Multimedia Tools and Applications (2024) 83:32301–32320
https://doi.org/10.1007/s11042-023-16593-2

Fusion hierarchy motion feature for video saliency detection

Fen Xiao[1] · Huiyu Luo[1] · Wenlei Zhang[1] · Zhen Li[1] · Xieping Gao[2]

Received: 11 July 2022 / Revised: 21 June 2023 / Accepted: 21 August 2023 /
Published online: 20 September 2023

Abstract
Saliency detection plays an important role in computer vision and scene understanding, which has attracted increasing attention in recent years. Compared to the widely studied image saliency prediction, there are still many problems to be solved in the area of video saliency. Different from images, effectively describing and utilizing the motion information contained in video data is a critical issue. In this paper, we propose a spatial and motion dual-stream framework for video saliency detection. The coarse motion features extracting from optical flow are fine-tuned with higher level semantic spatial features via a residual cross-connection. A hierarchical fusion structure is proposed to maintain contextual information by integrating spatial and motion features in each level. To model the inter-frame correlation in the video, the convolutional gated recurrent unit (convGRU) is used to retain global consistency of the saliency area between neighbor frames. Experimental results on four widely used datasets demonstrate the effectiveness of the proposed method with other state-of-the-art methods. Our source codes can be acquired at https://github.com/banhuML/MFHF.

Keywords Video saliency detection · Motion feature fine-tuning · Hierarchical fusion · Optical flow

1 Introduction

The human visual system (HVS) can quickly select and focus on relevant areas. This selective mechanism is known as the visual attention mechanism, which has a variety of applications in action recognition [1], video summarization [2], video segmentation [3], image caption [4] and image quality assessment [5]. To simulate visual attention mechanism, two saliency related vision tasks including salient object detection [6–14] and saliency detection have

✉ Xieping Gao
xpgao@hunnu.edu.cn

Fen Xiao
xiaof@xtu.edu.cn

[1] The MOE Key Laboratory of Intelligent Computing and Information Processing, Xiangtan University, Xiangtan, Hunan, China

[2] Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, Hunan, China

been developed during recent years. Differ from salient object detection models aiming to segment salient objects in pixel level, saliency detection predicts human fixation region and can be easily calculated in the video processing [15]. With the fast development of deep learning technology and plenty of image fixation datasets, many image saliency detection models [16–22] have achieved significant successes.

In contrast to image saliency detection, which acquires saliency cues only from a single image, video saliency detection additionally requires consideration of differences between multiple consecutive frames to infer saliency distribution. These differences across the temporal domain are generated by the combined motion of objects and camera. Human attention is more likely to be attracted to moving objects during free-viewing [24]. As shown in Fig. 1(a) and (b), most of eye fixations are located around the falling soccer. Therefore, it is necessary to extract motion information from video sequences to acquire saliency cues.

Optical flow reflects the changes and correlation in the time domain of the pixels between adjacent frames, which has been a prevailing way to describe motion information [25]. As the optical flow estimated by the method RAFT [26] shown in Fig. 1(c), most of the optical flow show well-defined salient objects that provide motion saliency cues, but others are blurred due to slow motion of the object or only partially moving within the object. Cong et al. [27] suggest that different motion states of objects can yield different optical flow estimates even in similar scenarios. The motion features extracted from the optical flow are concentrated around the salient object, as shown in Fig. 1(d). The motion features extracted from the blurred optical flow are more diffuse, which makes it difficult to find the exact location of salient objects. Therefore, the need to obtain more accurate saliency cues from motion features has become an urgent requirement for video saliency detection.

Existing optical flow-based models [28–31] for video saliency detection use two-stream networks to extract motion and spatial information separately, and then simply fuse them by concatenate, etc. These direct integration strategies ignore the fact that the two types of information come from different modalities. Lai et al. [31] enhanced spatial information with motion information through an attention mechanism and achieved good performance,

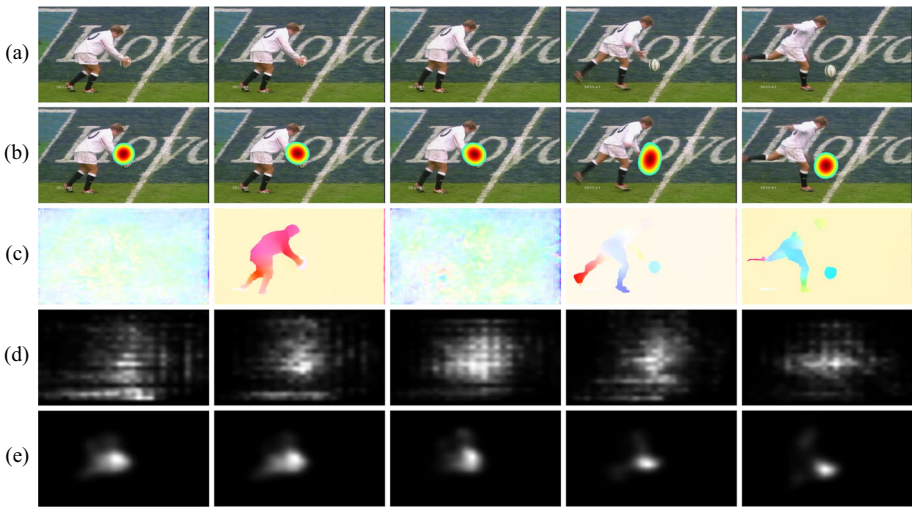

Fig. 1 (a) Video frames selected from UCF Sports [23]. (b) Ground truth (d) Optical flow. (e) Motion feature maps. (f) Fine-tuned motion feature maps

but neglected the lack of accurate motion saliency cues in some motion features, resulting in the inability to efficiently aggregate motion and spatial information for the saliency detection.

To alleviate the above challenges as well as to compensate for the shortcomings of existing methods, we propose a video saliency detection model that combines spatial and motion information in a multiscale manner, consisting of spatial subnet, motion subnet, hierarchical fusion subnet and convGRU subnet. We follow a two-stream network structure consisting of two same CNN models, with the spatial and motion subnets extracting spatial and motion features from optical flow and video frames, respectively. We propose the motion feature fine-tuning module to fine-tune the motion features using multiscale spatial features in the feature extraction process, which can focus motion features on the salient objects. The fine-tuned motion features are shown in Fig. 1(e). For further studying the relationships accross different scale motion features and extracting more semantic information, we design a hierarchical fusion subnet to integrate spatial and motion features in a multi-scale pattern. Considering that the saliency of adjacent frames is correlated, the convGRU [32] subnet generates the final saliency map based on the saliency cues of the current frame together with the saliency results of previous frames.

To sum up, the main contributions of this paper are summarized as follows:

1. We proposed a novel layered network MFHF for video saliency detection, which contains four subnets, spatial, motion, hierarchical fusion and convGRU subnets. The proposed method can extract informative motion features and fuse spatial features to predict video saliency accurately.
2. We developed the motion feature fine-tuning module for extracting new motion features. A series of optical flow have been used as coarse motion features, which were be fine-tuned by incorporating spatial features with cross-connections in the last three layers of spatial subnet.
3. We designed the hierarchical fusion subnet for fully combining spatial and motion features on five different scales, which can retain more multi-scale contextual information.

The rest of the paper is organized as follows. In Section 2, we review some typical related work. Section 3 elaborates the proposed video saliency detection model. Section 4 reports the experimental results and ablation analysis of our model on four publicly available benchmark datasets. Finally, the conclusions are drawn in Section 5.

2 Related work

In this section, we review related studies on saliency detection for images and videos.

2.1 Salient object detection

Video salient object detection, which aims to segment the most obvious objects from the picture, has remained high in the computer vision research community and has a wide range of applications in optimal path planning [33] and robot navigation [34].

Many models [6–14] use optical flow to better segment moving salient objects. Some video salient object detection methods use the motion information of the optical flow to better segment moving salient objects. Li et al. [12] develop a motion guided video salient object detection network, which leverages the motion saliency sub-network to attend and enhance the sub-network for still images. Chen et al. [14] introduce the concept of motion

quality and select video frames with high-quality motions as the new training set, which is used to fine-tune the model. Zhang et al. [10] used color contrast computation and optical flow computation to enhance spatio-temporal correlation, and combined with depth confidence optimization to accomplish stereoscopic video saliency detection.

2.2 Saliency detection

Earlier image saliency detection models typically used a bottom-up framework, also known as a stimuli-driven mechanism [31]. Many of these works are based on the calculation model of HVS, comprehensively considering color features, directional features and gray-scale features [35–40]. In the past years, several emerging deep learning models have achieved groundbreaking progress compared with traditional models. These saliency models based on deep learning mainly profited by extensive labeled training data and more expressive network structure. Vig et al. [16] used the Convolutional Neural Network to obtain feature vectors, then put them into the SVM [41] to generate the image saliency prediction results. SALICON [18] aimed to narrow the semantic gap and predict saliency results based on a pre-trained VGG-16 [42]. Similarly, DeepNet [19] connected more network layers into the VGG-16 and obtained more informative multi-scale features to promote the performance. SAM [43] iteratively enhanced the coarse feature to focus on the most salient region through a convolutional LSTM(convLSTM) [44] and a center priors attention mechanism. DVA [45] utilized the skip-layer network to acquire hierarchical saliency information, and achieved efficient properties for image saliency detection.

For video saliency detection, the spatial-temporal information contained in the video frames is critical. Similar to image saliency detection, traditional video saliency detection methods capture saliency cues based on hand-crafted spatial-temporal features [46, 47], but low-level hand-crafted features couldn't deliver a satisfactory performance for modeling dynamic saliency. Recently, a good deal of models based on deep learning have been proposed that adopt different ways of acquiring temporal information. ACLNet [48] proposed a supervised attentive module to encode static attention and then used it in convLSTM [44] to learn dynamic saliency representation. SALDPC [49] captured motion information through the multi-scale temporal recurrence and can better guide video coding with saliency. Compared with temporal modules like convLSTM used in the above methods, optical flow has better motion sensing abilities [50], which are closely related to the acquisition of motion saliency cues.

Optical flow represents per-pixel motion between two consecutive frames [51], which is sufficient to establish the link between motion and saliency [28] and has become a prevalent way to reflect the motion situation of objects in video saliency detection. Bak et al. [28] developed a two-stream network to extract spatial and motion information from video frames and optical flow, respectively, and integrated them with max fusion or convolutional fusion for video saliency prediction. DeepVS [29] is an another well-known video saliency model, which extracted spatial and motion features via YOLO [52] and FlowNet [53]. Then the extracted two kinds of features were concatenated to generate spatio-temporal features. Finally, the convLSTM was used for learning inter-frame correlation. However, these methods only integrate motion features with spatial features use direct fusion strategies, ignoring the problem of motion feature diffusion caused by blurred optical flow, and making insufficient use of both features.

To address the above problems, we develop a two-stream structure and applied multi-layer spatial features to fine-tune the motion features, and influenced motion features tend to focus

on salient regions. Furthermore, We combine spatial and motion saliency cues for saliency detection by fusing spatial features and fine-turned motion features in a multi-scale manner.

3 Our approach

3.1 Architecture overview

Research on human visual attention shows that moving objects tend to be more attractive and noticed [54, 55]. For video, the spatial features extracted from each frame and the motion information between consecutive frame are essential for saliency detection [31]. This inspired us to develop our MFHF with a two-stream [28] structure for extracting spatial features and motion features, as shown in Fig. 2(a). Our MFHF predicts video saliency map with four subnets, spatial, motion, hierarchical fusion and convGRU subnets. In detail, in the spatial subnet, we take the current frame $\mathbf{I}_t$ as input to extract five-level spatial features $\{\mathbf{SF}_t^k\}_{k=1}^5$ which generated from each block k of deep neural networks. In the motion subnet, the optical flow obtained by RAFT [26] is used as the input coarse motion frame and output multi-scale motion feature $\{\mathbf{MF}_t^k\}_{k=1}^5$ by incorporating dense residual cross connections with $\{\mathbf{SF}_t^k\}_{k=1}^5$. In the hierarchical fusion subnet, we combine both spatial and motion features in a multi-scale form to generate fused features $\{\mathbf{HF}_t^k\}_{k=1}^5$. Also we refine the intermediate saliency maps of the last three levels with the fused features to guide feature extraction. In order to learn the

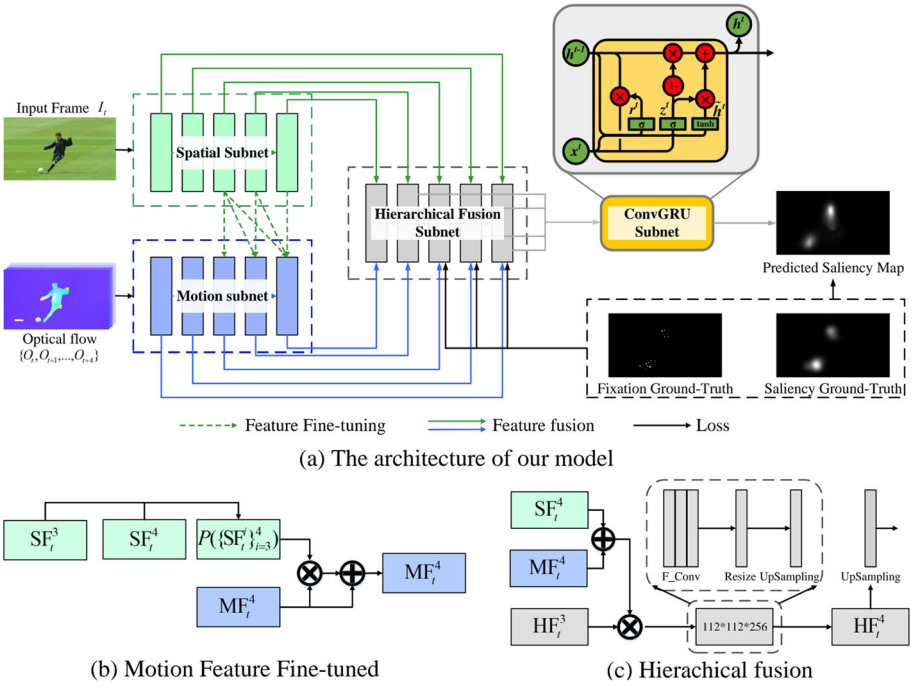

Fig. 2 (a) The overall structure of our MFHF. (b) An illustration of convGRU configuration. (c) Motion feature fine-tuning. (d) Feature fusion structure for the layer four in the hierarchical fusion subnet

inter-frame correlation of a video, we utilize convGRU [32] subnet to optimize the final result of the video saliency prediction.

3.2 Spatial and motion subnets

3.2.1 Spatial subnet

Considering that the visual information may be continuously lost in the convolution process and inter layer transmission, we expect to effectively extract spatial information of different scales. We build our spatial subnet based on ResNet-50 [56] which has faster forward and backward propagation as a residual network. Specifically, we retain the feature extraction layers of ResNet-50, and remove the fully connected layer to keep high-level spatial information. At time step t, the spatial subnet takes the current frame $\mathbf{I}_t$ as input, then produces the spatial features $\{\mathbf{SF}_t^k\}_{k=1}^5$ of different scales through five convolution blocks.

3.2.2 Motion subnet

For combination of multiscale spatial and motion features, we also use ResNet-50 to extract coarse motion features. Continuous optical flows $\{\mathbf{O}_{t-4}, \mathbf{O}_{t-3}, ..., \mathbf{O}_t\}$ from the neighboring frames of $\mathbf{I}_t$ are fed as inputs to obtain the corresponding motion features $\{\mathbf{MF}_t^k\}_{k=1}^5$. The optical flow method is the most mainstream method of motion feature extraction, it still produce some unsatisfactory results in the calculation process, as shown in Fig. 1(c). To get a more accurate and robust high-level optical flow estimation, we inject the spatial features of the last three layers $\{\mathbf{SF}_t^k\}_{k=3}^5$ with a cross connection to fine-tune higher level motion features $\{\mathbf{MF}_t^k\}_{k=3}^5$.

The motion fine-tuned process in the fourth layer of the motion subnet is shown in Fig. 2 (b). For reducing the loss of spatial information between convolutional layers and utilizing multi-scale characteristics effectively, we use a nonlinear mapping P to joint the third and fourth layer spatial features $\mathbf{SF}_t^3$, $\mathbf{SF}_t^4$. And then the joint spatial feature and $\mathbf{MF}_t^4$ are multiplied using a Hadamard product. Finally, the multiplication result is used to correct motion feature $\mathbf{MF}_t^4$. In generation, the feature fine-tuning process can be formulated as:

$$\mathbf{MF}_t^k = \mathbf{MF}_t^k + \mathbf{MF}_t^k \odot P(\{\mathbf{SF}_t^i\}_{i=3}^k), 3 \leq k \leq 5 \tag{1}$$

where '$\odot$' indicates the Hadamard operation, P is a nonlinear mapping that connect the spatial characteristics of different stages of the subnet.As is shown in Fig. 1,(d) represents the motion feature without fine-tuning process, and (e) represents the fine-tuned motion feature. We can find that compared with (d), the fine-tuned motion feature focus more on areas close to the saliency results, and the role of this fine-tuning process in objective indicators will also be further discussed in the ablation study.

3.3 Hierachical fusion subnet

For making full use of the important role of motion features in video sailency detection, we design a hierarchical fusion subnet to integrate the multi-scale spatial and motion features by using dense connections. In detail, we combine spatial features and motion features of each layer $\mathbf{SF}_t^k$ and $\mathbf{MF}_t^k$ to generate the fused feature $\mathbf{HF}_t^k$. As an example, the architecture of the fourth layer is shown in Fig. 2(c). The upsampled lower level fusion feature $\mathbf{HF}_t^3$ is used to

concatenate with the corresponding level features $\mathbf{SF}_t^4$ and $\mathbf{MF}_t^4$. In general, the hierarchical fusion operation is expressed as follows:

$$\mathbf{HF}_t^k = \begin{cases} h([\mathbf{SF}_t^k, \mathbf{MF}_t^k]), k = 1 \\ h([\mathbf{SF}_t^k + \mathbf{MF}_t^k, Up(\mathbf{HF}_t^{k-1})]), 1 < k \leq 5 \end{cases} \tag{2}$$

where h denotes the feature fusion operator, Up denotes Bilinear interpolation upsample operator, and [.] denotes the channel-wise concatenation operator. $\mathbf{SF}_t^k$ and $\mathbf{MF}_t^k$ denote the attentive spatial and fine-tuning motion features from the k convolutional layer of the spatial and motion subnet, respectively. Through the fusion operation, each stage of the fusion subnet will be supervised by the fusion features of the previous stage, the space and motion features of the corresponding scale, which can continuously update and optimize the sailency detection results.

For further monitoring the different stages of the fusion subnet, we upsample $\mathbf{HF}_t^k$ using Bilinear interpolation in the last four layers, and utilize a convolution layer with a $3 \times 3 \times 1$ kernel to obtain the corresponding output $\mathbf{S}_t^k$:

$$\mathbf{S}_t^k = Conv(Up(\mathbf{HF}_t^k)), 2 \leq k \leq 5 \tag{3}$$

where Up denotes the Bilinear interpolation, which upsampled to the size of 224×224 with strides of 8, 16, 4 and 4, respectively. As the output result of each frame after passing through the hierarchical fusion subnet, $\mathbf{S}_t^k$ will enter the convGRU subnet as input to learn the correlation between frames.

3.4 ConvGRU subnet

Compared with images, video has strong inter frame correlation that can not be ignored. The convGRU subnet, which has fewer parameters and higher computational efficiency compared to convLSTM [44], is utilized to model the inter-frame correlations and improve the dynamic saliency of videos. We first concatenate four-scale saliency maps $\{\mathbf{S}_t^k\}_{k=2}^5$ as the input x^t and then feed it into the convGRU unit, which introduces gating mechanism to learn the inter-frame correlation and temporal information:

$$\begin{aligned} z^t &= \sigma(W_h^z * h^{t-1} + W_x^z * x^t) \\ r^t &= \sigma(W_h^r * h^{t-1} + W_x^r * x^t) \\ \tilde{h}^t &= tanh(W_h^h * (r^t \odot h^{t-1}) + W_x^h * x^t) \\ h^t &= z^t \tilde{h}^t + (1 - z^t) h^{t-1} \end{aligned} \tag{4}$$

where r, z denote the reset and update gate, respectively, h represent the hidden states, W represents the learnable weights, '$*$' is the convolution operator. A sample of the convGRU configuration is shown in the top right of Fig. 2(a). Compared with LSTM [57] structure, convGRU [32] greatly reduces the amount of parameters in the fitting process and saves the time and calculation cost. The saliency detection result of our model $\mathbf{S}_t^{fin}$ will be produced by convolution of the hidden states h^t.

3.5 Loss function

We propose a loss function which is suitable for saliency prediction with our network structure. At time step t, both the final output $\mathbf{S}_t^{fin}$ and the intermediate layers outputs $\mathbf{S}_t^k$ are merged to supervise the network. The overall loss ℓ_{all} is designed as:

$$\ell_{all} = \ell(\mathbf{S}_t^{fin}, F, G) + \sum_{k=3}^{5} \ell(\mathbf{S}_t^k, F, G) \tag{5}$$

where G denotes the ground-truth saliency map, F denotes binary fixation map.

Similar to SAM [18], we design our loss ℓ by combining four loss metrics linearly, which can monitor the quality of the prediction results from several different quality factors. Specifically, our loss function can be expressed as:

$$\begin{aligned} \ell(S, F, G) =& \ell_{kl}(S, G) + \omega_1 \ell_{cc}(S, G) \\ &+ \omega_2 \ell_{nss}(S, F) + \omega_3 \ell_{sim}(S, G) \end{aligned} \tag{6}$$

where S denotes the predicted saliency map, and ω_1, ω_2, ω_3 indicates the weights of four loss metrics. They are set to 0.2, 0.1, 0.1 respectively.

ℓ_{kl} is obtained according to *Kullback-Leibler (KL) divergence* metric:

$$\ell_{kl}(S, G) = \sum_i S_i \log(\frac{G_i}{S_i}) \tag{7}$$

where i indexes the i^{th} pixel.

ℓ_{cc} is derived from *Linear Correlation Coefficient (CC)* metric that can be calculated using the following:

$$\ell_{cc}(S, G) = -\frac{\text{cov}(S, G)}{\rho(S)\rho(G)} \tag{8}$$

where $cov(\cdot)$ represents the calculation of the covariance operation, $\rho(\cdot)$ stands for the standard deviation.

ℓ_{nss} is derived from *Normalized Scanpath Saliency (NSS)* metric, which can be expressed as:

$$l_{nss}(S, F) = -\frac{1}{N} \sum_i \frac{S_i - \mu(S)}{\rho(S)} \times F_i \tag{9}$$

where $N = \sum_i F_i$ denotes the total number of fixated pixels, $\mu(\cdot)$ represents the calculation of mean, $\rho(\cdot)$ represents the standard deviation method.

ℓ_{sim} is derived from *Similarity (SIM)* metric, which is used to quantify similarity and can be represented as:

$$\ell_{sim}(S, G) = -\sum_i \min(S_i, G_i) \tag{10}$$

where S and G represents normalized probability distributions.

4 Experiments

In this subsection we will describe the datasets, evaluation metrics and implementation in detail.

Table 1 Statistics of four typical video saliency detection datasets we used

Dataset	Resolution	Training set	Testing set	Viewers
UCF sports	720*480	103	47	19
Hollywood-2	720*480	823	884	19
DIEM	1280*720	64	20	50
DHF1K	640*360	600	300	17

4.1 Experimental setup

1).Datasets: To evaluate our method, the four most popular datasets UCF Sports [23], Hollywood-2 [23], DIEM [58] and DHF1K [48] are used for performance analysis. The statistics for four datasets are summarized in Table 1.

UCF Sports is collected by 19 observers and divided into 103 and 47 videos for training and testing sets, respectively. Particularly, UCF Sports consists of a series of sport-related videos with a resolution of 720*480, which makes that the saliency detection model needs strong adaptability to deal with similar scene.

Hollywood-2 consists of a total of 1707 video data, which provides an important reference for a comprehensive evaluation of model performance. It contains 823 and 884 videos in the training and testing sets. Different from UCF Sports, Hollywood-2 has diverse background information and a wide range of moving objects, which make Hollywood-2 more challenging and difficult.

DIEM is observed by 50 volunteers, which is composed of multiple movie clips with extremely different styles and a high resolution of 1280*720. In DIEM, 64 videos are used for training and 20 videos are used for testing.

DHF1K is the most complex video saliency dataset. It consists of 1000 diverse videos, whereas only 700 videos have high-quality annotations, which are normally used for training/validating the model, and the remaining 300 videos are used for testing the model with

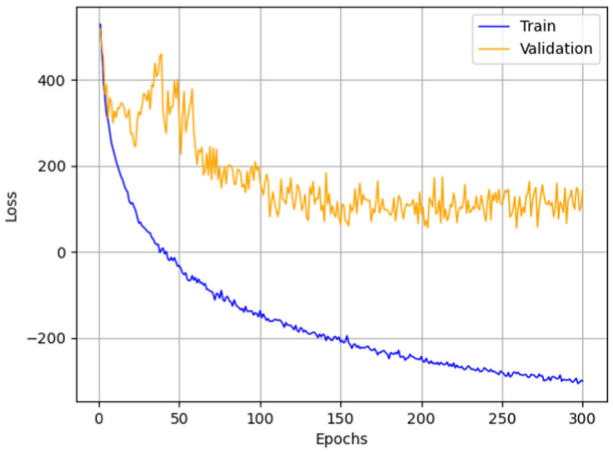

Fig. 3 The training and validation loss for each epoch in the training phase

the help of the owner. Unlike the previous two datasets, which collected fixations in a given visual task, both DHF1K and DIEM collect fixations with free-viewing.

2).Evaluation Metrics: We adopt five popular metrics to evaluate the results, including Area under the Curve by Judd(AUC-J) [59], Shuffled AUC(s-AUC) [60], Linear Correlation Cofficient(CC), Normalized Scanpath Saliency(NSS), and Similarity(SIM). For all of these metrics, higher scores indicate better performance.

3).Implementation Details: Our model is implemented with the Keras framework on a single Nvidia Tesla GPU (with 16G memory) and 2.2 GHz Intel Xeon CPU E5-2630 v4 CPU. During the trianing phase, we set the video batch size to 1 and frame batch size to 5.

Table 2 Quantitative comparison with 14 methods on UCF Sports

Methods	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
SALICON (ICCV 2015)	0.848	0.304	0.738	0.375	1.838
Deep-Net (CVPR 2016)	0.861	0.282	0.719	0.414	1.903
Shallow-Net (CVPR 2016)	0.846	0.276	0.691	0.382	1.789
Two-stream (TMM 2017)	0.832	0.264	0.685	0.343	1.753
DVA (TIP 2018)	0.872	0.339	0.725	0.439	2.311
DeepVS (ECCV 2018)	0.870	0.321	0.691	0.405	2.089
ACLNet (TPAMI 2019)	0.897	0.406	0.744	0.510	2.567
SalEMA (BMVC 2019)	0.906	0.431	0.740	0.544	2.638
TASED-Net (ICCV 2019)	0.899	0.469	0.752	0.582	2.920
STRA-Net (TIP 2019)	0.910	0.479	0.751	0.593	3.018
GDLC (TC 2020)	0.878	0.451	-	0.572	3.122
KSORA (PR 2020)	0.875	0.397	-	0.518	2.622
Chen et al. (Nc 2021)	0.910	0.488	0.761	0.601	2.916
STA3D (PRL 2021)	0.900	0.465	0.739	0.560	2.754
ECA-Net (NC 2022)	<u>0.917</u>	<u>0.498</u>	**0.797**	**0.636**	<u>3.189</u>
Ours	**0.920**	**0.510**	<u>0.791</u>	<u>0.635</u>	**3.509**

The best and suboptimal results are in **bold** and with <u>underline</u> respectively. "-" denotes that the results are not provided. Note that the results of DeepCT on UCF Sports are not provided

The whole model is trained in an end-to-end anner. The number of training epochs is set as 300. The learning rate is set to 1e-5, which remains unchanged during the training phase. Parameters of the model are learnd on the training data by Adam [61] optimizer.

Figure 3 shows the training loss and validation loss of the proposed model when trained on the UCF Sports dataset. The validation loss of the model is minimized when the number of epochs reaches 260, and we save the model weights at this point for testing.

Table 3 Quantitative comparison with 14 methods on Hollywood-2

Methods	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
SALICON (ICCV 2015)	0.856	0.321	0.711	0.425	2.013
Deep-Net (CVPR 2016)	0.884	0.300	0.736	0.451	2.066
Shallow-Net (CVPR 2016)	0.851	0.276	0.694	0.423	1.680
Two-stream (TMM 2017)	0.863	0.276	0.710	0.382	1.748
DVA (TIP 2018)	0.886	0.372	0.727	0.482	2.459
DeepVS (ECCV 2018)	0.887	0.356	0.693	0.446	2.313
ACLNet (TPAMI 2019)	0.913	0.542	0.757	0.623	3.086
SalEMA (BMVC 2019)	0.919	0.487	0.708	0.613	3.186
TASED-Net (ICCV 2019)	0.918	0.507	0.768	0.646	3.302
STRA-Net (TIP 2019)	0.923	0.536	0.774	0.662	<u>3.479</u>
GDLC (TC 2020)	0.892	0.409	-	0.544	3.086
KSORA (PR 2020)	0.889	0.524	-	0.628	3.108
Chen et al. (Nc 2021)	0.920	**0.548**	0.778	0.671	3.418
STA3D (PRL 2021)	0.927	0.534	0.731	0.659	3.329
ECA-Net (NC 2022)	<u>0.929</u>	0.526	**0.806**	<u>0.673</u>	3.380
Ours	**0.930**	<u>0.543</u>	<u>0.794</u>	**0.676**	**3.623**

The best and suboptimal results are in **bold** and with <u>underline</u> respectively. "-" denotes that the results are not provided. Note that the results of DeepCT on Hollywood-2 are not provided

4.2 Comparison results

We rigorously compare our MFHF with 16 other state-of-the-art methods, including SALICON [18], Two-streams [28], DeepNet [19], ACLNet [48], DVA [45], DeepVS [29], Shallow-Net [19], SalEMA [62], TASED-Net[63], GDLC [64], STRA-NET [31], KSORA [65], DeepCT [66], STA-3D [67], ECA-Net [68] and the video saliency detection model proposed by Chen et al. in [69]. For a fair comparison, we directly adopt the published results in the corresponding paper.

4.2.1 Performance on UCF sports

We trained MFHF with UCF Sports training videos and evaluated it on the test set in the normal way. Table 2 shows the performance of all models. Compared to other video saliency detection methods, our MFHF has advantages in *AUC-J*, *SIM* and *NSS* metrics. This may benefit from the fine-tuned motion features resulting more efficient and reliable for describing

Table 4 Quantitative comparison with 14 methods on DHF1K

Methods	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
SALICON (ICCV 2015)	0.857	0.232	0.590	0.327	1.901
Deep-Net (CVPR 2016)	0.855	0.201	0.592	0.331	1.775
Shallow-Net (CVPR 2016)	0.833	0.182	0.529	0.295	1.509
Two-stream (TMM 2017)	0.834	0.197	0.581	0.325	1.632
DVA (TIP 2018)	0.860	0.262	0.595	0.358	2.013
DeepVS (ECCV 2018)	0.856	0.256	0.583	0.344	1.911
ACLNet (TPAMI 2019)	0.890	0.315	0.601	0.434	2.354
SalEMA (BMVC 2019)	0.890	**0.466**	0.667	0.449	2.574
TASED-Net (ICCV 2019)	<u>0.895</u>	0.361	**0.712**	**0.470**	**2.667**
STRA-Net (TIP 2019)	<u>0.895</u>	0.355	0.663	0.458	2.558
DeepCT (PR 2020)	**0.896**	0.342	0.673	0.457	2.513
Chen et al. (Nc 2021)	<u>0.895</u>	0.349	<u>0.677</u>	0.459	2.530
Ours	0.894	<u>0.376</u>	0.655	<u>0.467</u>	<u>2.618</u>

The best and suboptimal results are in **bold** and with <u>underline</u> respectively. Note that the results of GDLC and KSORA on DHF1K are not provided

moving regions. The motion feature visualisation results, given in Fig. 1(d) and (e), also show that the fine-tuned motion features are more focused and provide clearer saliency cues than the original motion features.

4.2.2 Performance on Hollywood-2

All 823 videos in the training set are used to train our model, and Table 3 shows the performance of our MFHF and other 15 models on Hollywood-2. The proposed method achieves the best results for the *AUC-J*, *SIM* and *NSS* metrics and the next best results for the *s-AUC* and *CC* metrics. It may be due to our hierarchical fusion subnet, which can retain more multi-scale information and achieve better adaptability in complex scenarios. Simultaneously, our multi-level loss function design also plays a vital role in the performance improvement, which effectively supervises the extraction and fusion of features.

4.2.3 Performance on DHF1K

As suggested in [48], we also split the video sequences to 600/100/300 to train/validate/test our model. Quantitative results on the test set of DHF1K are shown in Table 4. From the Table 4, we can see that TASED-Net achieved the best performance on four metrics although it doesn't perform well on the first two datasets. The proposed method gives better results compared to the rest methods.

To further understand the efficiency of our model, we analyze the influence of frame batch sizes on the detection performance. TASED-Net gets the highest s-AUC value on DHF1K, which 32 past frames are used for the next frame saliency prediction. For our model, only 6 frames are included for estimation. Table 5 shows the comparisons of our model and three variants of TASED-Net on DHF1K, which includes the results of input 4, 8 and 32 frames given in [63]. It indicates the performance can vary with the number of input frames. Although fewer frames are involved in the network, our model gets the best performance on two metrics. The proposed model is qualified to capture spatio-temporal saliency cues with fewer frames.

4.2.4 Performance on DIEM

DIEM has 84 high-resolution videos with plentiful common life scenes, in which normally 64 videos are used for training, and 20 videos are left for testing. In [31], authors compared the performance of STRA-Net on DIEM with different other training datasets and demonstrated that more training samples could improve the performance of the model and training with Hollywood-2 get the comparable results. To evaluate the generalization ability, all 20 testing

Table 5 Quantitative comparison with 3 varients of TASED-Net on the validation set of DHF1K

Methods	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
TASED-Net (4)	0.887	0.327	0.689	0.441	2.434
TASED-Net (8)	0.889	0.348	<u>0.696</u>	0.46	<u>2.585</u>
TASED-Net (32)	<u>0.894</u>	<u>0.362</u>	**0.718**	**0.481**	**2.706**
Ours (6)	**0.896**	**0.373**	0.658	<u>0.462</u>	2.571

The best and suboptimal results are in **bold** and with <u>underline</u> respectively

Table 6 Quantitative comparison with 14 methods on DIEM

Methods	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
SALICON (ICCV 2015)	0.793	0.171	0.674	0.210	1.650
Deep-Net (CVPR 2016)	0.849	0.164	0.697	0.291	1.650
Shallow-Net (CVPR 2016)	0.838	0.188	0.620	0.297	1.646
Two-stream (TMM 2017)	0.859	0.256	0.682	0.366	2.171
DVA (TIP 2018)	0.868	0.237	**0.721**	0.386	2.347
DeepVS (ECCV 2018)	0.857	0.238	0.693	0.371	2.235
ACLNet (TPAMI 2019)	0.881	0.277	0.693	0.396	2.368
STRA-Net (TIP 2019)	0.870	0.306	0.678	0.408	**2.452**
DeepCT (PR 2020)	0.875	-	-	**0.504**	2.22
Ours	**0.883**	**0.388**	0.962	0.482	2.335

The best and suboptimal results are in **bold** and with underline respectively. "-" denotes that the results are not provided. Note that the results of SalEMA, TASED-Net, GDLC, KSORA, and the model proposed by Chen et al. on DIEM are not provided

videos are used as test samples following [31]. For simplicity, we test the performance of our network which only be trained with Hollywood-2 and the results are shown in Table 6. Noted that we only list the best results of STRA-Net on DIEM, which uses plenty of training sets, while we only use the parts of training data. The results are shown in Table 6 clearly illustrate our approach achieves competitive performance on most metrics and has comparable generalization capability with other advanced models.

Table 7 Ablation study on UCF sports

idx	motion feature fine-tuned	hierachical fusion	convGRU	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
1				0.872	0.433	0.735	0.523	2.56
2	✓			0.891	0.459	0.756	0.564	2.763
3		✓		0.880	0.462	0.738	0.549	2.770
4	✓	✓		0.895	0.494	0.762	0.595	3.167
5	✓	✓	✓	**0.920**	**0.510**	**0.791**	**0.635**	**3.509**

The best results are in **bold**

Table 8 Quantitative comparison of model variants with different settings on UCF Sports

idx	spatial enhancement	motion enhancement	max fusion	convolutional fusion	hierachical fusion	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
1	✓				✓	0.867	0.448	0.731	0.525	2.795
2		✓	✓			0.886	0.467	0.737	0.467	2.828
3		✓		✓		0.887	0.477	0.761	0.575	2.918
4		✓			✓	**0.895**	**0.494**	**0.762**	**0.595**	**3.167**

The best results are in **bold**

Table 9 Quantitative comparison with 14 methods on DHF1K

Side outputs	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
HF-Output1	0.881	0.484	0.759	0.581	2.953
HF-Output2	0.891	0.486	0.760	0.589	3.055
HF-Output3	0.893	0.490	0.761	0.591	3.089
Ours	**0.895**	**0.494**	**0.762**	**0.595**	**3.167**

4.3 Ablation study

From the previous subsection, our MFHF performs well on multiple datasets, which may be due to our motion fine-tuning and hierarchical fusion module. For verifying the effectiveness of each component, we designed kinds of MFHF variants and tested them on UCF sports.

We firstly explore the effectiveness of the proposed motion feature fine-tuned module and hierarchical fusion module in this section. The first row in Table 7 is the baseline which have the two-stream structure, the fifth row is our method and in rows 2 to 4 we have different combinations of our modules. The effectiveness of the motion feature fine-tuned module and hierarchical fusion module can be verified from the comparison results in rows 1-3. The result in line 4 shows that the above two modules are valid when combined. The last two rows of the results in Table 7 demonstrate the efficiency of the ConvGRU as well. Combining all of them achieves the best performance.

To further verify the effectiveness of the structure of the proposed model, We introduce modules from other methods to conduct comparative experiments. We used the spatial enhancement module in STRA-Net [31] to replace the motion feature fine-tuned module,

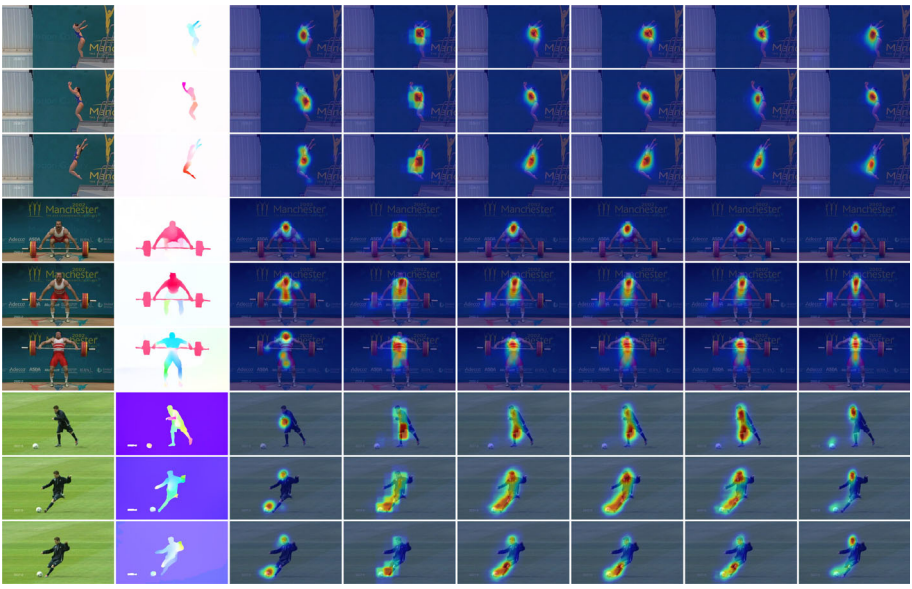

Fig. 4 Visualization of saliency detection. (a) Frame. (b) Optical flow. (c) Ground truth. (d) HF-Output1. (e) HF-Output2. (f) HF-Output3. (g) Ours(*w/o convGRU*). (h) Ours(*w. convGRU*)

comparison results are shown in the first and fourth rows of Table 7. Compared to the spatial enhancement module in STRA-Net, the motion fine-tuning module we used makes more effective use of motion information to model video saliency. To verify the effectiveness of hierarchical fusion, we used fusion machanisms such as convolutional fusion [28] and max fusion [28] in our framework, they only fuse single-layer features. We can see from the comparison of the last three rows that the hierarchical fusion machanism achieves better performance using multi-scale contextual features (Table 8).

In addition, we compared the saliency maps generated by the fusion features *HF-Output1*, *HF-Output2* and *HF-Output3*, which are from the last three layers of the hierarchical fusion subnet. These saliency maps are represented as S_t^3, S_t^4, S_t^5 in Section 3.3, respectively. The overall efficiency of multiscale fusion features for video saliency was assessed by comparing the results shown in of Table 9 and the visualization results shown in Fig. 4. These show that the multilevel features is helpful for modeling video saliency, the deeper hierarchical fused features is better to distinguish saliency parts in the videos.

5 Conclusions

In this paper, we propose a novel spatial and motion dual-stream framework to model video saliency detection. In order to get saliency related features, a dual-stream network was introduced to extract multiscale spatial features and then used to fine-tune motion features in a dense residual cross connection architecture. With the help of the higher level semantic spatial features, the fine-tuned motion features can get some saliency related features. Then, we fuse the multi-scale features with the hierarchical fusion subnet to retain more contextual saliency information. Also we integrate the multi-scale saliency map at the same frame to a loss function for monitoring the saliency detection process. ConvGRU subnet is revised to get the relationship between the frames. Extensive results on four video saliency benchmark datasets demonstrate the superiority of the proposed model to precisely predict dynamic human fixations. The ablation experiments show the necessity and effectiveness of each component in our model. Motion features play an important role in the video processing, if the fine-tune motion feature can implement in the transformer framework, the final results can further be improved.

Acknowledgements This research was supported by the National Science and Technology Major Project (Grant No. 2020YFA0713504), the National Natural Science Foundation of China (Nos. 62376238, 62372170), and the Scientific Research Foundation of Education Department of Hunan Province of China (Grant Nos. 21A0109)

Data Availibility The datasets generated during and/or analysed during the current study are available in the DHF1K repository, https://github.com/wenguanwang/DHF1K.

Declarations

Conflicts of interest The authors declare that there are no conflicts of interest.

References

1. Wang X, Qi C (2020) Detecting action-relevant regions for action recognition using a three-stage saliency detection technique. Multimed Tools Appl 79(11):7413–7433

2. Cizmeciler K, Erdem E, Erdem A (2022) Leveraging semantic saliency maps for query-specific video summarization. Multimed Tools Appl 81(12):17457–17482
3. Ullah J, Khan A, Jaffar MA (2018) Motion cues and saliency based unconstrained video segmentation. Multimed Tools Appl 77(6):7429–7446
4. Li S, Xu M, Wang Z, Sun X (2016) Optimal bit allocation for ctu level rate control in hevc. IEEE Trans Circ Syst Video Technol 27(11):2409–2424
5. Xu M, Liu Y, Hu R, He F (2018) Find who to look at: turning from action to saliency. IEEE Trans Image Proc 27(9):4529–4544
6. Chen C, Li S, Wang Y, Qin H, Hao A (2017) Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion. IEEE Trans Image Proc 26(7):3156–3170
7. Chen C, Li S, Qin H, Pan Z, Yang G (2018) Bilevel feature learning for video saliency detection. IEEE Trans Multimed 20(12):3324–3336
8. Li Y, Li S, Chen C, Hao A, Qin H (2019) Accurate and robust video saliency detection via self-paced diffusion. IEEE Trans Multimed 22(5):1153–1167
9. Chen C, Wang G, Peng C, Zhang X, Qin H (2019) Improved robust video saliency detection based on long-term spatial-temporal information. IEEE Trans Image Proc 29:1090–1100
10. Zhang P, Liu J, Wang X, Pu T, Fei C, Guo Z (2020) Stereoscopic video saliency detection based on spatiotemporal correlation and depth confidence optimization. Neurocomputing 377:256–268
11. Wang G, Chen C, Fan D, Hao A, Qin H (2021) Weakly supervised visual-auditory saliency detection with multigranularity perception. arXiv preprint arXiv:2112.13697
12. Li H, Chen G, Li G, Yu Y (2019) Motion guided attention for video salient object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7274–7283
13. Chen C, Song J, Peng C, Wang G, Fang Y (2021) A novel video salient object detection method via semisupervised motion quality perception. IEEE Trans Circ Syst Video Technol 32(5):2732–2745
14. Chen C, Wang H, Fang Y, Peng C (2022) A novel long-term iterative mining scheme for video salient object detection. IEEE Trans Circ Syst Video Technol 32(11):7662–7676
15. Borji A, Cheng M-M, Jiang H, Li J (2015) Salient object detection: a benchmark. IEEE Trans Image Proc 24(12):5706–5722
16. Vig E, Dorr M, Cox D (2014) Large-scale optimization of hierarchical features for saliency prediction in natural images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2798–2805
17. Liu N, Han J, Zhang D, Wen S, Liu T (2015) Predicting eye fixations using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 362–370
18. Huang X, Shen C, Boix X, Zhao Q (2015) Salicon: reducing the semantic gap in saliency prediction by adapting deep neural networks. In: Proceedings of the IEEE international conference on computer vision, pp 262–270
19. Pan J, Sayrol E, Giro-i-Nieto X, McGuinness K, O'Connor NE (2016) Shallow and deep convolutional networks for saliency prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 598–606
20. Liu N, Han J, Liu T, Li X (2016) Learning to predict eye fixations via multiresolution convolutional neural networks. IEEE Trans Neur Netw Learn Syst 29(2):392–404
21. Kruthiventi SS, Ayush K, Babu RV (2017) Deepfix: a fully convolutional neural network for predicting human eye fixations. IEEE Trans Image Proc 26(9):4446–4456
22. Liu N, Han J (2018) A deep spatial contextual long-term recurrent convolutional network for saliency detection. IEEE Trans Image Proc 27(7):3264–3274
23. Mathe S, Sminchisescu C (2014) Actions in the eye: dynamic gaze datasets and learnt saliency models for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(7):1408–1424
24. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Mach Intell 20(11):1254–1259
25. Gibson JJ (1950) The perception of the visual world
26. Teed Z, Deng J (2020) Raft: recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. Springer, pp 402–419
27. Cong R, Song W, Lei J, Yue G, Zhao Y, Kwong S (2022) Psnet: parallel symmetric network for video salient object detection. IEEE Trans Emerg Top Comput Intell
28. Bak C, Kocak A, Erdem E, Erdem A (2017) Spatio-temporal saliency networks for dynamic saliency prediction. IEEE Trans Multimed 20(7):1688–1698
29. Jiang L, Xu M, Liu T, Qiao M, Wang Z (2018) Deepvs: a deep learning based video saliency prediction approach. In: Proceedings of the European conference on computer vision (eccv), pp 602–617
30. Zhang K, Chen Z (2018) Video saliency prediction based on spatial-temporal two-stream network. IEEE Trans Circ Syst Video Technol 29(12):3544–3557

31. Lai Q, Wang W, Sun H, Shen J (2019) Video saliency prediction using spatiotemporal residual attentive networks. IEEE Trans Image Proc 29:1113–1126
32. Ballas N, Yao L, Pal C, Courville A (2015) Delving deeper into convolutional networks for learning video representations. arXiv preprint arXiv:1511.06432
33. Srinivasu PN, Bhoi AK, Jhaveri RH, Reddy GT, Bilal M (2021) Probabilistic deep q network for real-time path planning in censorious robotic procedures using force sensors. J Real-Time Image Proc 18(5):1773–1785
34. Craye C, Filliat D, Goudou J-F (2016) Environment exploration for object-based visual saliency learning. In: 2016 IEEE international conference on robotics and automation (ICRA), pp 2303–2309. IEEE
35. Le Meur O, Le Callet P, Barba D, Thoreau D (2006) A coherent computational approach to model bottom-up visual attention. IEEE Trans Pattern Ana Mach Intell 28(5):802–817
36. Zhang L, Tong MH, Marks TK, Shan H, Cottrell GW (2008) Sun: a bayesian framework for saliency using natural statistics. J Vision 8(7):32–32
37. Gao D, Vasconcelos N (2005) Discriminant saliency for visual recognition from cluttered scenes. In: Adv Neural Inf Proc Syst, pp 481–488
38. Bruce N, Tsotsos J (2005) Saliency based on information maximization. Adv Neural Inf Proc Syst 18:155–162
39. Cheng M-M, Mitra NJ, Huang X, Torr PH, Hu S-M (2014) Global contrast based salient region detection. IEEE Trans Pattern Anal Machi Intell 37(3):569–582
40. Xu M, Ren Y, Wang Z (2015) Learning to predict saliency on face images. In: Proceedings of the IEEE international conference on computer vision, pp 3907–3915
41. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
42. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
43. Cornia M, Baraldi L, Serra G, Cucchiara R (2018) Predicting human eye fixations via an lstm-based saliency attentive model. IEEE Trans Image Processing 27(10):5142–5154
44. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-c (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. Adv Neural Inf Proc Syst 28
45. Wang W, Shen J (2017) Deep visual attention prediction. IEEE Trans Image Proc 27(5):2368–2378
46. Guo C, Ma Q, Zhang L (2008) Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: 2008 IEEE conference on computer vision and pattern recognition, pp 1–8 IEEE
47. Itti L, Dhavale N, Pighin F (2003) Realistic avatar eye and head animation using a neurobiological model of visual attention. In: Applications and science of neural networks, fuzzy systems, and evolutionary computation VI, vol 5200, pp 64–78. SPIE
48. Wang W, Shen J, Xie J, Cheng M-M, Ling H, Borji A (2019) Revisiting video saliency prediction in the deep learning era. IEEE Trans Pattern Anal Mach Intell 1–1. https://doi.org/10.1109/TPAMI.2019.2924417
49. Zhu S, Chang Q, L, Q (2022) Video saliency aware intelligent hd video compression with the improvement of visual quality and the reduction of coding complexity. Neural Computing and Applications 1–20
50. Chen C, Wang G, Peng C, Fang Y, Zhang D, Qin H (2021) Exploring rich and efficient spatial temporal interactions for real-time video salient object detection. IEEE Trans Image Proc 30:3995–4007
51. Zhang F, Woodford OJ, Prisacariu VA, Torr PH (2021) Separable flow: Learning motion cost volumes for optical flow estimation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 10807–10817
52. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
53. 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: Proceedings of the IEEE international conference on computer vision, pp 2758–2766
54. Mital P, mith TJ, Luke S, Henderson J (2013) Do low-level visual features have a causal influence on gaze during dynamic scene viewing? J Vision 13(9):144–144
55. Abrams RA (2003) Christ SE (2003) Motion onset captures attention. Psychol Sci 14(5):427–432
56. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
57. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8): 1735–1780
58. Mital PK, Smith TJ, Hill RL, Henderson JM (2011) Clustering of gaze during dynamic scene viewing is predicted by motion. Cogn Comput 3(1):5–24
59. Judd T, Ehinger K, Durand F, Torralba A (2009) Learning to predict where humans look. In: 2009 IEEE 12th international conference on computer vision, pp 2106–2113. IEEE

60. Borji A, Tavakoli HR, Sihite DN, Itti L (2013) Analysis of scores, datasets, and models in visual saliency prediction. In: Proceedings of the IEEE international conference on computer vision, pp 921–928
61. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
62. Linardos P, Mohedano E, Nieto JJ, O'Connor NE, Giro-i-Nieto X, McGuinness K (2019) Simple vs complex temporal recurrences for video saliency prediction. arXiv preprint arXiv:1907.01869
63. Min K, Corso JJ (2019) Tased-net: temporally-aggregating spatial encoder-decoder network for video saliency detection. In: Proceedings of the IEEE international conference on computer vision, pp 2394–2403
64. Wang Z, Zhou Z, Lu H, Hu Q, Jiang J (2020) Video saliency prediction via joint discrimination and local consistency. IEEE Transactions on Cybernetics
65. Wang Z, Zhou Z, Lu H, Jiang J (2020) Global and local sensitivity guided key salient object re-augmentation for video saliency detection. Pattern Recogn 103:107275
66. Jiang L, Xu M, Zhang S, Sigal L (2020) Deepct: a novel deep complex-valued network with learnable transform for video saliency prediction. Pattern Recogn 102:107234
67. Zou W, Zhuo S, Tang Y, Tian S, Li X, Xu C (2021) Sta3d: spatiotemporally attentive 3d network for video saliency prediction. Pattern Recognition Letters 147:78–84
68. Xue H, Sun M, Liang Y (2022) Ecanet: explicit cyclic attention-based network for video saliency prediction. Neurocomput 468:233–244
69. Chen J, Li Z, Jin Y, Ren D, Ling H (2021) Video saliency prediction via spatio-temporal reasoning. Neurocomput 462:59–68