12], SALICON [13], Xu [37], BMS [39] and SalGAN [28]. Among these methods, [9, 11, 12, 31] and [37] are 5 state-of-the-art saliency prediction methods for videos. Moreover, we compare two latest DNN-based methods: [13, 28]. Note that other DNN-based methods on video saliency prediction [1, 2, 23] are not compared in our experiments, since their codes are not public. In our experiments, we apply four metrics to measure the accuracy of saliency prediction: the area under the receiver operating characteristic curve (AUC), normalized scanpath saliency (NSS), CC, and KL divergence. Note that larger values of AUC, NSS or CC indicate more accurate prediction of saliency, while a smaller KL divergence means better saliency prediction. Table 2 tabulates the results of AUC, NSS, CC and KL divergence for our method and 8 other methods, which are averaged over the 41 test videos of our eye-tracking

database. As shown in this table, our DeepVS method performs considerably better than all other methods in terms of all 4 metrics. Specifically, our method achieves at least 0.01, 0.51, 0.12 and 0.33 improvements in AUC, NSS, CC and KL, respectively. Moreover, the two DNN-based methods, SALICON [13] and SalGAN [28], outperform other conventional methods. This verifies the effectiveness of saliency-related features automatically learned by DNN. Meanwhile, our method is significantly superior to [13, 28]. The main reasons for this result are as follows. (1) Our method embeds the objectness subnet to utilize objectness information in saliency prediction. (2) The object motion is explored in the motion subnet to predict video saliency. (3) The network of SS-ConvLSTM is leveraged to model saliency transition across video frames. Section 5.4 analyzes the above three reasons in more detail.

Table 2. Mean (standard deviation) of saliency prediction accuracy for our and 8 other methods over all test videos in our database.

	Ours	GBVS [11]	PQFT [9]	Rudoy [31]	OBDL [12]	SALICON* [13]	Xu [37]	BMS [39]	SalGAN* [28]
AUC	0.90 (0.04)	0.84(0.06)	0.70(0.08)	0.80(0.08)	0.80(0.09)	0.89(0.06)	0.83(0.06)	0.76(0.09)	0.87(0.06)
NSS	2.94 (0.85)	1.54(0.74)	0.69(0.46)	1.45(0.64)	1.54(0.84)	2.43(0.87)	1.47(0.47)	0.98(0.48)	2.39(0.59)
CC	0.57 (0.12)	0.32(0.13)	0.14(0.08)	0.32(0.14)	0.32(0.16)	0.43(0.13)	0.38(0.11)	0.21(0.09)	0.45(0.09)
KL	1.24 (0.39)	1.82(0.39)	2.46(0.39)	2.42(1.53)	2.05(0.74)	1.57(0.42)	1.65(0.30)	2.23(0.39)	1.62(0.33)

* DNN-based methods have been fine-tuned by our database with their default settings.



Fig. 6. Saliency maps of 8 videos randomly selected from the test set of our eye-tracking database. The maps were yielded by our and 8 other methods as well the ground-truth human fixations. Note that the results of only one frame are shown for each selected video.

Next, we compare the subjective results in video saliency prediction. Figure 6 demonstrates the saliency maps of 8 randomly selected videos in the test set, detected by our DeepVS method and 8 other methods. In this figure, one frame is selected for each video. As shown in Fig. 6, our method is capable of well locating the salient regions, which are close to the ground-truth maps of human fixations. In contrast, most of the other methods fail to accurately predict the regions that attract human attention.

5.3 Evaluation on Other Databases

To evaluate the generalization capability of our method, we further evaluate the performance of our method and 8 other methods on two widely used databases, SFU [10] and DIEM [25]. In our experiments, the models of OM-CNN and SS-ConvLSTM, learned from the training set of our eye-tracking database, are directly used to predict the saliency of test videos from the DIEM and SFU databases. Table 3 presents the average results of AUC, NSS, CC and KL for our method and 8 other methods over SFU and DIEM. As shown in this table, our method again outperforms all compared methods, especially in the DIEM database. In particular, there are at least 0.05, 0.57, 0.11 and 0.34 improvements in AUC, NSS, CC and KL, respectively. Such improvements are comparable to those in our database. This demonstrates the generalization capability of our method in video saliency prediction.

Table 3. Mean (standard deviation) values for saliency prediction accuracy of our and other methods over SFU and DIEM databases.

		SFU								
		Ours	GBVS [11]	PQFT [9]	Rudoy [31]	OBDL [12]	SALICON* [13]	Xu [37]	BMS [39]	SalGAN* [28]
AUC	0.81 (0.07)	0.76(0.07)	0.61(0.09)	0.73(0.08)	0.74(0.10)	0.78(0.08)	0.80(0.07)	0.66(0.08)	0.79(0.07)	
NSS	1.46 (0.65)	0.91(0.47)	0.31(0.34)	0.83(0.45)	1.03(0.64)	1.24(0.60)	1.24(0.39)	0.50(0.31)	1.25(0.47)	
CC	0.55 (0.15)	0.44(0.15)	0.12(0.15)	0.34(0.15)	0.42(0.21)	0.58(0.22)	0.43(0.12)	0.25(0.11)	0.51(0.13)	
KL	0.67(0.24)	0.61 (0.19)	0.98(0.27)	0.93(0.36)	0.80(0.33)	1.12(1.76)	1.35(0.25)	0.83(0.20)	0.70(0.25)	
		DIEM								
		Our	GBVS [11]	PQFT [9]	Rudoy [31]	OBDL [12]	SALICON* [13]	Xu [37]	BMS [39]	SalGAN* [28]
AUC	0.86 (0.08)	0.81(0.09)	0.71(0.11)	0.80(0.11)	0.75(0.14)	0.79(0.11)	0.80(0.11)	0.77(0.11)	0.81(0.08)	
NSS	2.25 (1.16)	1.21(0.82)	0.86(0.71)	1.40(0.83)	1.26(1.03)	1.68(1.04)	1.34(0.74)	1.20(0.80)	1.60(0.71)	
CC	0.49 (0.21)	0.30(0.18)	0.19(0.14)	0.38(0.20)	0.29(0.22)	0.36(0.19)	0.35(0.17)	0.28(0.17)	0.35(0.13)	
KL	1.30 (0.55)	1.64(0.48)	1.73(0.44)	2.33(2.05)	2.77(1.58)	1.66(0.58)	1.67(0.39)	1.96(1.13)	1.64(0.41)	

* DNN-based methods have been fine-tuned by our database with their default settings.

5.4 Performance Analysis of DeepVS

Performance Analysis of Components. Depending on the independently trained models of the objectness subnet, motion subnet and OM-CNN, we further analyze the contribution of each component for saliency prediction accuracy in DeepVS, i.e., the combination of OM-CNN and SS-ConvLSTM. The comparison results are shown in Fig. 7. We can see from this figure that OM-CNN performs

better than the objectness subnet with a 0.05 reduction in KL divergence, and it outperforms the motion subnet with a 0.09 KL divergence reduction. Similar results hold for the other metrics of AUC, CC and NSS. These results indicate the effectiveness of integrating the subnets of objectness and motion. Moreover, the combination of OM-CNN and SS-ConvLSTM reduces the KL divergence by 0.09 over the single OM-CNN architecture. Similar results can be found for the other metrics. Hence, we can conclude that SS-ConvLSTM can further improve the performance of OM-CNN due to exploring the temporal correlation of saliency across video frames.

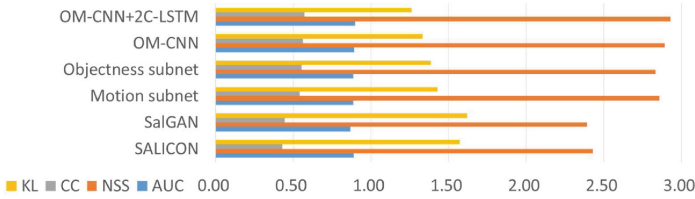


Fig. 7. Saliency prediction accuracy of objectness subnet, motion subnet, OM-CNN and the combination of OM-CNN and SS-ConvLSTM (i.e., DeepVS), compared with SALICON [13] and SalGAN [28]. Note that the smaller KL divergence indicates higher accuracy in saliency prediction.

Performance Analysis of SS-ConvLSTM. We evaluate the performance of the proposed CB dropout of SS-ConvLSTM. To this end, we train the SS-ConvLSTM models at different values of hidden dropout rate p_h and feature dropout rate p_f , and then test the trained SS-ConvLSTM models over the validation set. The averaged KL divergences are shown in Fig. 8(a). We can see that the CB dropout can reduce KL divergence by 0.03 when both p_h and p_f are set to 0.75, compared to the model without CB dropout ($p_h = p_f = 1$). Meanwhile, the KL divergence sharply rises by 0.08, when both p_h and p_f decrease from 0.75 to 0.2. This is caused by the under-fitting issue, as most connections in SS-ConvLSTM are dropped. Thus, p_h and p_f are set to 0.75 in our model. The SS-ConvLSTM model is trained for a fixed video length ($T = 16$). We further evaluate the saliency prediction performance of the trained SS-ConvLSTM model over variable-length videos. Here, we test the trained SS-ConvLSTM model over the validation set, the videos of which are clipped at different lengths. Figure 8(b) shows the averaged KL divergences for video clips at various lengths. We can see that the performance of SS-ConvLSTM is even a bit better, when the video length is 24 or 32. This is probably because the well-trained LSTM cell is able to utilize more inputs to achieve a better performance for video saliency prediction.

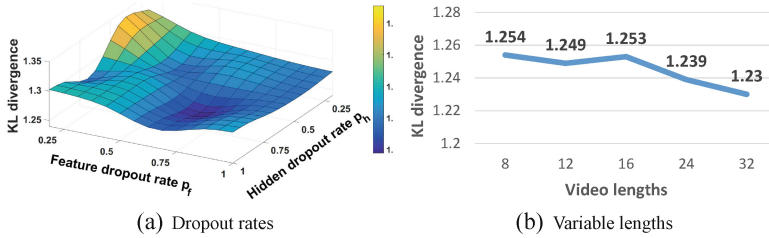


Fig. 8. (a): KL divergences of our models with different dropout rates. (b): KL divergences over test videos with variable lengths.

6 Conclusion

In this paper, we have proposed the DeepVS method, which predicts video saliency through OM-CNN and SS-ConvLSTM. For training the DNN models of OM-CNN and SS-ConvLSTM, we established the LEDOV database, which has the fixations of 32 subjects on 538 videos. Then, the OM-CNN architecture was proposed to explore the spatio-temporal features of the objectness and object motion to predict the intra-frame saliency of videos. The SS-ConvLSTM architecture was developed to model the inter-frame saliency of videos. Finally, the experimental results verified that DeepVS significantly outperforms 8 other state-of-the-art methods over both our and other two public eye-tracking databases, in terms of AUC, CC, NSS, and KL metrics. Thus, the prediction accuracy and generalization capability of DeepVS can be validated.

Acknowledgment. This work was supported by the National Nature Science Foundation of China under Grant 61573037 and by the Fok Ying Tung Education Foundation under Grant 151061.

References

1. Bak, C., Kocak, A., Erdem, E., Erdem, A.: Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Trans. Multimed.* (2017)
2. Bazzani, L., Larochele, H., Torresani, L.: Recurrent mixture density network for spatiotemporal visual attention (2017)
3. Chaabouni, S., Benois-Pineau, J., Amar, C.B.: Transfer learning with deep networks for saliency prediction in natural video. In: *ICIP*, pp. 1604–1608. IEEE (2016)
4. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.M.: Global contrast based salient region detection. *IEEE PAMI* **37**(3), 569–582 (2015)
5. Deng, X., Xu, M., Jiang, L., Sun, X., Wang, Z.: Subjective-driven complexity control approach for HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **26**(1), 91–106 (2016)
6. Dosovitskiy, A., et al.: FlowNet: learning optical flow with convolutional networks. In: *ICCV*, pp. 2758–2766 (2015)

7. Fang, Y., Lin, W., Chen, Z., Tsai, C.M., Lin, C.W.: A video saliency detection model in compressed domain. *IEEE TCSVT* **24**(1), 27–38 (2014)
8. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: *NIPS*, pp. 1019–1027 (2016)
9. Guo, C., Zhang, L.: A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE TIP* **19**(1), 185–198 (2010)
10. Hadizadeh, H., Enriquez, M.J., Bajic, I.V.: Eye-tracking database for a set of standard video sequences. *IEEE TIP* **21**(2), 898–903 (2012)
11. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: *NIPS*, pp. 545–552 (2006)
12. Hossein Khatoonabadi, S., Vasconcelos, N., Bajic, I.V., Shan, Y.: How many bits does it take for a stimulus to be salient? In: *CVPR*, pp. 5501–5510 (2015)
13. Huang, X., Shen, C., Boix, X., Zhao, Q.: SALICON: reducing the semantic gap in saliency prediction by adapting deep neural networks. In: *ICCV*, pp. 262–270 (2015)
14. T. T. INC.: Tobii TX300 eye tracker. <http://www.tobii.com/product-listing/tobii-pro-tx300/>
15. Itti, L., Baldi, P.: Bayesian surprise attracts human attention. *Vis. Res.* **49**(10), 1295–1306 (2009)
16. Itti, L., Dhavale, N., Pighin, F.: Realistic avatar eye and head animation using a neurobiological model of visual attention. *Opt. Sci. Technol.* **64**, 64–78 (2004)
17. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: *ICCV*, pp. 2106–2113 (2009)
18. Kruthiventi, S.S., Ayush, K., Babu, R.V.: DeepFix: a fully convolutional neural network for predicting human eye fixations. *IEEE TIP* (2017)
19. Leboran, V., Garcia-Diaz, A., Fdez-Vidal, X.R., Pardo, X.M.: Dynamic whitening saliency. *IEEE PAMI* **39**(5), 893–907 (2017)
20. Lee, S.H., Kim, J.H., Choi, K.P., Sim, J.Y., Kim, C.S.: Video saliency detection based on spatiotemporal feature learning. In: *ICIP*, pp. 1120–1124 (2014)
21. Li, S., Xu, M., Ren, Y., Wang, Z.: Closed-form optimization on saliency-guided image compression for HEVC-MSP. *IEEE Trans. Multimed.* (2017)
22. Li, S., Xu, M., Wang, Z., Sun, X.: Optimal bit allocation for CTU level rate control in HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **27**(11), 2409–2424 (2017)
23. Liu, Y., Zhang, S., Xu, M., He, X.: Predicting salient face in multiple-face videos. In: *CVPR*, July 2017
24. Mathe, S., Sminchisescu, C.: Actions in the eye: dynamic gaze datasets and learnt saliency models for visual recognition. *IEEE PAMI* **37**(7), 1408–1424 (2015)
25. Mital, P.K., Smith, T.J., Hill, R.L., Henderson, J.M.: Clustering of gaze during dynamic scene viewing is predicted by motion. *Cogn. Comput.* **3**(1), 5–24 (2011)
26. Nguyen, T.V., Xu, M., Gao, G., Kankanhalli, M., Tian, Q., Yan, S.: Static saliency vs. dynamic saliency: a comparative study. In: *ACMM*, pp. 987–996. *ACM* (2013)
27. Palazzi, A., Solera, F., Calderara, S., Alletto, S., Cucchiara, R.: Learning where to attend like a human driver. In: *Intelligent Vehicles Symposium (IV)*, 2017 *IEEE*, pp. 920–925. *IEEE* (2017)
28. Pan, J., et al.: SalGAN: visual saliency prediction with generative adversarial networks. In: *CVPR workshop*, January 2017
29. Pan, J., Sayrol, E., Giro-i Nieto, X., McGuinness, K., O’Connor, N.E.: Shallow and deep convolutional networks for saliency prediction. In: *CVPR*, pp. 598–606 (2016)

30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: CVPR, pp. 779–788 (2016)
31. Rudoy, D., Goldman, D.B., Shechtman, E., Zelnik-Manor, L.: Learning video saliency from human gaze using candidate selection. In: CVPR, pp. 1147–1154 (2013)
32. Wang, L., Wang, L., Lu, H., Zhang, P., Ruan, X.: Saliency detection with recurrent fully convolutional networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 825–841. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_50
33. Wang, W., Shen, J., Guo, F., Cheng, M.M., Borji, A.: Revisiting video saliency: a large-scale benchmark and a new model (2018)
34. Wang, W., Shen, J., Shao, L.: Consistent video saliency using local gradient flow optimization and global refinement. *IEEE Trans. Image Process.* **24**(11), 4185–4196 (2015)
35. Wang, W., Shen, J., Shao, L.: Video salient object detection via fully convolutional networks. *IEEE TIP* (2017)
36. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: NIPS, pp. 802–810 (2015)
37. Xu, M., Jiang, L., Sun, X., Ye, Z., Wang, Z.: Learning to detect video saliency with HEVC features. *IEEE TIP* **26**(1), 369–385 (2017)
38. Xu, M., Liu, Y., Hu, R., He, F.: Find who to look at: turning from action to saliency. *IEEE Transactions on Image Processing* **27**(9), 4529–4544 (2018)
39. Zhang, J., Sclaroff, S.: Exploiting surroundedness for saliency detection: a boolean map approach. *IEEE PAMI* **38**(5), 889–902 (2016)
40. Zhang, L., Tong, M.H., Cottrell, G.W.: SUNDAy: saliency using natural statistics for dynamic analysis of scenes. In: Annual Cognitive Science Conference, pp. 2944–2949 (2009)
41. Zhang, Q., Wang, Y., Li, B.: Unsupervised video analysis based on a spatiotemporal saliency detector. arXiv preprint (2015)
42. Zhou, F., Bing Kang, S., Cohen, M.F.: Time-mapping using space-time saliency. In: CVPR, pp. 3358–3365 (2014)