International Journal of Computer Vision (2021) 129:203–224
https://doi.org/10.1007/s11263-020-01371-6

DeepVS2.0: A Saliency-Structured Deep Learning Method for Predicting Dynamic Visual Attention

Lai Jiang[1,2] · **Mai Xu**[1] · **Zulin Wang**[1] · **Leonid Sigal**[2]

Received: 30 July 2018 / Accepted: 12 August 2020 / Published online: 28 August 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Deep neural networks (DNNs) have exhibited great success in image saliency prediction. However, few works apply DNNs to predict the saliency of generic videos. In this paper, we propose a novel DNN-based video saliency prediction method, called DeepVS2.0. Specifically, we establish a large-scale eye-tracking database of videos (LEDOV), which provides sufficient data to train the DNN models for predicting video saliency. Through the statistical analysis of LEDOV, we find that human attention is normally attracted by objects, particularly moving objects or the moving parts of objects. Accordingly, we propose an object-to-motion convolutional neural network (OM-CNN) in DeepVS2.0 to learn spatio-temporal features for predicting the intra-frame saliency via exploring the information of both objectness and object motion. We further find from our database that human attention has a temporal correlation with a smooth saliency transition across video frames. Therefore, a saliency-structured convolutional long short-term memory network (SS-ConvLSTM) is developed in DeepVS2.0 to predict inter-frame saliency, using the extracted features of OM-CNN as the input. Moreover, the center-bias dropout and sparsity-weighted loss are embedded in SS-ConvLSTM, to consider the center-bias and sparsity of human attention maps. Finally, the experimental results show that our DeepVS2.0 method advances the state-of-the-art video saliency prediction.

Keywords Deep neural networks · Saliency prediction · Convolutional LSTM · Eye-tracking database · Video · Video database

1 Introduction

A foveation mechanism (Matin 1974) 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 detection has been widely studied in recent years, with multiple applications (Borji and Itti 2013) in object recognition, object segmentation, action recognition, image captioning, and image/video compression, among others. Typically, saliency detection can be classified into saliency prediction (Itti et al. 1998) and salient object detection (Liu et al. 2011). Saliency prediction refers to the task of predicting human fixations, while salient object detection aims at detecting/segmenting the most salient objects in a scene. In this paper, we focus on video saliency prediction, which models pixel level attention on each video frame.

In the early stages of this field, the traditional video saliency prediction methods mainly followed integration theory (Itti et al. 2004; Ren et al. 2013; Nguyen et al. 2013;

Communicated by Antonio Torralba.

This work was supported by the NSFC Projects 61922009, 61876013 and 61573037.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11263-020-01371-6) contains supplementary material, which is available to authorized users.

✉ Mai Xu
Maixu@buaa.edu.cn

Lai Jiang
jianglai.china@buaa.edu.cn

Zulin Wang
wzulin@buaa.edu.cn

Leonid Sigal
lsigal@cs.ubc.ca

[1] School of Electronic and Information Engineering, Beihang University, Beijing, China

[2] Department of Computer Science, University of British Columbia, Vancouver, BC, Canada

Zhong et al. 2013; Lee et al. 2014; Leboran et al. 2017), i.e., the saliency of video frames can be detected through two steps: (1) extract spatial and temporal features from videos for obtaining conspicuous maps, and (2) conduct a fusion strategy to combine conspicuous maps of different feature channels together for generating saliency maps. Benefiting from the state-of-the-art image saliency prediction methods, a considerable amount of spatial features have been incorporated to predict video saliency (Nguyen et al. 2013; Fang et al. 2014b; Lee et al. 2014). Additionally, some works have focused on designing temporal features for video saliency prediction, mainly in three aspects: motion-based features (Harel et al. 2006; Zhong et al. 2013; Zhou et al. 2014), temporal contrast features (Itti et al. 2004; Zhang et al. 2009; Ren et al. 2013) and compressed domain features (Hossein Khatoonabadi et al. 2015; Xu et al. 2017). For fusing spatial and temporal features, many machine learning algorithms have been utilized, such as support vector machine (SVM) (Huang et al. 2014; Lee et al. 2014; Xu et al. 2017), probabilistic model (Itti and Baldi 2009; Zhang et al. 2009; Rudoy et al. 2013) and phase spectrum analysis (Guo and Zhang 2010; Leboran et al. 2017).

Differing from integration theory, deep neural network (DNN)-based methods have recently been proposed to learn human attention in an end-to-end manner, significantly enhancing the accuracy of image saliency prediction (Kümmerer et al. 2014; Kruthiventi et al. 2017; Huang et al. 2015; Pan et al. 2016; Li et al. 2016; Wang et al. 2016a; Pan et al. 2017; Wang and Shen 2018). However, only a few works have managed to apply DNNs in saliency prediction (Bak et al. 2017; Bazzani et al. 2017; Wang et al. 2018), or salient object detection (Le and Sugimoto 2017; Li et al. 2018) for videos. Specifically, Bak et al. (2017) has applied a two-stream CNN structure taking both RGB frames and motion maps as the inputs for video saliency prediction. Bazzani et al. (2017) has 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. However, the above DNN-based methods for video saliency prediction are still in their infancy due to the following drawbacks: (1) insufficient eye-tracking data for training the DNN, (2) lack of a sophisticated network architecture that learns to simultaneously combine object and motion information, and (3) neglect of dynamic pixel-wise transition of video saliency across video frames.

To overcome the above drawbacks, we propose a novel large-scale eye-tracking database and a new DNN-based architecture to predict video saliency. Our DNN-architectural design is motivated by extensive experiments conducted on the data of 32 subjects viewing a total of 538 diverse videos. Our model takes into account a core observation that people tend to attend to moving objects or moving parts of objects, by designing a dual stream network that combines hierarchical spatio-temporal features from both streams to predict saliency maps. In particular, our DNN-architecture consists of two structural components: (1) a spatio-temporal object-to-motion convolutional neural network (OM-CNN) for single frame prediction and (2) dynamic recurrent temporal saliency propagation mechanism, which is also called saliency-structured convolutional long short-term memory (SS-ConvLSTM). Note that our database and code are available online [1].

This paper extends our conference paper (Jiang et al. 2018) as follows. In this paper, the existing video eye-tracking databases are thoroughly reviewed and compared, where the fixation dropping rates are calculated to evaluate the efficiency of recording fixations. Besides, we provide more details about the establishment of our database, by introducing the apparatus, fixation classification algorithm, calibration and pre-test in our eye-tracking experiment. This paper thoroughly analyzes our eye-tracking database to motivate our DNN method, which is not discussed in Jiang et al. (2018). More importantly, we advance DeepVS in our conference paper (Jiang et al. 2018), by proposing the sparsity-weighted loss, which considers the sparsity prior of saliency in SS-ConvLSTM. Additionally, this paper further updates the backbone structures of the motion and objectness subnets, taking the advantage of more recent YOLOv3 (Redmon and Farhadi 2018) and FlowNet2 (Ilg et al. 2017) models. The advanced architecture, called DeepVS2.0, is effective as verified in the ablation experiment of this paper. In comparison to Jiang et al. (2018), much more experiments are conducted in this paper to analyze and visualize the effect of the key components in our method.

In brief, the main contributions of this paper are summarized as follows.

- We establish an eye-tracking database that consists of 538 videos with diverse content, along with the thorough analysis and findings on our database.
- We propose a novel OM-CNN structure to predict intra-frame saliency, which integrates both objectness and object motion in a uniform deep structure.
- We develop an SS-ConvLSTM network with the center-bias (CB) dropout and sparsity-weighted loss, to learn the saliency transition across inter-frames at the pixel-wise level.

The remainder of this paper is organized as follows. In Sect. 2, we briefly review the related works and eye-tracking

[1] The database and code can be found at https://github.com/remega/LEDOV-eye-tracking-database and https://github.com/remega/OMCNN_2CLSTM, respectively.

databases for video saliency prediction. In Sect. 3, we establish and analyze our large-scale eye-tracking database. Based on the findings from our database, we propose a DNN for video saliency prediction in Sect. 4, including OM-CNN and SS-ConvLSTM. Section 5 presents the experimental results for validating the performance of our method. Section 6 concludes this paper.

2 Related Work

In this section, we briefly review the recent works and eye-tracking databases for video saliency prediction.

2.1 Video Saliency Prediction

Most of the traditional methods for video saliency prediction (Itti et al. 2004; Ren et al. 2013; Nguyen et al. 2013; Zhong et al. 2013; Lee et al. 2014) rely on integration theory, and they consist of two main steps: feature extraction and feature fusion. In the image saliency prediction task, many effective spatial features succeed in predicting human attention with either a top-down (Judd et al. 2009; Goferman et al. 2012) or bottom-up (Itti et al. 1998; Cheng et al. 2015) strategy. However, video saliency prediction is more challenging because temporal features also play an important role in drawing human attention. To achieve this, motion-based features (Harel et al. 2006; Zhong et al. 2013; Zhou et al. 2014), temporal difference (Itti et al. 2004; Zhang et al. 2009; Ren et al. 2013) and compressed domain methods (Fang et al. 2014a; Xu et al. 2017) are widely used in the existing works of video saliency prediction. Taking motion as an additional temporal feature, Zhong et al. (2013) proposed predicting video saliency using modified optical flow with a restriction of dynamic consistency. Similarly, Zhou et al. (2014) extended the motion feature by computing center motion, foreground motion, velocity motion and acceleration motion in their saliency prediction method. In addition to motion, other methods (Itti et al. 2004; Zhang et al. 2009; Ren et al. 2013) utilize the temporal changes in videos for saliency prediction via computing the contrast between successive frames. For example, Ren et al. (2013) proposed estimating the temporal difference of each patch by finding the minimal reconstruction error of the sparse representation over the co-located patches of neighboring frames. Similarly, in Zhang et al. (2009), the temporal difference is obtained by adding pre-designed exponential filters to the spatial features of successive frames. Taking advantage of sophisticated video coding standards, the compressed domain features have also been explored as spatio-temporal features for video saliency prediction (Fang et al. 2014a; Xu et al. 2017).

In addition to feature extraction, many works have focused on the fusion strategy to generate video saliency maps. Specifically, a set of probability models were constructed to calculate the posterior/prior beliefs (Itti and Baldi 2009), the joint probability distribution of features (Zhang et al. 2009) and candidate transition probability (Rudoy et al. 2013) in predicting video saliency. Similarly, Li et al. (2010) developed a probabilistic multi-task learning method to incorporate the task-related prior in video saliency prediction. Moreover, other machine learning algorithms, such as SVM and neural network, were also applied to linearly (Nguyen et al. 2013) or non-linearly (Lee et al. 2014) combine the saliency-related features. Other advanced methods (Guo and Zhang 2010; Wang et al. 2016b) apply phase spectrum analysis in the fusion model to bridge the gap between features and video saliency. For instance, Guo and Zhang (2010) applied phase spectrum of quaternion Fourier transform (PQFT) on four feature channels (two color channels, one intensity channel, and one motion channel) to predict video saliency.

Most recently, DNNs have succeeded in many computer vision tasks (Simonyan and Zisserman 2015; Du et al. 2015; Redmon et al. 2016). In the field of saliency prediction, DNNs have also been successfully incorporated to automatically learn spatial features for predicting the saliency of images (Kruthiventi et al. 2017; Huang et al. 2015; Pan et al. 2016; Li et al. 2016; Wang et al. 2016a; Pan et al. 2017; Cornia et al. 2018; Wang and Shen 2018). Specifically, as one of the pioneering works, Deepfix (Kruthiventi et al. 2017) proposed a DNN-based structure on VGG-16 (Simonyan and Zisserman 2015) and inception module (Szegedy et al. 2015) to learn a multi-scale semantic representation for saliency prediction. In Deepfix, a dilated convolutional structure was developed to extend the receptive field, and then a location-biased convolutional layer was proposed to learn the CB pattern for saliency prediction. Similarly, SALICON (Huang et al. 2015) was also proposed to fine tune the existing DNNs for object recognition, in which an efficient loss function was developed for training the DNN model in saliency prediction. Later, some advanced DNN methods (Pan et al. 2016; Li et al. 2016; Wang et al. 2016a; Cornia et al. 2018) were proposed to improve the performance of image saliency prediction.

However, only a few works have managed to apply DNNs in saliency prediction (Chaabouni et al. 2016; Bak et al. 2017; Bazzani et al. 2017; Liu et al. 2017; Palazzi et al. 2017; Wang et al. 2018) or salient object detection (Wang et al. 2017; Le and Sugimoto 2017; Li et al. 2018) for videos. The main reasons are as follows. (1) Different from image saliency, additional dynamic structure needs to be designed in DNNs for video saliency prediction. (2) The eye-tracking data are insufficient for training DNN, since it is costly to record human fixations over videos. In the existing DNNs for video saliency prediction, the dynamic characteristics were explored in two ways: adding temporal information to CNN structures (Chaabouni et al. 2016; Bak et al. 2017) or devel-

oping a dynamic structure with LSTM (Bazzani et al. 2017; Liu et al. 2017). For adding temporal information, a four-layer CNN in Chaabouni et al. (2016) and a two-stream CNN in Bak et al. (2017) were trained to predict video saliency, taking both RGB frames and motion maps as the inputs. Similarly, Li et al. (2018) applied optical flow and object proposal as the pre-processing steps to generate saliency cues for the proposed stacked autoencoders. However, the optical flow methods applied in these models cause heavy computational complexity. In Wang et al. (2017), the pair of video frames, concatenated with a static saliency map (generated by the static CNN), are input into the dynamic CNN for salient object detection, allowing the CNN to generalize temporal features. In this paper, the OM-CNN structure of our method 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, Le and Sugimoto (2017) temporally combined the local features of each frame through a Gaussian averaging, and further trained a C3D network to extract global features of whole video for salient object detection. Similarly, Bazzani et al. (2017) and Liu et al. (2017) applied LSTM networks to predict human attention, relying on both short- and long-term memory. However, the fully-connected units in LSTM limit the dimensions of both the input and output; thus, it is unable to obtain the end-to-end saliency map. As such, in Bazzani et al. (2017) and Liu et al. (2017), strong prior knowledge needs to be assumed for the distribution of saliency. Specifically, in Bazzani et al. (2017), human attention is assumed to distribute as a Gaussian mixture model (GMM); then, an LSTM was constructed to learn the parameters of GMM. Similarly, Liu et al. (2017) focuses on predicting the saliency of conference videos and assumes that the saliency in each face is a Gaussian distribution. In Liu et al. (2017), the face saliency transition across video frames is learned by an LSTM, and the final saliency map is generated via combining the saliency of all faces in the video. In our work, we first explore SS-ConvLSTM with the CB dropout 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.

2.2 Video Eye-Tracking Databases

The eye-tracking databases of videos collect the fixations of subjects on each video frame, which can be used as the ground-truth for video saliency prediction. The existing eye-tracking databases benefit from mature eye-tracking technology. Specifically, an eye tracker is used to obtain the fixations of subjects on videos by tracking the pupil and corneal reflections (Holmqvist et al. 2011). The pupil locations are then mapped to the real-world stimuli, i.e., video frames, through a pre-defined calibration matrix. Consequently, fixations can be located in each video frame, indicating where people pay their attention.

We now review the existing video eye-tracking databases. Table 1 summarizes the basic properties of these databases. It is worth noting that fixation dropping rate r_f is defined to evaluate the efficiency of recording fixations for database by calculating the average missed fixations per subject during recording. Specifically, r_f can be defined as

$$r_f = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{V_c}\sum_{i=1}^{V_c}\frac{P_c - f_c^i}{P_c}, \tag{1}$$

where C is the total number of videos in the database. Additionally, V_c and P_c are the numbers of frames and subjects for the c-th video, respectively. Besides, f_c^i (smaller than P_c) indicates the number of recorded fixations at the i-th frame of video c. In addition to r_f, we also calculate the average number of fixations per frame (ANFF) for each dataset, as listed in Table 1.

Among the databases in Table 1, to the best of our knowledge, CRCNS (Itti 2004), SFU (Hadizadeh et al. 2012), DIEM (Mital et al. 2011) and Hollywood (Mathe and Sminchisescu 2015) are the most popular databases, and they have been widely used in most of the recent video saliency prediction works (Fang et al. 2014b, a; Hossein Khatoonabadi et al. 2015; Rudoy et al. 2013; Nguyen et al. 2013; Zhong et al. 2013; Wang et al. 2016b; Chaabouni et al. 2016; Mauthner et al. 2015). These databases are reviewed in more detail as follows.

CRCNS Itti (2004) is one of the earliest video eye-tracking databases, and it was established by Itti et al. in 2004. This database is still used as a benchmark in the recent video saliency prediction works, such as Fang et al. (2014b). CRCNS contains 50 videos, which mainly include outdoor scenes, TV shows and video games. The length of each video ranges from 5.5 to 93.9 s, and the frame rate of all videos is 30 frames per second (fps). For each video, 4–6 subjects were asked to look at the main actors or actions. Subsequently, they were required to depict the main content of the video. Thus, CRNS is a task-driven eye-tracking database for videos. Later, a new database (Carmi and Itti 2006) was established by manually cutting all 50 videos of CRCNS into 523 "clippets" with a 1–3 s duration, according to the abrupt cinematic cuts. Another 8 subjects were recruited to view these video clippets, and their eye-tracking data were recorded in Carmi and Itti (2006).

SFU Hadizadeh et al. (2012) is a public video database that contains the eye-tracking data of 12 uncompressed YUV videos, which are frequently used as the standard test set for video compression and processing algorithms. Each video is in CIF resolution (352 × 288) and is of 3–10 s at a frame rate of 30 fps. The eye-tracking data were collected from 15 non-

Table 1 The basic properties of the existing eye-tracking databases

Database	Year	Videos	Resolution	Duration (s)	Subjects	r_f^a	ANFF[a]	Mode
CRCNS (Itti 2004)	2004	50	640 × 480	6–94	4.7 (ave)	23.4%	3.6	Task
SFU (Hadizadeh et al. 2012)	2012	12	352 × 288	3–10	15	0.4%	14.9	Free
DIEM (Mital et al. 2011)	2011	84	≤ 1280 × 720	27–217	50 (ave)	13.1%	43.5	Free
Hollywood (Mathe and Sminchisescu 2015)	2015	1857	≤ 720 × 480	2–90	19	30.7%	13.2	Task
(Xu et al. 2017)	2016	32	≤ 1920 × 1080	6–25	32	4.5%	30.6	Free
IRCCyN (Boulos et al. 2009)	2009	51	720 × 576	8–10	37	–	–	Free
VAGBA (Li et al. 2011)	2011	50	1920 × 1080	10	14	24.9%	10.5	Free
GazeCom (Dorr et al. 2010)	2010	18	1280 × 720	20	54	–		Free
ASCMN (Riche et al. 2012)	2012	24	≤ 704 × 576	2–76	13	28.0%	9.4	Free
Coutrot-2 (Coutrot and Guyader 2015)	2015	15	1232 × 504	20–80	40	1.7%	39.2	Free
CAMO (Nguyen et al. 2013)	2013	120	640 × 480	1.7–8.6	10	–	–	Free
Marat (Marat et al. 2007)	2007	53	720 × 576	1–3	15	–	–	Free
SAVAM (Gitman et al. 2014)	2014	43	1920 × 1080	18.1 (ave)	50	3.9%	48.0	Free
TUD (Alers et al. 2012)	2012	50	1280 × 720	20	12	–	–	Task
Peters (Peters and Itti 2007)	2007	24	640 × 480	267	5	–	–	Task
Coutrot-1 (Coutrot and Guyader 2013)	2013	60	720 × 576	10–24.8	20	7.1%	18.6	Task
Our database	2018	538	≥ 1280 × 720	5–60	32	12.8%	27.9	Free

[a] r_f is the fixation dropping rates, and ANFF indicates the average number of fixations per frame. Note that r_f and ANFF of some databases are not calculated because the raw fixation data cannot be downloaded

expert subjects, who were asked to free view all 12 videos twice.

DIEM Mital et al. (2011) is another widely used database, designed to evaluate the contributions of different visual features on gaze clustering. DIEM consists of 84 videos sourced from publicly accessible videos, including advertisements, game trailers, movie trailers and news clips. Most of these videos have frequent cinematic cuts. Each video lasts for 27–217 s at 30 fps. The free-viewing fixations of approximately 50 subjects were tracked for each video.

Hollywood Mathe and Sminchisescu (2015) is a large-scale eye-tracking database for video saliency prediction, which contains all videos from two action recognition databases: Hollywood-2 (Marszalek et al. 2009) and UCF sports (Rodriguez 2010). All of the 1707 videos in Hollywood-2 were selected from 69 movies according to 12 action classes, such as answering phone, eating and shaking hands. UCF sports is another action database that includes 150 videos with 9 sport action classes. The human fixations of 19 subjects were captured under 3 conditions: free viewing (3 subjects), action recognition task (12 subjects), and context recognition task (4 subjects). Although the number of videos in Hollywood is large, its video content is not diverse and is constrained by human actions. Moreover, it mainly focuses on the task-driven viewing mode rather than free viewing.

As discussed in Sect. 2.1, video saliency prediction may benefit from the recent development of deep learning. Unfortunately, as shown in Table 1, the existing databases for video saliency prediction lack sufficient eye-tracking data for training DNNs. Although Hollywood (Mathe and Sminchisescu 2015) contains 1857 videos, it mainly focuses on task-driven visual saliency, and its fixation dropping rate is rather large. Moreover, the video content of Hollywood is limited, only involving human actions in movies. In fact, a large-scale eye-tracking database for video should satisfy 3 criteria: (1) a large number of videos, (2) sufficient fixations per frame, and (3) varied video content. To this end, we establish a large-scale eye-tracking database of videos that satisfies the above three criteria. The details about our database are discussed in Sect. 3.

3 Database

In this section, a new large-scale eye-tracking database of videos (LEDOV) is established for facilitating future research. The basic information of our database is listed in Table 1 in comparison with the existing eye-tracking databases. Besides, more details and analysis of our LEDOV database are discussed in the following.

3.1 Database Establishment

We present our LEDOV database from the aspects of stimuli, apparatus, participants and procedure.

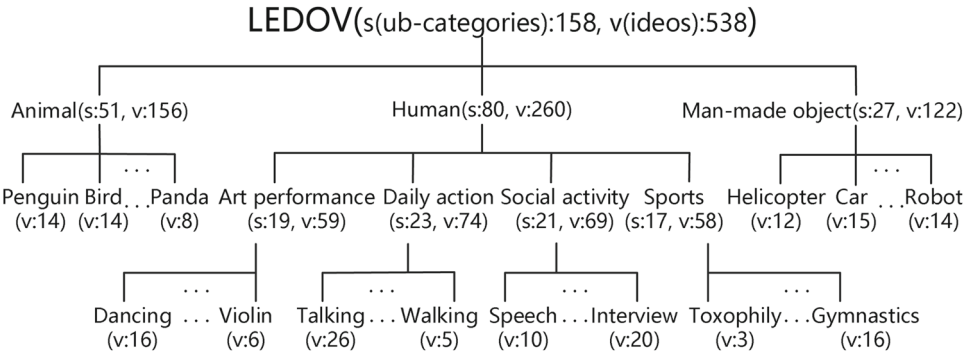


Fig. 1 Category tree of videos in LEDOV according to the content. The numbers of categories/sub-categories are shown in the brackets. Besides, the number of videos for each category/sub-category is also shown in the brackets. Note that the categories/sub-categories are not mutually exclusive in video search. For each video, it is categorized and counted depending on the main object in the video. The videos have multiple kinds of attractive objects are also filtered in the second round of video selection by 3 PhD students

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. 1. In practice, we asked 3 volunteers to make a quick survey on the recent videos of YouTube. Then, some categories were summarized from these videos, referring to the keywords of WordTree in Redmon and Farhadi (2017). After that, the hierarchical tree was manually constructed according to these categories, including 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, e.g., bird, giraffe and wolf. Similarly, the category of man-made objects was composed of 27 sub-categories, such as car and airplane. 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. 1. Given the hierarchical tree of key words on video categories, 10 undergraduate students were recruited to collect 658 videos from different channels of YouTube (such as daily vlogs, documentaries, movies, sport casts, and TV shows.) upon the following 3 criteria. Subsequently, the collected videos were filtered by 3 PhD students also upon the following 3 criteria. Specifically, the downloaded videos were excluded from our database, once vetoed by one PhD student according to the following criteria. Finally, we had 158 sub-categories in total, and then collected 538 videos (with a total of 179,336 frames and 6431 s) belonging to these 158 sub-categories from YouTube. Criteria for selecting videos are as follows.

- *Including at Least One Object* The videos with at least one main object were qualified. Specifically, our database includes 377, 103 and 58 videos with one, two and more than two main objects.
- *High Quality* The videos at low subjective quality were excluded. To further ensure the subjective quality, the resolutions and frame rates of videos were at least 720p and 24 Hz, respectively. Furthermore, the bit rates were maintained when converting the videos to the uniform MP4 format, for avoiding quality degradation.
- *Stable Shot* Videos with unsteady camera motions[2] and frequent cinematic cuts were not included in LEDOV. Specifically, there are 222 videos with stable camera motion. The other 316 videos are without any camera motion.

Apparatus For monitoring the binocular eye movements, a Tobii TX300 eye tracker (Tobii 2017) was used in our experiment. TX300 is an integrated eye tracker with a 23" TFT monitor at a screen resolution of 1920 × 1080. During the experiment, TX300 captured gaze data at 300 Hz. According to Tobii (2017), the gaze accuracy can reach 0.4 vision angle (approximately 15 pixels in stimuli) under ideal working conditions[3]. After recording the raw gaze data, a fixation classification algorithm called the I-VT filter (Olsen 2012) was applied to filter out fixations from other eye movements, such as saccades and smooth pursuits. In the I-VT filter, eye movements were classified based on the velocity of the directional shifts of the eyes. For more details about Tobii TX300 and the I-VT filter, refer to Tobii (2017) and Olsen (2012).

Calibration and pre-test To ensure the eye-tracking precision, each subject was required for a 9-point calibration embedded in Tobii TX300. Additionally, we design a pre-test to select the qualified subjects who have small fixation dropping rate. In the pre-test, each subject was asked to view 15 videos, and once the dropping rate of fixations was above

[2] The videos with regular camera motions, such as pans, tilts and zooms, were not excluded unless the motion was too intensive.

[3] The ideal conditions are that the illumination in the working environment is constant at 300 lux, and that the distance between subjects and the monitor is fixed at 65 cm. Such conditions were satisfied in our eye-tracking experiment.

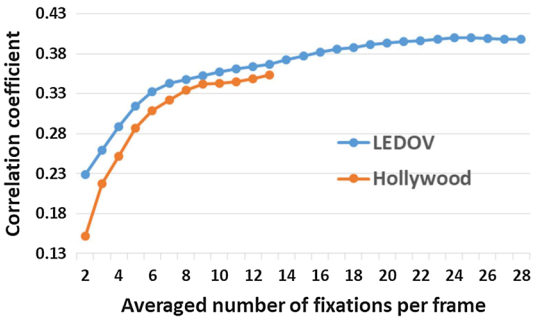


Fig. 2 Averaged CC values alongside increased numbers of fixations per frame over LEDOV and Hollywood. For one frame, the CC of fixation maps is measured between each fixation and the reminder

15%, the subject cannot participate in the following eye-tracking experiment. Consequently, 32 subjects among 60 subjects passed the 9-point calibration and pre-test. It is worth mentioning that the dropping rate of fixations reaches 12.8% in our database, which is considerably lower than other large-scale databases as listed in Table 1.

Procedure Since visual fatigue may arise after viewing videos for a long time, the 538 videos in LEDOV were equally divided into 6 non-overlapping groups with similar numbers of videos in terms of content (i.e., human, animal and man-made object). During the experiment, each subject was seated on an adjustable chair at approximately 65 cm from the screen, followed by a 9-point calibration. Then, the subject was required to free view the 6 groups of videos in a random order. In each group, the videos were also displayed at random. Between two successive videos, we inserted a 3-s rest period with a black screen and a 2-s guidance image with a red circle in the screen center. Thus, the eyes can be relaxed, and then the initial gaze location can be reset to center. The eye-tracking experiment of one subject lasted around 30 min for each video group, and all videos were divided into 6 video groups. Therefore, the total viewing time for each subject was about 3 h. Between two groups, the subjects were also required to have a rest between viewing two groups, in order to avoid eye fatigue. Note that the whole experiment of one subject may be divided into a couple of days.

Participants Similar to Bylinskii et al. (2018), Kim et al. (2017), we conducted a specific scheme for determining the sufficient number of participants. We stopped recruiting subjects for eye-tracking experiments once recorded fixations converged. Specifically, after recording eye-tracking data of each subject, we calculated the consistency of that subject's fixations with an existing pool of data to determine whether additional subjects were needed. For one frame, the consistency was measured by the linear correlation coefficient (CC) of fixation maps between each fixation and the reminder. Figure 2 shows the averaged CC values at different numbers of fixations per frame over all videos. Note that the fixation number is increased along with more subjects participating in the experiment. We can see from Fig. 2 that the CC value converges at the fixation number of 28, i.e., 32 subjects with a dropping rate of 12.8% are enough in our experiment. Finally, 5,058,178 fixations of all 32 subjects (18 males and 14 females) on 538 videos were collected for our eye-tracking database. Figure 2 also shows that the 19 subjects (dropping rate: 30.7%) of the Hollywood database are not enough. It is worth mentioning that the previous databases did not investigate the sufficient numbers of subjects in their eye-tracking experiments.

3.2 Database Analysis

In this section, we mine our database to analyze human attention on videos. First, following (Rajashekar et al. 2008), the fixation map **G** is generated by convoluting the fixations with a Gaussian mask **N**. The Gaussian mask **N** is defined with an assumption that the mask value decreases to half-max at the boundary of the fovea region. Specifically, the Gaussian mask can be computed as

$$\mathbf{N}(i,j) = \frac{1}{2\pi\sigma_n}\exp\left(\frac{-d_{ij}^2}{2\sigma_n^2}\right), \quad \text{where} \quad \sigma_n = \frac{A \cdot P}{\sqrt{2\ln(2)}}. \quad (2)$$

In (2), d_{ij} is the distance between pixel (i, j) and the center of Gaussian mask; A refers to the visual angle of the fovea ($= 1.5°$ in our work); P ($= 40$ in our work) indicates the number of pixels for each visual angle. Consequently, σ_n of our the Gaussian mask is around 25, and the size of the mask is set to 100×100. Finally, the fixation maps **G** can be obtained upon fixations and Gaussian mask **N** from (2), which are also regarded as the ground-truth maps in this paper.

3.2.1 Correlation Between Objectness and Human Attention

It is intuitive that people may be attracted to objects rather than background when viewing videos. Therefore, we investigate how much attention is related to object regions. First, we apply a CNN-based object detection method named YOLO (Redmon et al. 2016) to detect the main objects in each video frame. Here, we generate different numbers of candidate objects in YOLO, via changing thresholds of confidence probability and non-maximum suppression. Figure 3b shows examples of one detected object, while Fig. 3c shows the results for more than one object. We can observe from Fig. 3b that attention is normally attended to object regions. We can also observe from Fig. 3c that more human fixations can be included along with an increased number of detected candidate objects. To quantify the correlation between human attention and objectness, we measure the

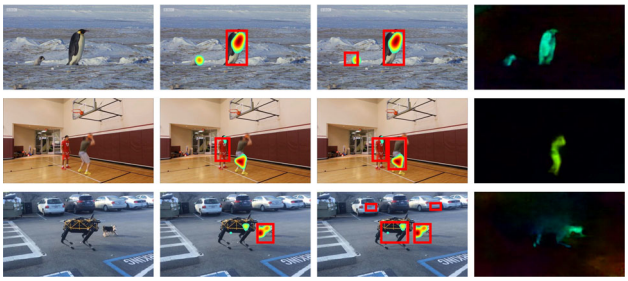


Fig. 3 Examples of ground-truth fixation maps and candidate objects detected by YOLO. **a** Shows randomly selected frames from three videos in our LEDOV database. **b** Illustrates the fixation maps as well as one candidate object in each frame. **c** Demonstrates fixation maps and multiple candidate objects. **d** Displays optical flow maps of each frame, represented in HSV color space

proportion of fixations falling into object regions to those into all regions. In Fig. 4a, we show the fixation proportion at an increased number of candidate objects, averaged over all videos in LEDOV. As shown in this figure, the proportion of fixation hitting object regions is much higher than that hitting random regions. This implies that a high correlation exists between objectness and human attention when viewing videos. Figure 4a also shows that the fixation proportion increases alongside more candidate objects, which indicating that human attention may be attracted by more than one object.

In addition, one may find from Fig. 3 that human attention is attended to only small parts of object regions. Therefore, we measure the proportion that fixation area (inside the object regions) occupies the entire object area. Note that the fixation area is obtained through a pre-set threshold of 0.5 on the fixation map. This threshold is consistent with the way we generate fixation map, which is introduced in Sect. 3.2. According to (2), the regions with values over 0.5 in our fixation map are roughly equal to the retinal fovea of people when viewing the video. We can see from Fig. 4b that the proportions of all fixation areas for different numbers of objects are far from 100%, meaning that human attention is attended to only small parts of objects. Besides, compared to random regions, the proportions of fixation areas for objects are higher, again indicating that the objects are easy to attract human attention.

3.2.2 Correlation Between Motion and Human Attention

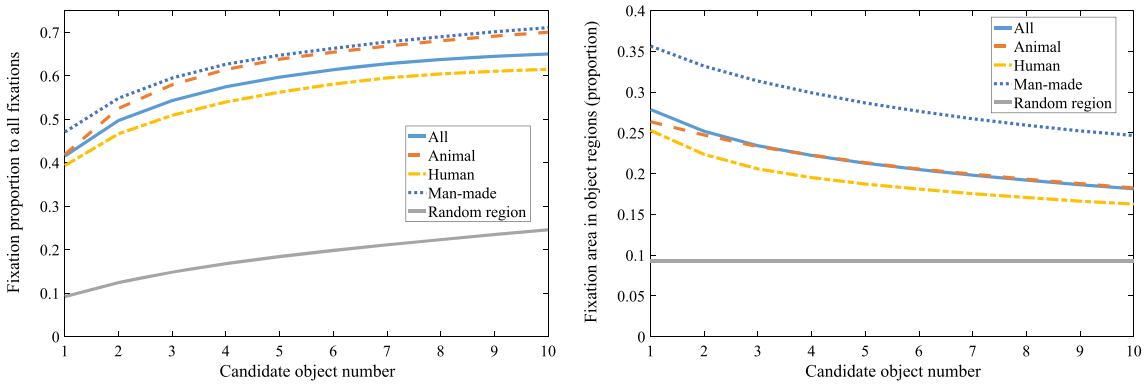
From our LEDOV database, we find that human attention tends to focus on moving objects or the moving parts of objects. Specifically, as shown in the first row of Fig. 5, human attention is transferred to the large penguin when it suddenly falls with a rapid motion. Additionally, the second row of Fig. 5 shows that in the scene with a single salient object, the intensive moving parts of the player may attract considerably more fixations than the other parts. It is interesting to further explore the correlation between motion and human attention inside the regions of objects. Here, we apply FlowNet (Dosovitskiy et al. 2015), a DNN-based optical flow method, to measure the motion intensity in all frames (some results are shown in Fig. 3d). At each frame, pixels are ranked according to the descending order of motion intensity. Subsequently, we cluster the ranked pixels into 10 groups with equal number of pixels over all video frames in the LEDOV database. For example, the first group includes pixels with top 10% ranked motion intensity. The numbers of fixations falling into each group are shown in Fig. 6. We can see from Figure 6 that 44.9% of the fixations belong to the group with the top 10% high-valued motion intensity. This result implies a high correlation between motion and human attention within the region of objects.

3.2.3 Temporal Correlation of Attention on Consecutive Frames

It is interesting to explore the temporal correlation of attention across consecutive frames. In Fig. 7, we show human fixation maps along with some consecutive frames for 3 selected videos. As shown in Fig. 7, there exists a high temporal correlation of attention across consecutive frames of videos. To quantify this correlation, we further measure CC of the fixation map between two consecutive frames. Assume that $\mathbf{G}_c$ and $\mathbf{G}_p$ are fixation maps of the current and previous frames, respectively. Then, the CC value of fixation maps averaged over a video can be calculated as follows:

$$\begin{aligned} &\mathrm{CC}(\mathbf{G}_\mathrm{c}, \mathbf{G}_\mathrm{p}) \\ &= \frac{1}{|\mathbf{V}_c|} \sum_{c \in \mathbf{V}_c} \frac{1}{|\mathbf{V}_p|} \sum_{p \in \mathbf{V}_p} \frac{\mathrm{Cov}(\mathcal{N}(\mathbf{G}_\mathrm{c}), \mathcal{N}(\mathbf{G}_\mathrm{p}))}{\mathrm{Std}(\mathcal{N}(\mathbf{G}_\mathrm{c})) \cdot \mathrm{Std}(\mathcal{N}(\mathbf{G}_\mathrm{p}))}, \\ &\text{where} \quad \mathcal{N}(\mathbf{G}_\mathrm{c}) = \frac{\mathbf{G}_\mathrm{c} - \mathrm{Mean}(\mathbf{G}_\mathrm{c})}{\mathrm{Std}(\mathbf{G}_\mathrm{c})}. \end{aligned} \tag{3}$$

In (3), $\mathbf{V}_c$ is the set of all frames in the video, and $\mathbf{V}_p$ is the set of consecutive frames before frame c. Additionally, Cov(·), Std(·) and Mean(·) are covariance, standard deviation and mean operators, respectively. For $\mathbf{V}_\mathrm{p}$, we choose 4 sets of previous frames, i.e., 0–0.5s before, 0.5–1s before, 1–1.5s before and 1.5–2s before. Then, in Fig. 8, we plot the CC results of these 4 sets of $\mathbf{V}_\mathrm{p}$, which are averaged over all videos in our LEDOV database. Figure 8 also shows the CC results of two baselines, i.e., one-vs-all and video content. The one-vs-all baseline calculates the averaged CC between the fixation maps of one subject and the remainder, indicating attention correlation across subjects. The video content baseline indicates the temporal correlation of the input videos, by measuring the averaged CC between the current video frame



(a) Fixation hit proportion **(b)** Fixation area proportion

Fig. 4 **a** Fixation proportion belonging to object regions at increased numbers of detected candidate objects. **b** Proportion of fixation area in object regions along with increased numbers of detected candidate objects. The results of all videos as well as animals, human and man-made object videos are plotted with different curves. Besides, the results of fixations hitting random region are plotted as the baseline

Fig. 5 Attention heat maps of some frames selected from video *animal_penguin07* and *human_basketball02*, where the human attention is attracted by moving objects or the moving parts of objects

and a set of consecutive video frames (0–2 s before). The following two observations are obtained from Fig. 8.

(1) The CC value of temporal attention is considerably higher than that of the one-vs-all baseline, implying a high temporal correlation of attention across consecutive frames of video. This is partly due to high inter-frame consistency of video content with large CC values.
(2) The temporal correlation of attention decreases along with the increased distance between the current and previous frames. Consequently, there exist the long- and short-term dependencies of attention across video frames.

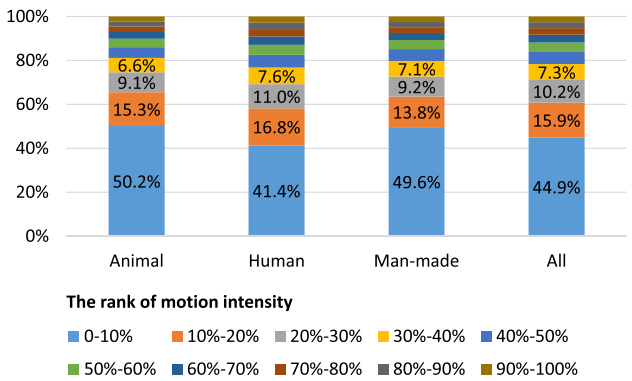
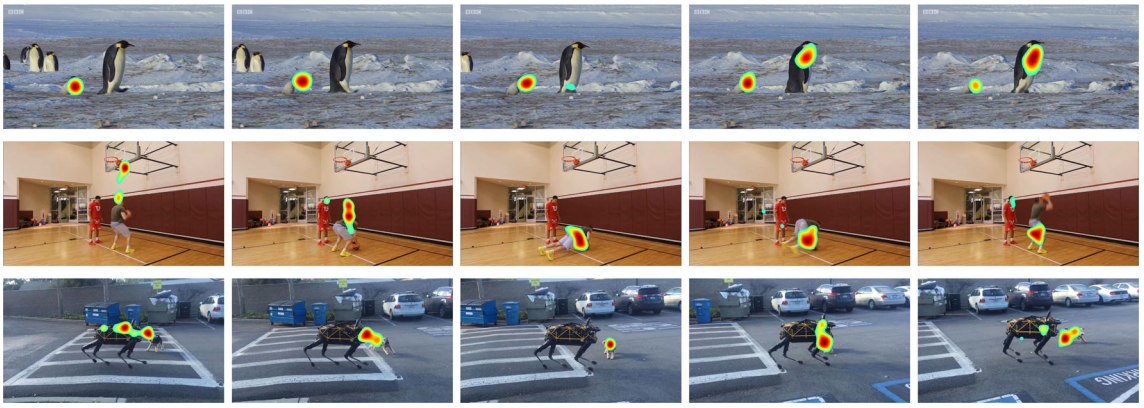


Fig. 6 Proportion of fixations belonging to 10 groups, ranked according to the motion intensity

4 Proposed Method

4.1 Framework

For video saliency prediction, we develop a new DNN architecture called DeepVS2.0 that combines OM-CNN and SS-ConvLSTM. According to the findings in Sects. 3.2.1 and 3.2.2, 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, which weights 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 fea-



(a) 2s before **(b)** 1.5s before **(c)** 1s before **(d)** 0.5s before **(e)** Current frame

Fig. 7 Examples for human fixation maps across consecutive frames

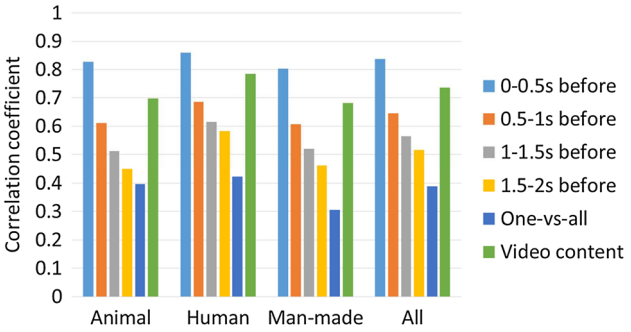


Fig. 8 The CC results of temporal attention correlation averaged over animal, human, man-made object and all videos in LEDOV

ture normalization to generate the spatio-temporal features of OM-CNN. The architecture of OM-CNN is shown in Fig. 9. Besides, SS-ConvLSTM with the CB dropout and sparsity-weighted loss is developed to learn the dynamic saliency of video clips, in which the spatio-temporal features of OM-CNN serve as the input. Then, the saliency map of each frame is generated from 2 *deconvolutional layers* of SS-ConvLSTM. The architecture of SS-ConvLSTM is shown in Fig. 10.

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 YOLOv3 (Redmon and Farhadi 2018). To avoid over-fitting, a pruned structure of YOLOv3, denoted as Darknet-53 in Redmon and Farhadi (2018), is applied as the objectness subnet. In short, it is a residual network with 53 *convolutional layers*. In addition, a *batch-normalization layer* and a *leaky ReLU activation* follow each *convolutional layer*. Assuming that $*$, $BN(\cdot)$ and $L_{0.1}(\cdot)$ are the convolution operation, batch-normalization and leaky ReLU with coefficient of 0.1, the output of the k-th *convolutional layer* $\mathbf{C}_o^k$ in the objectness subnet can be computed as

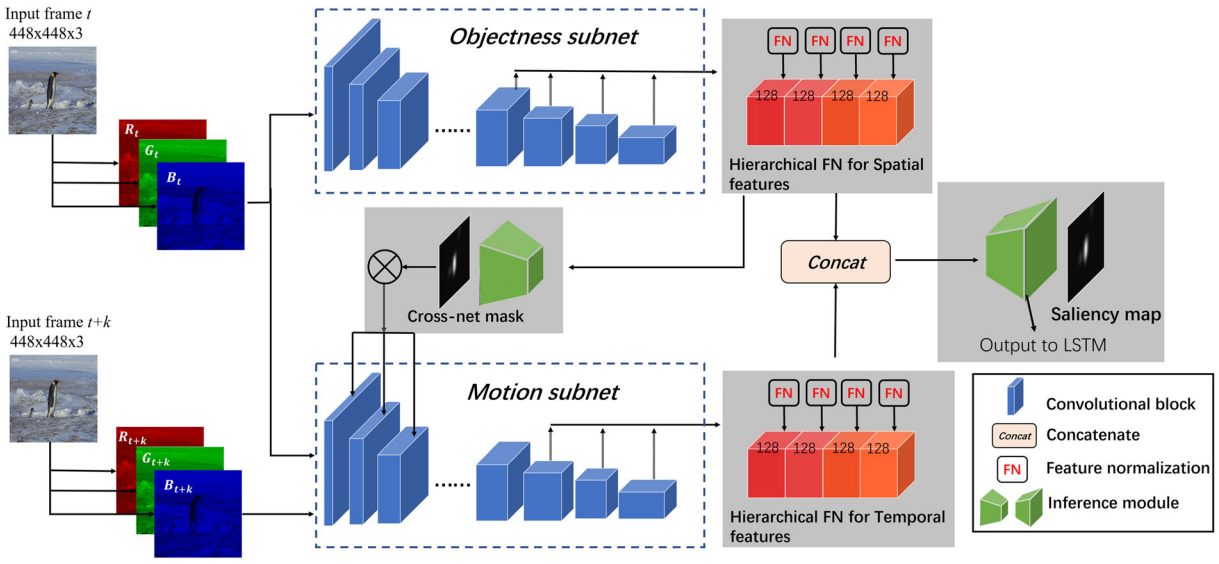
$$\mathbf{C}_o^k = L_{0.1}\left(BN(\mathbf{C}_o^{k-1} * \mathbf{W}_o^{k-1} + \mathbf{B}_o^{k-1})\right), \tag{4}$$

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. In addition to the objectness subnet, a FlowNet2 (Ilg et al. 2017) based motion subnet is also incorporated in OM-CNN to extract multi-scale temporal features from the pair of neighboring frames. Similar to the objectness subnet, the motion subnet is applied by a pruned structure of FlowNet2, denoted as FlowNet2-S in Ilg et al. (2017), with 10 *convolutional layers*. For details about the objectness and motion subnets, refer to Redmon and Farhadi (2018) and Ilg et al. (2017). In the following, we propose some new modules that combine the subnets of objectness and motion.

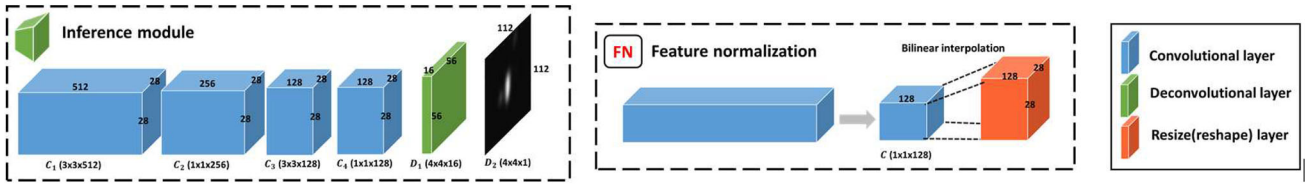
4.3 Combination of Objectness and Motion Subnets

In OM-CNN, the hierarchical FN and the cross-net mask are proposed 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 the hierarchical FN for spatial features as an example. First, the output features of each residual block in the objectness subnet are normalized through the FN module to



(a) The overall architecture of OM-CNN



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

Fig. 9 **a** shows the overall architecture of our OMCNN in DeepVS2.0 for predicting video saliency of intra-frame. The detailed structures of the objectness and motion subnets are based on YOLOv3 and FlowNet2, respectively. 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

obtain 4 sets of spatial features $\{\mathbf{FS}_i\}_{i=1}^4$. As shown in Fig. 9b, each FN module is composed of a *convolutional layer* with 1×1 kernels and a *bilinear layer* to normalize the input features into 128 channels at a resolution of 28×28. All spatial features $\{\mathbf{FS}_i\}_{i=1}^4$ are concatenated in a hierarchy to obtain a total size of $28 \times 28 \times 512$, as the output of the 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 the 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 $\{\mathbf{FS}_i\}_{i=1}^4$ and temporal features $\{\mathbf{FT}_i\}_{i=1}^4$, 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

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

The inference module I_f is a CNN structure that consists of 4 *convolutional layers* and 2 *deconvolutional layers* with a stride of 2. For each *convolutional/deconvolutional layer*, the batch-normalization and leaky ReLU activation are applied, the same as that in (4). For the last *deconvolutional layer*, the sigmoid activation is used instead of the leaky ReLU. The detailed architecture of I_f is shown in Fig. 9b. Consequently, $\mathbf{S}_f$ is used to train the OM-CNN model, as discussed in Sect. 4.5. Additionally, the output of the last *convolutional layer* (C_4 in Fig. 9b, size: $28 \times 28 \times 128$) in the inference module I_f is viewed as the final spatio-temporal features of OM-CNN, denoted by $\mathbf{F}$. Afterwards, $\mathbf{F}$ is fed into SS-ConvLSTM for predicting inter-frame saliency.

Cross-net mask The finding in Sect. 3.2.2 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 $\{\mathbf{FS}_i\}_{i=1}^4$ of the objectness subnet, $\mathbf{S}_c$ can be generated by another inference module I_c as follows,

$$\mathbf{S}_c = I_c(\{\mathbf{FS}_i\}_{i=1}^4). \tag{6}$$

Note that the structure of I_c is same as that of I_f as shown in (5), 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 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

$$\begin{aligned} &\mathbf{C}_m^k = L_{0.1}\left(M(\mathbf{C}_m^{k-1}, \mathbf{S}_c) * \mathbf{W}_m^{k-1} + \mathbf{B}_m^{k-1}\right), \\ &\text{where} \quad \mathrm{M}(\mathbf{C}_\mathrm{m}^{\mathrm{k}-1}, \mathbf{S}_\mathrm{c}) = \mathbf{C}_\mathrm{m}^{\mathrm{k}-1} \cdot (\mathbf{S}_\mathrm{c} \cdot (1-\gamma) + \mathbf{1} \cdot \gamma). \end{aligned} \tag{7}$$

In (7), $\mathbf{W}_m^{k-1}$ 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 $(0 \le \gamma \le 1)$ 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.

4.4 SS-ConvLSTM

According to the finding in Sect. 3.2.3, we develop an SS-ConvLSTM network in DeepVS2.0 for learning to predict the dynamic saliency of a video clip. At frame t, taking the OM-CNN features $\mathbf{F}$ as the input (denoted as $\mathbf{F}^t$), 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 frame $t-1$. 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 $\mathbf{S}_l^t$ at frame t. The architecture of SS-ConvLSTM is shown in Fig. 10.

Particularly, a CB dropout is developed in SS-ConvLSTM to improve 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 (Kruthiventi et al. 2017; Xu et al. 2017). Specifically, the CB dropout is inspired by the Bayesian dropout (Gal and Ghahramani 2016). 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:

$$\begin{aligned} &\mathbf{Z}(p_b) = \mathrm{Bino}(L,\, p_b \cdot \mathbf{S}_{\mathrm{CB}})/(L \cdot \mathrm{Mean}(\mathbf{S}_{\mathrm{CB}})), \\ &\text{where} \quad \mathbf{S}_{\mathrm{CB}}(i,j) = 1 - \frac{\sqrt{(i - W/2)^2 + (j - H/2)^2}}{\sqrt{(W/2)^2 + (H/2)^2}}. \end{aligned} \tag{8}$$

$\mathrm{Bino}(L, \mathbf{P})$ is a randomly generated mask, in which each pixel (i, j) is subject to an L-trial Binomial distribution according to probability $\mathbf{P}(i, j)$. Here, the probability matrix $\mathbf{P}$ is modeled by CB map $\mathbf{S}_{\mathrm{CB}}$, 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 Xingjian et al. (2015), 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

$$\begin{aligned} \mathbf{I}_1^t &= \sigma((\mathbf{H}_1^{t-1} \circ \mathbf{Z}_i^h) * \mathbf{W}_i^h + (\mathbf{F}^t \circ \mathbf{Z}_i^f) * \mathbf{W}_i^f + \mathbf{B}_i), \\ \mathbf{A}_1^t &= \sigma((\mathbf{H}_1^{t-1} \circ \mathbf{Z}_a^h) * \mathbf{W}_a^h + (\mathbf{F}^t \circ \mathbf{Z}_a^f) * \mathbf{W}_a^f + \mathbf{B}_a), \\ \mathbf{O}_1^t &= \sigma((\mathbf{H}_1^{t-1} \circ \mathbf{Z}_o^h) * \mathbf{W}_o^h + (\mathbf{F}^t \circ \mathbf{Z}_o^f) * \mathbf{W}_o^f + \mathbf{B}_o), \\ \mathbf{G}_1^t &= \tanh((\mathbf{H}_1^{t-1} \circ \mathbf{Z}_g^h) * \mathbf{W}_g^h + (\mathbf{F}^t \circ \mathbf{Z}_g^f) * \mathbf{W}_g^f + \mathbf{B}_g), \\ \mathbf{M}_1^t &= \mathbf{A}_1^t \circ \mathbf{M}_1^{t-1} + \mathbf{I}_1^t \circ \mathbf{G}_1^t, \quad \mathbf{H}_1^t = \mathbf{O}_1^t \circ \tanh(\mathbf{M}_1^t), \end{aligned} \tag{9}$$

where σ and tanh are the activation functions of sigmoid and hyperbolic tangent, respectively. In (9), $\{\mathbf{W}_i^h, \mathbf{W}_a^h, \mathbf{W}_o^h, \mathbf{W}_g^h, \mathbf{W}_i^f, \mathbf{W}_a^f, \mathbf{W}_o^f, \mathbf{W}_g^f\}$ and $\{\mathbf{B}_i, \mathbf{B}_a, \mathbf{B}_o, \mathbf{B}_g\}$ denote the kernel parameters of weight and bias at the corresponding *convolutional layers*; $\mathbf{I}_1^t$, $\mathbf{A}_1^t$ 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 $28 \times 28 \times 128$ in SS-ConvLSTM. 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 (8) 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 $\mathbf{S}_l^t$ 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 (Huang et al. 2015) has proven that the KL divergence is more effective than other metrics

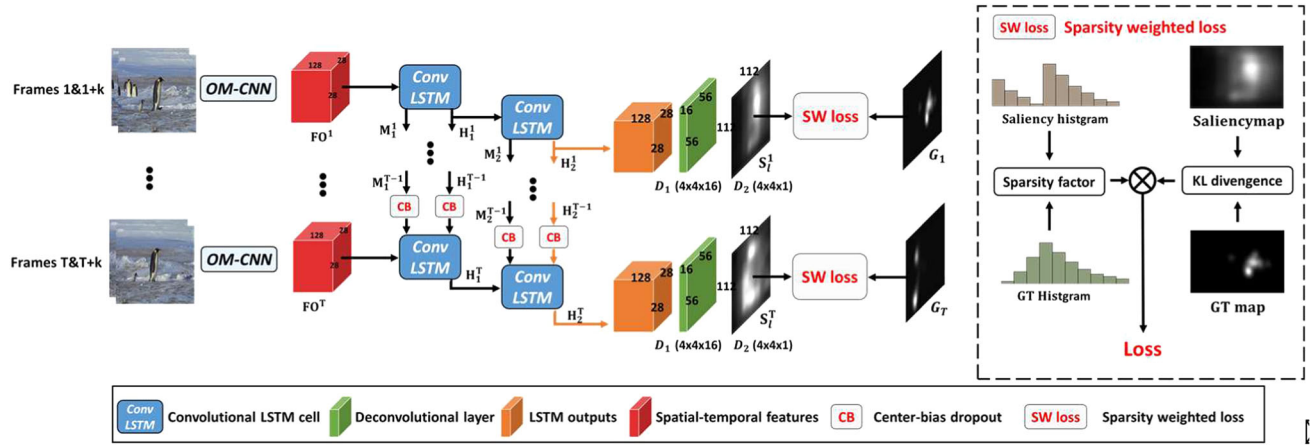


Fig. 10 Architecture of our SS-ConvLSTM in DeepVS2.0 for predicting saliency transition across frames, following the OM-CNN. Note that the training process is not annotated in the figure

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:

$$D_{\mathrm{KL}}(\mathbf{G}, \mathbf{S}_f) = (1/W \times H) \sum_{i=1}^{W} \sum_{j=1}^{H} G_{ij} \log(G_{ij}/S_f^{ij}), \tag{10}$$

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 (10), a smaller KL divergence indicates higher accuracy in saliency prediction.

Loss function for OM-CNN Different from other DNN based saliency prediction methods, an auxiliary function is introduced to train OM-CNN, considering the KL divergence between the cross-net mask $\mathbf{S}_c$ of OM-CNN and the ground-truth $\mathbf{G}$. 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:

$$L_{\mathrm{OM-CNN}} = \frac{1}{1+\lambda} D_{\mathrm{KL}}(\mathbf{G}, \mathbf{S}_f) + \frac{\lambda}{1+\lambda} D_{\mathrm{KL}}(\mathbf{G}, \mathbf{S}_c). \tag{11}$$

In (11), λ is a hyper-parameter for controlling the weights of two KL divergences. Note that in addition to the pre-trained parameters from YOLO and FlowNet, the remaining parameters of OM-CNN, including those in the hierarchical FN and inference modules, are initialized by the Xavier initializer (Glorot and Bengio 2010). Then, all parameters in OM-CNN are updated during the training process.

Loss function for SS-ConvLSTM We propose a sparsity-weighted loss function, which leverages the sparsity prior of saliency maps, for training SS-ConvLSTM. Specifically, the training videos are cut into clips with the same length T, in order to train SS-ConvLSTM. 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. According to Jiang et al. (2015), attention is consistent across different subjects for both videos and images, such that the saliency maps are sparse with a certain histogram, seen as the sparsity prior. Therefore, a sparsity factor f_s is proposed in training SS-ConvLSTM, to make our method generate saliency maps with similar histogram as ground-truth. Sparsity factor f_s can be computed based on the Jensen-Shannon divergence (Manning and Schütze 1999) between the normalized M-bin histogram of the saliency map ($\mathbf{Hist}_s$) and its corresponding ground-truth histogram ($\mathbf{Hist}_g$). Then, the loss function of SS-ConvLSTM is defined as the averaged KL divergence multiplied by the sparsity factor over T frames:

$$\begin{aligned} L_{\mathrm{SS-ConvLSTM}} &= \frac{1}{T} \sum_{i=1}^{T} \eta \cdot f_s \cdot D_{\mathrm{KL}}(\mathbf{S}_l^i, \mathbf{G}_i), \\ \text{where} \quad f_s &= \sum_i \mathbf{Hist}_g(i) \log \frac{2\mathbf{Hist}_g(i)}{\mathbf{Hist}_g(i) + \mathbf{Hist}_s(i)} \\ &\quad + \sum_i \mathbf{Hist}_s(i) \log \frac{2\mathbf{Hist}_s(i)}{\mathbf{Hist}_g(i) + \mathbf{Hist}_s(i)}. \end{aligned} \tag{12}$$

In (12), $\{\mathbf{S}_l^i\}_{i=1}^T$ are the final saliency maps of T frames generated by SS-ConvLSTM, and $\{\mathbf{G}_i\}_{i=1}^T$ are their ground-truth fixation maps. Besides, η is the normalization parameter for sparsity factor, while $\mathbf{Hist}_g(i)$ and $\mathbf{Hist}_s(i)$ are the i-th item in $\mathbf{Hist}_g$ and $\mathbf{Hist}_s$. The details about imposing the sparsity-weighted loss on SS-ConvLSTM are also shown in Fig. 10. For each LSTM cell, the kernel parameters are initialized by the Xavier initializer, while the memory cells and hidden states are initialized by zeros.

Table 2 The values of hyper-parameters in DeepVS2.0

OM-CNN	Objectness mask parameter γ in (7)	0.4
	KL divergences weight λ in (11)	0.75
	Stride k between input frames	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 and 0.75
	Times of Monte Carlo integration L	100
	Video length T	16
	Histogram bins M	50
	Sparsity factor parameter η in (12)	1
	Initial learning rate	1×10^{-4}
	Training epochs (iterations)	$15(\sim 2 \times 10^5)$
	Weight decay	5×10^{-6}

5 Experimental Results

5.1 Settings

In our experiment, the 538 videos of our LEDOV database are randomly divided into training (456 videos), validation (41 videos) and test (41 videos) sets. Specifically, to learn SS-ConvLSTM in our DeepVS2.0 method, 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 in DeepVS2.0, 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 (Kingma and Ba 2015). Here, the hyper-parameters of OM-CNN as well as 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 2. Given the trained models of OM-CNN and SS-ConvLSTM, all 41 test videos in LEDOV database are used to evaluate the performance of our method, in comparison with 10 other state-of-the-art methods. All experiments are conducted on a computer with an Intel(R) Core(TM) i7-4770 CPU@3.4 GHz, 16 GB of RAM and a single Nvidia GeForce GTX 1080 GPU.

5.2 Evaluation on Our LEDOV Database

In this section, we compare the video saliency prediction accuracy of our DeepVS2.0 method and other state-of-the-art methods, including GBVS (Harel et al. 2006), PQFT (Guo and Zhang 2010), Rudoy (Rudoy et al. 2013), OBDL (Hossein Khatoonabadi et al. 2015), SALICON (Huang et al. 2015), Xu et al. (2017), BMS (Zhang and Sclaroff 2016), SalGAN (Pan et al. 2017), AWSD (Leboran et al. 2017), SAM (Cornia et al. 2018) and Wang (Wang et al. 2018). Among these methods, Harel et al. (2006), Guo and Zhang (2010), Rudoy et al. (2013), Hossein Khatoonabadi et al. (2015), Xu et al. (2017), Leboran et al. (2017) and Wang et al. (2018) are 7 traditional saliency prediction methods for videos. Besides, Huang et al. (2015), Pan et al. (2017), Cornia et al. (2018) and Wang et al. (2018) are 4 state-of-the-art DNN-based methods. Table 3 tabulates some attributes and the averaged running time of the above methods, where all methods are tested on the same computer embedded with a GPU. We can see from this table that our method can achieve real-time saliency prediction with around 33 fps (0.03 s per frame), which is the second fastest method.

In our experiments, we apply 7 metrics to measure the performance of saliency prediction: AUC, NSS, CC, KL divergence, similarity metric (SIM), information gain (IG) and earth mover's distance (EMD). It worth noting that IG is calculated by measuring the mutual information between the saliency map and a fixed center-bias baseline. Besides, in order to reduce the computational cost, the saliency map and ground-truth fixation map were downsampled to 14 by 14, when calculating EMD. Among these 7 metrics, the larger value of AUC, NSS, CC, SIM or IG indicates more accurate prediction of saliency, while a smaller KL or EMD means better saliency prediction. Table 4 tabulates the results of AUC, NSS, CC, KL, SIM, IG and EMD for our and 11 other methods, which are averaged over the 41 test videos of LEDOV. As shown in this table, compared to other methods, our DeepVS2.0 method performs best in terms of most metrics. More specifically, our method achieves at least 0.25 KL reduction and 0.24 IG improvement over other meth-

Table 3 Attributes and running time of our and 11 other methods

Method	Year	Video method	DNN based	Implementation	Input size (W × H)	Running time (per frame) (s)	Parameter
GBVS	2006	✓		Matlab	Full size	2.63	–
PQFT	2010	✓		Matlab	1/8 Full size	1.15	–
Rudoy	2013	✓		Matlab	Height = 144	144	–
OBDL	2015	✓		Matlab	Height = 288	1.27	–
SALICON	2015		✓	Python+Caffe*	800 × 600	0.41s	117M
Xu	2016	✓		C++	full size	0.14	–
BMS	2016			Matlab+C++	400 × 400	0.42	–
SalGAN	2017		✓	Python+Theano*	256 × 192	0.14	130M
AWSD	2017	✓		Matlab	Full size	9.62	–
SAM	2018		✓	Python+Theano*	640 × 480	0.07	151M
Wang	2018	✓	✓	Python+Tensorflow*	224 × 224	0.02	96M
Ours	2019	✓	✓	Python+Tensorflow*	448 × 448	0.03	84M

*DNN-based methods run by GPU

Table 4 Mean of saliency prediction accuracy for our and 11 other methods over all test videos in our database

	GBVS	PQFT	Rudoy	OBDL	SALICON*	Xu	BMS	SalGAN*	AWSD	SAM*	Wang°	Ours
AUC	0.84	0.70	0.80	0.80	0.89	0.83	0.76	0.87	0.80	0.88	0.89	**0.90**
NSS	1.54	0.69	1.45	1.54	2.43	1.47	0.98	2.39	1.36	2.94	2.87	**3.02**
CC	0.32	0.14	0.32	0.32	0.43	0.38	0.21	0.45	0.29	0.57	0.56	**0.60**
KL	1.82	2.46	2.42	2.05	1.57	1.65	2.23	1.62	2.02	1.47	1.45	**1.20**
SIM	0.26	0.19	0.23	0.26	0.32	0.22	0.21	0.28	0.22	**0.44**	0.44	0.42
IG	0.57	−0.14	0.28	0.15	0.94	0.29	0.22	0.85	0.26	1.22	1.21	**1.46**
EMD	3.74	5.80	3.09	4.69	3.95	4.55	5.28	4.34	5.19	2.29	2.11	**2.06**

*DNN-based methods have been fine-tuned by our database with their default settings
°We cannot fine-tune this model since it does not release the training code
The best result of each metric is listed in bold

ods. Meanwhile, among the compared methods, Wang et al. (2018) and SAM (Cornia et al. 2018) are the best video and image saliency prediction methods, respectively, which are both based on DNN. This verifies the effectiveness of saliency related features automatically learned by DNN. Note that our method is still significantly superior to Wang et al. (2018). The main reasons for the superior performance of our method 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) SS-ConvLSTM is leveraged to model saliency transition across video frames. Sect. 5.4 analyzes the above three reasons in more detail.

Next, we compare the subjective results in video saliency prediction. Figure 11 demonstrates the saliency maps of 8 randomly selected videos in the test set, predicted by our method and 11 other methods. In this figure, one frame is selected for each video. As shown in Fig. 11, 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 compared methods fail to accurately predict the regions that attract human attention. In addition, Fig. 12 shows the saliency maps of some frames selected from one test video. As shown in this figure, our method is able to model human fixation with a smooth transition, better than all other methods. In summary, our method is superior to other state-of-the-art methods in both objective and subjective results, as tested on our LEDOV database.

5.3 Evaluation on Other Databases

To evaluate the generalization capability of our method, we further evaluate the performance of our DeepVS2.0 method and 11 other methods on two widely used databases, SFU (Hadizadeh et al. 2012) and DIEM (Mital et al. 2011). In the experiments, the models of OM-CNN and SS-ConvLSTM, learned from the training set of our LEDOV database, are directly used to predict the saliency of test videos from the DIEM and SFU databases. Table 5 presents the averaged results of AUC, NSS, CC, KL, SIM, IG and EMD for our

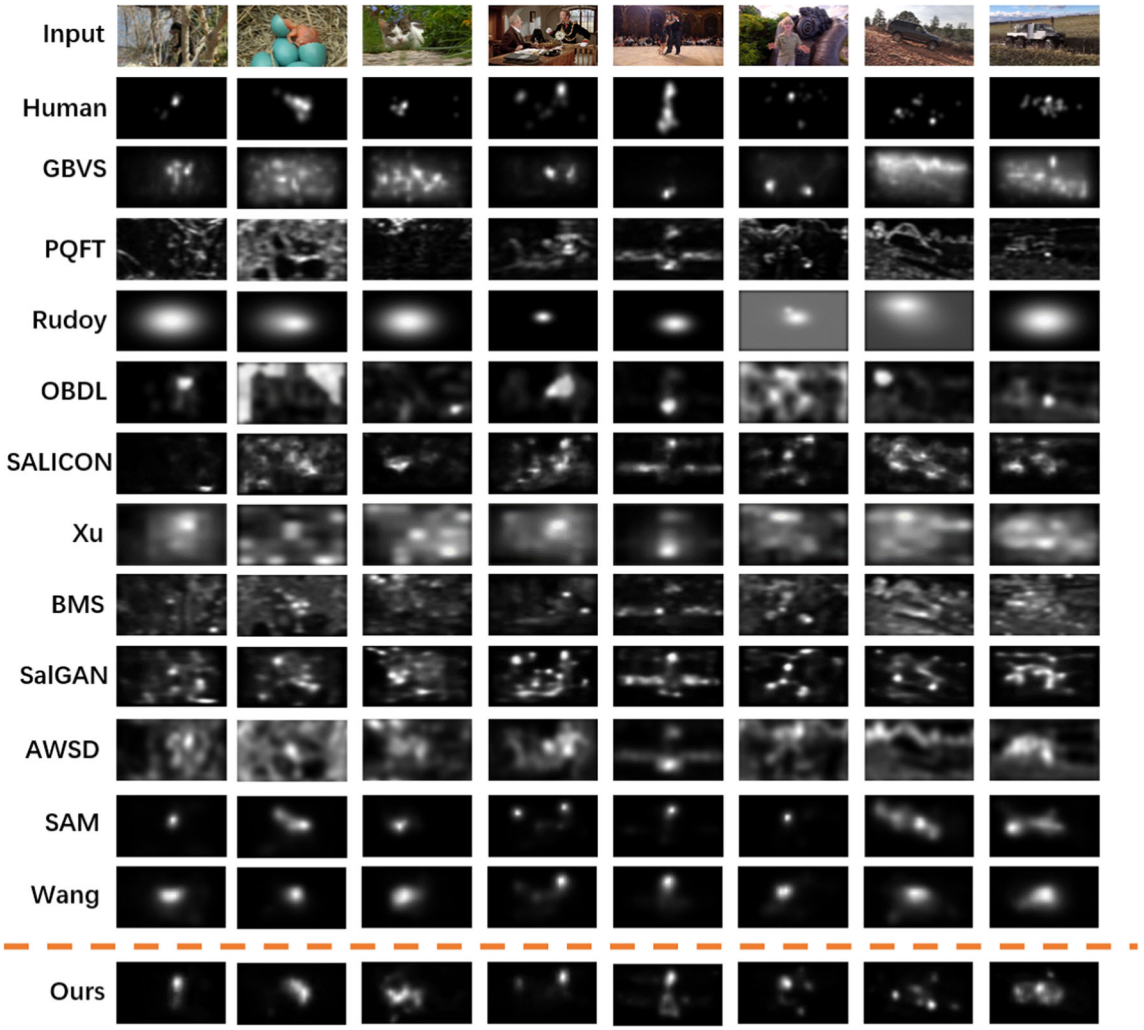


Fig. 11 Saliency maps of 8 videos randomly selected from the test set of our LEDOV database. The maps were yielded by our and 11 other methods as well the ground-truth human fixations. Note that the results of only one frame are shown for each selected video

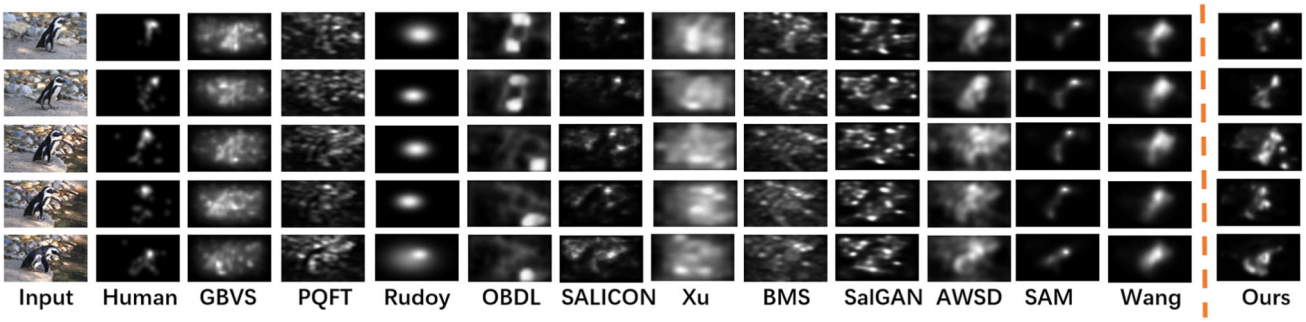


Fig. 12 Saliency maps of several frames randomly selected from a single test video in LEDOV. The maps were yielded by our method and the 11 other methods as well the ground-truth human fixations

and 11 other methods over SFU and DIEM. As shown in this table, our method again outperforms the compared methods, especially over the DIEM database. For instance, there are at least 0.05, 0.52, 0.11, 0.19, 0.01 and 0.09 improvement in AUC, NSS, CC, KL, IG and EMD, respectively. Similar improvement can be found over the SFU database. Such improvement is comparable to that over our LEDOV database. This demonstrates the generalization capability of our method in video saliency prediction.

Table 5 Mean values for saliency prediction accuracy of our and other methods over SFU and DIEM databases

	GBVS	PQFT	Rudoy	OBDL	SALICON*	Xu	BMS	SalGAN*	AWSD	SAM*	Wang°	Ours
AUC	0.76	0.61	0.73	0.74	0.78	0.80	0.66	0.79	0.76	0.79	0.81	**0.82**
NSS	0.91	0.31	0.83	1.03	1.24	1.24	0.50	1.25	1.26	1.22	1.31	**1.53**
CC	0.44	0.12	0.34	0.42	0.58	0.43	0.25	0.51	0.28	0.53	0.54	**0.59**
KL	0.61	0.98	0.93	0.80	1.12	1.35	0.83	0.70	1.82	0.88	0.86	**0.60**
SIM	0.54	0.44	0.34	0.48	0.44	0.55	0.49	0.56	0.49	0.53	0.53	**0.59**
IG	0.44	−0.17	0.02	0.06	−1.29	0.43	0.17	**0.48**	0.23	0.21	0.22	0.47
EMD	2.33	3.08	2.37	2.65	2.65	2.35	2.68	2.20	2.71	2.25	2.14	**2.01**
	DIEM											
	GBVS	PQFT	Rudoy	OBDL	SALICON*	Xu	BMS	SalGAN*	AWSD	SAM*	Wang°	Ours
AUC	0.81	0.71	0.80	0.75	0.79	0.80	0.77	0.81	0.76	0.78	0.81	**0.86**
NSS	1.21	0.86	1.40	1.26	1.68	1.34	1.20	1.60	1.26	1.78	1.50	**2.30**
CC	0.30	0.19	0.38	0.29	0.36	0.35	0.28	0.35	0.28	0.38	0.40	**0.51**
KL	1.64	1.73	2.33	2.77	1.66	1.67	1.96	1.64	1.82	1.55	1.44	**1.25**
SIM	0.30	0.24	0.28	0.27	0.30	0.27	0.27	0.28	0.28	0.40	**0.44**	0.40
IG	0.27	−0.21	−1.89	−2.81	0.39	0.09	−0.10	0.36	0.14	0.48	0.94	**0.95**
EMD	2.92	3.58	2.56	3.57	3.12	3.26	3.39	3.12	3.40	2.77	2.49	**2.40**

*DNN-based methods have been fine-tuned by our database with their default settings
°We cannot fine-tune this model since it does not release the training code
The best result of each metric is listed in bold

Table 6 Averaged CC between saliency maps and object region/object motion maps

	OM-CNN	Objectness subnet	Motion subnet
Object region	0.42	0.44	0.36
Object motion map	0.35	0.30	0.35

5.4 Performance Analysis of DeepVS2.0

Supervision evaluation of subnets Since OM-CNN is composed of the objectness and motion subnets, we evaluate the supervision of objectness and object motion in our saliency prediction method. Specifically, the objectness subnet, motion subnet and OM-CNN are trained independently with the same settings introduced above. For each video in the validation set, 3 sets of saliency maps are obtained upon the trained models of objectness subnet, motion subnet and OM-CNN. Meanwhile, the maps of object regions and object motion are also generated based on YOLO and FlowNet. Then, the CC values between saliency maps and the maps of object regions/object motion are shown in Table 6. We can see from Table 6 that the saliency maps generated from OM-CNN and the objectness subnet are much more correlated to object regions than those from the motion subnet. This indicates that the objectness subnet contributes to learning objectness cues for saliency prediction. Similarly, the motion subnet is verified to highlight the object motion information in video saliency prediction.

Evaluation on the cross-net mask For evaluating the effectiveness of the cross-net mask, we visualize the intermediate cross-net masks at different iterations in training OM-CNN. Figure 13 shows the cross-net masks, ground-truth fixation maps and output saliency maps of some video frames. We can see from this figure that, at the beginning of the training process (500 iterations), the cross-net masks are coarse due to the random initialization of the inference module. Then, along with the increased training iterations, the cross-net masks become sparse and center on the object regions. Thus, the cross-net mask is effective in adding the object information to the motion subnet, for saliency prediction.

Performance analysis of components We further analyze the contribution of each component in our method through ablation experiments. The KL and CC results of our ablation experiments are shown in Fig. 14, in which each model is trained independently. Note that the improvement by the sparsity weighted loss is not significant, since it is used for adjusting the sparsity of saliency maps, not for improving the accuracy of saliency prediction. In Fig. 14, DeepVS (Jiang et al. 2018) and the method (Wang et al. 2018) with second best performance in Table 4, are used as the baselines. As shown in this figure, there are 0.070 reduction in KL and 0.021 improvement in CC for DeepVS2.0 against DeepVS. We can see from Fig. 14 that OM-CNN performs

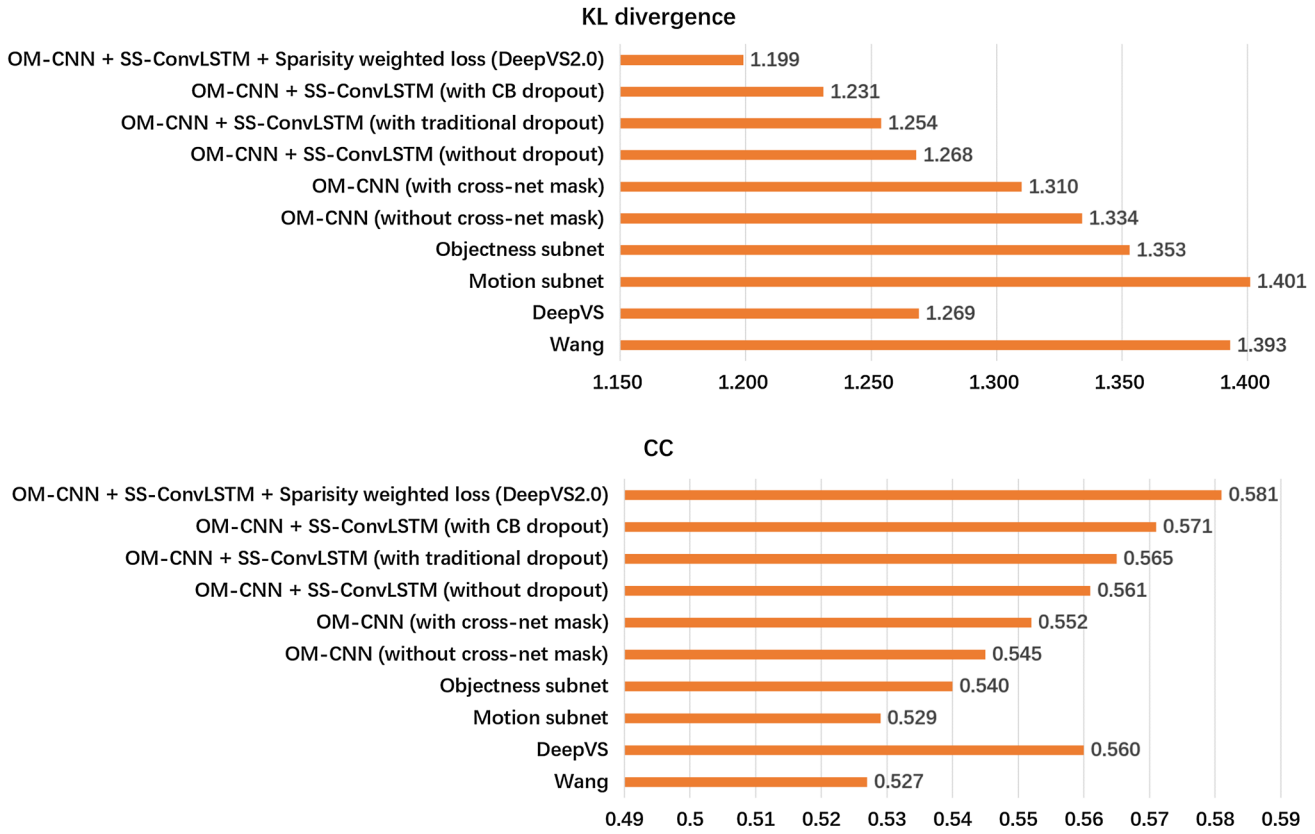


Fig. 14 Ablation results of our DeepVS2.0 method, compared with DeepVS (our conference paper) and Wang. Note that the smaller KL or larger CC indicates higher accuracy in saliency prediction

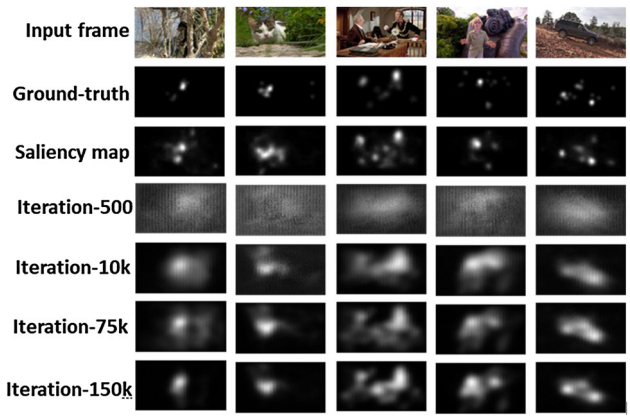


Fig. 13 Cross-net masks at different iterations of 5 randomly selected input frames, as well as their corresponding ground-truths and output saliency maps of our method

better than the objectness subnet with 0.043 KL reduction and 0.012 CC improvement. Meanwhile, OM-CNN outperforms the motion subnet by 0.091 reduction in KL and 0.023 improvement in CC. These results indicate the effectiveness of integrating the subnets of objectness and motion. Moreover, Fig. 14 shows that the final model with OM-CNN and SS-ConvLSTM improves the performance by 0.111 in KL and 0.029 in CC, against the single OM-CNN. Hence, we can conclude that SS-ConvLSTM can further improve the performance of OM-CNN. In addition, the cross-net mask, CB dropout and sparsity-weighted loss can reduce KL divergence by 0.024, 0.037 and 0.032, respectively. Similar improvement can be found in terms of the other 6 metrics. Note that the ablation results of all metrics are provided in our supplementary material. This verifies the effectiveness of the proposed components in our method.

Performance analysis of backbone structures As introduced above, we update the backbone structures of the motion and objectness subnets in DeepVS2.0, taking the advantage of more recent YOLO (Redmon and Farhadi 2018) and FlowNet (Ilg et al. 2017) models. Here, we evaluate the performance of our method with different backbone structures from YOLO and FlowNet. Note that in order to avoid over-fitting and reduce computational cost, we do not apply the whole structure of YOLO or FlowNet. Specifically, we apply the basic structures of YOLOv1 (Redmon et al. 2016) (*i.e.*, Fast YOLO), YOLOv2 (Redmon and Farhadi 2017) (*i.e.*, Darknet-19) and YOLOv3 (Redmon and Farhadi 2018) (*i.e.*, Darknet-53) as the objectness subnet. Meanwhile, we implement the motion subnet by FlowNet-Simple in FlowNet1 (Dosovitskiy et al. 2015), FlowNet2-SD,

Table 7 Performance of our method with different backbone structures

	CC	KL
YOLOv1 + FlowNet1 (DeepVS[a])	0.567	1.252
YOLOv2 + FlowNet1	0.562	1.271
YOLOv3 + FlowNet1	0.574	1.223
YOLOv3 + FlowNet2-SD	0.572	1.230
YOLOv3 + FlowNet2-C	0.566	1.256
YOLOv3 + FlowNet2-S (DeepVS2.0)	0.581	1.199

[a]Sparsity weighted loss is added for fair comparison

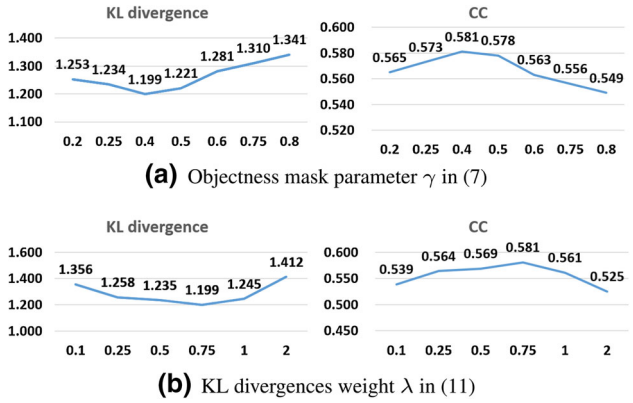


Fig. 15 KL divergences of our models with different values of γ and λ

FlowNet2-S and FlowNet2-C in FlowNet2 (Ilg et al. 2017). According to the results in Table 7, the combination of YOLOv3 and FlowNet2-S achieves the best performance. Compared with DeepVS, the new backbone structures can bring 0.014 CC improvement and 0.053 KL reduction, respectively. In addition to CC and KL, similar results can be found in terms of other 5 metrics, which are provided in our supplementary material.

Performance analysis of OM-CNN Here, we analyze the impacts of 2 hyper-parameters of OM-CNN for saliency prediction: i.e., cross-net mask weight γ and auxiliary KL weight λ. We can see from (7) that γ is an adjustable parameter to control the degree of the cross-net mask, while λ in (11) is used to balance the KL divergences of main and auxiliary loss function in training the OM-CNN model. Figure 15 shows the KL divergences of our OM-CNN model with various values of γ and λ. Therefore, in our method, cross-net mask weight γ and auxiliary KL weight λ are set to 0.4 and 0.75, respectively, to achieve the appropriate performance of saliency prediction.

Performance analysis of SS-ConvLSTM We evaluate the performance of proposed CB dropout in 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 we test the trained models over the validation set. The averaged KL divergence results are shown in Fig. 16a. We can see that the CB dropout can reduce KL divergence by 0.032 and improve CC by 0.014, when both p_h and p_f are 0.75, compared to the model without CB dropout (i.e., $p_h = p_f = 1$). Meanwhile, the performance sharply degrades by 0.064 in KL and 0.034 in CC, 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 at a fixed video length ($T = 16$). We further evaluate the saliency prediction performance of the trained SS-ConvLSTM model on variable-length videos. Here, we test the trained SS-ConvLSTM model over the validation set, the videos of which are clipped with their length varying from 8 to 54 frames. Fig. 16b shows the averaged KL divergence and CC for video clips at various lengths. We can see from this figure that the performance of SS-ConvLSTM fluctuates from 8 to 54 frames, with the maximum at $T = 32$. On the other hand, the performance fluctuation of SS-ConvLSTM is tiny, indicating the robustness of our method to varying video length.

6 Conclusion

In this paper, we have proposed a DNN-based method, which predicts video saliency through DeepVS2.0. For training the DNN models of DeepVS2.0, we established the LEDOV database, which has the fixations of 32 subjects on 538 videos. Then, the OM-CNN architecture was proposed in DeepVS2.0 to explore the spatio-temporal features of the objectness and object motion to predict the intra-frame saliency of videos. In DeepVS2.0, SS-ConvLSTM was developed to model the inter-frame saliency of videos. Finally, the experimental results verified that our method significantly outperforms 11 other state-of-the-art methods over both our and 2 other public eye-tracking databases, in terms of AUC, CC, NSS, KL, SIM, IG and EMD metrics. Thus, the prediction accuracy and generalization capability of our DeepVS2.0 method can be validated.

It is interesting to discuss the of inspiration this work may bring to other computer vision tasks. For instance, the cross-net mask can be regarded as a kind of attention mechanism, which has been wildly used in other vision tasks (Hu et al. 2018; Woo et al. 2018; Fu et al. 2019). Different from the existing attention mechanisms, the cross-net mask is extracted from the objectness subnet, indicating the region of interest (ROI) related to objects. Besides, the CB dropout can be utilized in other computer vision tasks for considering the center-bias prior, such as image classification and object detection. Similarly, the sparsity weighted loss can be used in some vision tasks that need to generate images with certain sparsity, such as optical flow.

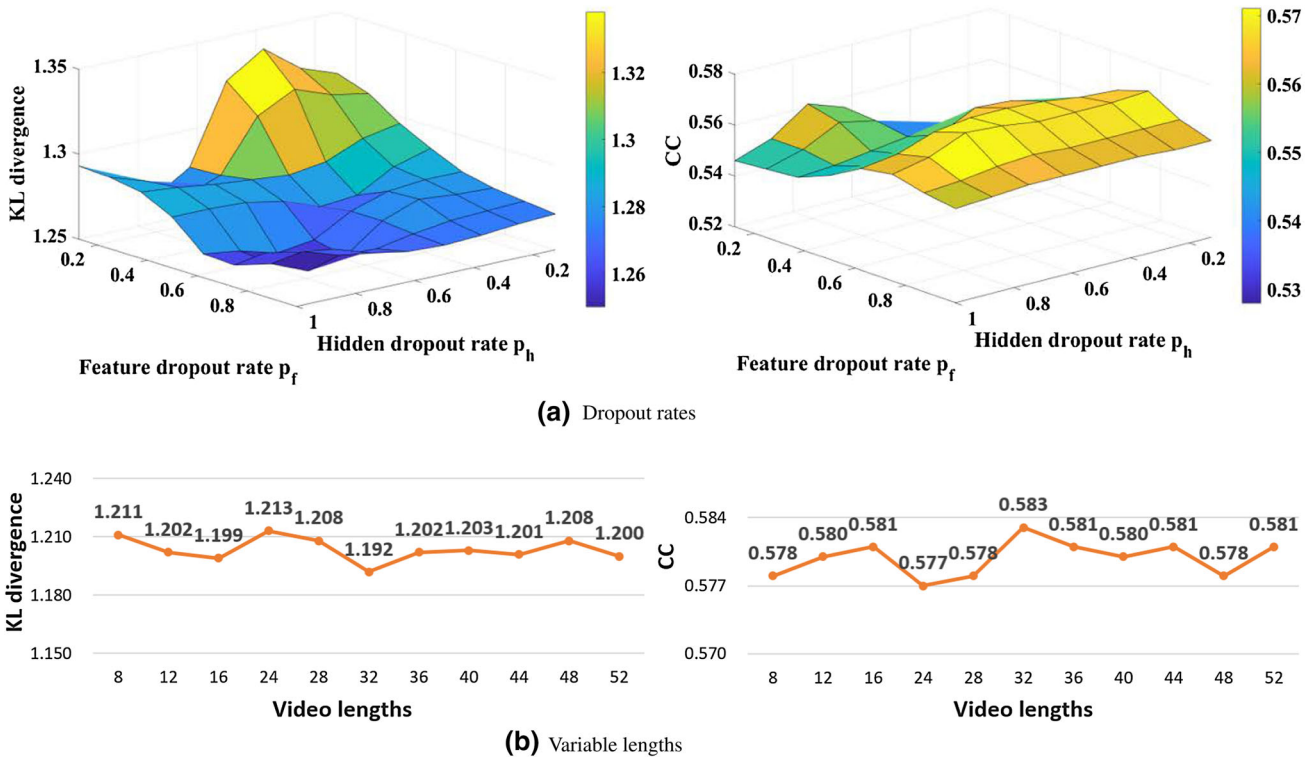


Fig. 16 **a** KL divergence and CC of our models with different dropout rates. **b** KL divergence and CC over validation videos with variable lengths

References

Alers, H., Redi J. A., & Heynderickx, I. (2012). Examining the effect of task on viewing behavior in videos using saliency maps. In *IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics* (pp. 82910X–82910X).

Bak, C., Kocak, A., Erdem, E., & Erdem, A. (2017). Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Transactions on Multimedia*, *20*, 1688–1698.

Bazzani, L., Larochelle, H., & Torresani, L. (2017). Recurrent mixture density network for spatiotemporal visual attention.

Borji, A., & Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 185–207.

Boulos, F., Chen, W., Parrein, B., & Le Callet, P. (2009). Region-of-interest intra prediction for h. 264/AVC error resilience. In *ICIP, IEEE* (pp. 3109–3112).

Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., & Durand, F. (2018). What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(3), 740–757.

Carmi, R., & Itti, L. (2006). Visual causes versus correlates of attentional selection in dynamic scenes. *Vision Research*, *46*(26), 4333–4345.

Chaabouni, S., Benois-Pineau, J., & Amar, C.B. (2016) Transfer learning with deep networks for saliency prediction in natural video. In: *ICIP, IEEE*, pp 1604–1608.

Cheng, M. M., Mitra, N. J., Huang, X., Torr, P. H., & Hu, S. M. (2015). Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 569–582.

Cornia, M., Baraldi, L., Serra, G., & Cucchiara, R. (2018). SAM: Pushing the limits of saliency prediction models. In *Proceedings of the IEEE/CVF international conference on computer vision and pattern recognition workshops*.

Coutrot, A., & Guyader, N. (2013). Toward the introduction of auditory information in dynamic visual attention models. In *2013 14th international workshop on image analysis for multimedia interactive services (WIAMIS), IEEE* (pp 1–4).

Coutrot, A., & Guyader, N. (2015). An efficient audiovisual saliency model to predict eye positions when looking at conversations. In *23rd European signal processing conference (EUSIPCO)*, IEEE (pp. 1531–1535).

Dorr, M., Martinetz, T., Gegenfurtner, K. R., & Barth, E. (2010). Variability of eye movements when viewing dynamic natural scenes. *Journal of vision*, *10*(10), 28–28.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., & Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. In *ICCV* (pp. 2758–2766).

Du, Y., Wang, W., & Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR* (pp. 1110–1118).

Fang, Y., Lin, W., Chen, Z., Tsai, C. M., & Lin, C. W. (2014a). A video saliency detection model in compressed domain. *IEEE Transactions on Circuits and Systems for Video Technology*, *24*(1), 27–38.

Fang, Y., Wang, Z., Lin, W., & Fang, Z. (2014b). Video saliency incorporating spatiotemporal cues and uncertainty weighting. *IEEE Transactions on Image Processing*, *23*(9), 3910–3921.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3146–3154).

Gal, Y., & Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In: *NIPS* (pp. 1019–1027).

Gitman, Y., Erofeev, M., Vatolin, D., & Andrey, B. (2014). Semi-automatic visual-attention modeling and its application to video compression. In *ICIP, IEEE* (pp. 1105–1109).

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).

Goferman, S., Zelnik-Manor, L., & Tal, A. (2012). Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(10), 1915–1926.

Guo, C., & Zhang, L. (2010). A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Transactions on Image Processing*, *19*(1), 185–198.

Hadizadeh, H., Enriquez, M. J., & Bajic, I. V. (2012). Eye-tracking database for a set of standard video sequences. *IEEE Transactions on Image Processing*, *21*(2), 898–903.

Harel, J., Koch, C., & Perona, P. (2006). Graph-based visual saliency. In *NIPS* (pp. 545–552).

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Van de Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures*. Oxford: OUP.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, X., Shen, C., Boix, X., & Zhao, Q. (2015). Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *ICCV* (pp. 262–270).

Huang, C. R., Chang, Y. J., Yang, Z. X., & Lin, Y. Y. (2014). Video saliency map detection by dominant camera motion removal. *IEEE Transactions on Circuits and Systems for Video Technology*, *24*(8), 1336–1349.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2462–2470).

Itti, L. (2004). Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, *13*(10), 1304–1318.

Itti, L., & Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision Research*, *49*(10), 1295–1306.

Itti, L., Dhavale, N., & Pighin, F. (2004). Realistic avatar eye and head animation using a neurobiological model of visual attention. *Optical Science and Technology*, *64*, 64–78.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1254–1259.

Jiang, L., Xu, M., Liu, T., Qiao, M. & Wang, Z. (2018). Deepvs: A deep learning based video saliency prediction approach. In *ECCV*, Berlin: Springer.

Jiang, L., Xu, M., Ye, Z., & Wang, Z. (2015). Image saliency detection with sparse representation of learnt texture atoms. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 54–62).

Judd, T., Ehinger, K., Durand, F., & Torralba, A. (2009). Learning to predict where humans look. In *ICCV* (pp. 2106–2113).

Khatoonabadi, S. H., Vasconcelos, N., Bajic, I. V., & Shan, Y. (2015). How many bits does it take for a stimulus to be salient? In *CVPR* (pp. 5501–5510).

Kim, N. W., Bylinskii, Z., Borkin, M. A., Gajos, K. Z., Oliva, A., Durand, F., et al. (2017). Bubbleview: an interface for crowdsourcing image importance maps and tracking visual attention. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *24*(5), 1–40.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization.

Kruthiventi, S. S., Ayush, K., & Babu, R. V. (2017). Deepfix: A fully convolutional neural network for predicting human eye fixations. *IEEE Transactions on Image Processing.*, *26*, 4446–4456.

Kümmerer, M., Theis, L., & Bethge, M. (2014) Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. arXiv preprint arXiv:1411.1045.

Le, T. N., & Sugimoto, A. (2017). Video salient object detection using spatiotemporal deep features. arXiv preprint arXiv:1708.01447.

Leboran, V., Garcia-Diaz, A., Fdez-Vidal, X. R., & Pardo, X. M. (2017). Dynamic whitening saliency. *IEEE PAMI*, *39*(5), 893–907.

Lee, S. H., Kim, J. H., Choi, K. P., Sim, J. Y., & Kim, C. S. (2014). Video saliency detection based on spatiotemporal feature learning. In *ICIP* (pp 1120–1124).

Li, Z., Qin, S., & Itti, L. (2011). Visual attention guided bit allocation in video compression. *Image and Vision Computing*, *29*(1), 1–14.

Li, J., Tian, Y., Huang, T., & Gao, W. (2010). Probabilistic multi-task learning for visual saliency estimation in video. *International Journal of Computer Vision*, *90*(2), 150–165.

Liu, Y., Zhang, S., Xu, M., & He, X. (2017). Predicting salient face in multiple-face videos. In: *CVPR*.

Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., et al. (2011). Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(2), 353–367.

Li, J., Xia, C., & Chen, X. (2018). A benchmark dataset and saliency-guided stacked autoencoders for video-based salient object detection. *IEEE Transactions on Image Processing*, *27*(1), 349–364.

Li, X., Zhao, L., Wei, L., Yang, M. H., Wu, F., Zhuang, Y., et al. (2016). Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing*, *25*(8), 3919–3930.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge: MIT Press.

Marat, S., Guironnet, M., & Pellerin, D. (2007). Video summarization using a visual attention model. In *Signal processing conference, IEEE* (pp. 1784–1788).

Marszalek, M., Laptev, I., & Schmid, C. (2009). Actions in context. In *CVPR, IEEE* (pp. 2929–2936).

Mathe, S., & Sminchisescu, C. (2015). Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(7), 1408–1424.

Matin, E. (1974). Saccadic suppression: A review and an analysis. *Psychological Bulletin*, *81*(12), 899–917.

Mauthner, T., Possegger, H., Waltner, G., & Bischof, H. (2015). Encoding based saliency detection for videos and images. In: *CVPR* (pp. 2494–2502).

Mital, P. K., Smith, T. J., Hill, R. L., & Henderson, J. M. (2011). Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, *3*(1), 5–24.

Nguyen, T.V., Xu, M., Gao, G., Kankanhalli, M., Tian, Q., & Yan, S. (2013). Static saliency vs. dynamic saliency: A comparative study. In: *ACMM, ACM* (pp. 987–996).

Olsen, A. (2012). *The Tobii i-vt fixation filter*. Danderyd: Tobii Technology.

Palazzi, A., Solera, F., Calderara, S., Alletto, S., & Cucchiara, R. (2017). Learning where to attend like a human driver. *Intelligent Vehicles Symposium (IV)* (pp. 920–925). IEEE: IEEE.

Pan, J., Canton, C., McGuinness, K., O'Connor, N.E., Torres, J., Sayrol, E., & Xa, G. N. (2017). Salgan: Visual saliency prediction with generative adversarial networks. In *CVPR Workshop*.

Pan, J., Sayrol, E., Giro-i Nieto, X., McGuinness, K., & O'Connor, N. E. (2016). Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp 598–606).

Peters, R.J., & Itti, L. (2007). Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention. In: *CVPR, IEEE* (pp. 1–8).

Rajashekar, U., van der Linde, I., Bovik, A. C., & Cormack, L. K. (2008). Gaffe: A gaze-attentive fixation finding engine. *IEEE Transactions on Image Processing*, *17*(4), 564–573.

Redmon, J., & Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263–7271).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In: *CVPR* (pp. 779–788).

Ren, Z., Gao, S., Chia, L. T., & Rajan, D. (2013). Regularized feature reconstruction for spatio-temporal saliency detection. *IEEE Transactions on Image Processing*, *22*(8), 3120–3132.

Riche, N., Mancas, M., Culibrk, D., Crnojevic, V., Gosselin, B., & Dutoit, T. (2012). Dynamic saliency models and human attention: A comparative study on videos. In *ACCV* (pp. 586–598), Berlin: Springer.

Rodriguez, M. (2010). *Spatio-temporal maximum average correlation height templates in action recognition and video summarization*. Princeton: Citeseer.

Rudoy, D., Goldman, D. B., Shechtman, E., & Zelnik-Manor, L. (2013) Learning video saliency from human gaze using candidate selection. In *CVPR* (pp. 1147–1154).

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *CVPR* (pp. 1–9).

Tobii I TECHNOLOGY (2017). Tobii tx300 eye tracker. Retrieved July, 2018, from http://www.tobiipro.com/product-listing/tobii-pro-tx300/.

Wang, W., Shen, J, Guo, F., Cheng, M. M., & Borji, A. (2018). Revisiting video saliency: A large-scale benchmark and a new model. In *CVPR*.

Wang, L., Wang, L., Lu, H., Zhang, P., & Ruan, X. (2016a) Saliency detection with recurrent fully convolutional networks. In: *ECCV* (pp. 825–841). Berlin: Springer.

Wang, Y., Zhang, Q., & Li, B. (2016b). Efficient unsupervised abnormal crowd activity detection based on a spatiotemporal saliency detector. In *WACV, IEEE* (pp. 1–9).

Wang, W., & Shen, J. (2018). Deep visual attention prediction. *IEEE Transactions on Image Processing*, *27*(5), 2368–2378.

Wang, W., Shen, J., & Shao, L. (2017). Video salient object detection via fully convolutional networks. *IEEE Transactions on Image Processing.*, *27*, 38–49.

Woo, S., Park, J., Lee, J.Y., & Kweon, I.S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3–19).

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

Xu, M., Jiang, L., Sun, X., Ye, Z., & Wang, Z. (2017). Learning to detect video saliency with hevc features. *IEEE Transactions on Image Processing*, *26*(1), 369–385.

Zhang, L., Tong, M. H., & Cottrell, G. W. (2009). Sunday: Saliency using natural statistics for dynamic analysis of scenes. In *Annual cognitive science conference* (pp. 2944–2949).

Zhang, J., & Sclaroff, S. (2016). Exploiting surroundedness for saliency detection: A boolean map approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(5), 889–902.

Zhong, S. H., Liu, Y., Ren, F., Zhang, J., & Ren, T. (2013). Video saliency detection via dynamic consistent spatio-temporal attention modelling. In *AAAI*.

Zhou, F., Kang, S. B., & Cohen, M. F. (2014). Time-mapping using space-time saliency. In *CVPR* (pp. 3358–3365).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.