# Video Saliency Prediction Using Enhanced Spatiotemporal Alignment Network

Jin Chen $^1$ , Huihui Song $^1$ , Kaihua Zhang $^{1*}$ , Bo Liu $^2$ , Qingshan Liu $^1$   $^1$ B-DAT,CICAEET, Nanjing University of Information Science and Technology, Nanjing, China  $^2$ JD Finance America Corporation

# Abstract

Due to a variety of motions across different frames, it is highly challenging to learn an effective spatiotemporal representation for accurate video saliency prediction (VSP). To address this issue, we develop an effective spatiotemporal feature alignment network tailored to VSP, mainly including two key subnetworks: a multi-scale deformable convolutional alignment network (MDAN) and a bidirectional convolutional Long Short-Term Memory (Bi-ConvLSTM) network. The MDAN learns to align the features of the neighboring frames to the reference one in a coarse-to-fine manner, which can well handle various motions. Specifically, the MDAN owns a pyramidal feature hierarchy structure that first leverages deformable convolution (Dconv) to align the lowerresolution features across frames, and then aggregates the aligned features to align the higher-resolution features, progressively enhancing the features from top to bottom. The output of MDAN is then fed into the Bi-ConvLSTM for further enhancement, which captures the useful long-time temporal information along forward and backward timing directions to effectively guide attention orientation shift prediction under complex scene transformation. Finally, the enhanced features are decoded to generate the predicted saliency map. The proposed model is trained end-to-end without any intricate post processing. Extensive evaluations on four VSP benchmark datasets demon-

<sup>\*</sup>Corresponding author. Phone number: 086-13851581017 Email address: zhkhua@gmail.com (Kaihua Zhang<sup>1</sup>)

strate that the proposed method achieves favorable performance against state-of-the-art methods. The source codes and all the results will be released at https://github.com/cj4L/ESAN-VSP.

Keywords: Video saliency prediction, feature alignment, deformable convolution, bidirectional ConvLSTM

#### 1. Introduction

The objective of VSP is to faithfully model the human's gaze eye-fixation when watching a dynamic scene. As a branch of object saliency detection, VSP contributes to the cognitive research of human vision attention through understanding and analyzing dynamic video frames. VSP has been widely used to assist various computer vision applications, such as autonomous driving [1], object detection and recognition [2, 3], video segmentation [4, 5], visual tracking [6], video captioning [7], human-robot interaction [8] and video summarization [9], to name a few.

In recent years, benefiting from the breakthrough of deep learning (DL), a variety of DL-based VSP methods [10, 11, 12, 13, 14] have been proposed to predict eye-fixation allocation in each frame. Those DL-based techniques take advantage of a large amount of labeled eye-tracking data to learn an effective semantic feature representation in an end-to-end manner that can accurately predict the salient object locations, greatly outperforming the traditional methods [15, 16, 17, 18] with hand-crafted features. Different from the static image saliency detection that only needs to consider spatial cues in one image, VSP should also take into account the temporal cues to handle challenging motion scenarios across frames. The human eye's fixation mechanism is affected by subjective consciousness, which shifts the target of attention as the scene changes caused by camera motion, light change, scene scaling and fast target movement, etc. Complex motions from background and inconsistent foreground patterns together result in the difficulty of VSP. Hence, how to learn an effective spatiotemporal feature representation that can well guide attention orientation shift

prediction under various scene transitions plays a key role in VSP.

Existing approaches for VSP [19, 20, 21, 21] often explicitly estimate opticalflow field between the reference frame and its adjacent frames to capture shortterm temporal information and then simply fuse the temporal and spatial information to complement their characteristics. STSConvNet [19] extracts temporal information using optical flow between consecutive video frames and investigates different ways to integrate spatial and temporal cues within a deep two-stream spatiotemporal network architecture for VSP. OM-CNN [20] leverages a CNNbased optical-flow estimation method to measure the motion intensity in all frames to solve dynamic consistence restriction. STRA-Net [21] makes use of two parallel DNN streams to extract the spatial and temporal cues with optical flows as input. Besides, the aforementioned methods further leverage LSTM or Gated Recurrent Unit (GRU) to capature the long-term temporal information across frames to learn effective spatiotemporal feature representations for VSP. OM-CNN [20] designs a 2C-LSTM architecture to learn temporal correlation of high-dimensional features for VSP. ACLNet [22] presents an attentive CNN-LSTM mechanism to predict human gaze, and encodes static attention to learn a dynamic salient representation by using frame-wise image saliency maps. SalEMA [23] extends an image saliency structure to VSP by integrating a ConvLSTM module and wrapping a convolutional layer with a temporal exponential moving average. STRA-Net [21] develops a spatiotemporal residual attentive network that leverages convolutional GRUs to model the attention transitions across video frames. Despite demonstrated success of widely applying LSTM or GRU to VSP, all the aforementioned methods only leverage forward sequence modeling that only captures the forward-frame information, while omitting the useful backward-frame cues that are also helpful to enhance the spatiotemporal feature representations. To address this issue, we design the Bi-ConvLSTM that makes full use of the forward and backward frame cues to learn a robust feature representation.

However, directly using LSTM or GRU to learn spatiotemporal representations from the simply fused short-term spatiotemporal cues cannot work well when the attention target appearances across frames suffer from severe distortions due to large and complex motions. An effective approach to address this issue is to employ multi-frame alignment technique to enhance feature representation, which has been widely applied in video super-resolution [24, 25, 26] for motion compensation. However, we have not found any work that applies feature alignment technique to VSP. To this end, we design a novel spatiotemporal alignment network to implement feature alignment between the reference frame and its adjacent frame via Dconv [27]. The aligned features are then fed into the Bi-ConvLSTM to learn a robust spatiotemporal feature representation for VSP.

In summary, our main contributions are summarized into threefold:

- A Muti-scale Deformable convolutional Alignment Network (MDAN) is designed to align the features across frames with the help of Dconv [27].
   To the best of our knowledge, this is the first work to apply Dconv to VSP.
- A novel Bi-ConvLSTM is introduced to effectively model the long-term attention shift across video frames, which makes full use of the long-term temporal context information in the forward and backward time directions.
- Extensive evaluations on four VSP benchmarks including DHF1K [22], HollyWood2 [28], UCF-sports [28] and DIEM [29] demonstrate the proposed method achieves competing performance against state-of-the-art methods.

#### 2. Related Work

# 2.1. Computational Models for VSP

Existing VSP methods could be roughly grouped into two categories including static models [30, 13, 12, 10, 31, 14] and dynamic ones [32, 22, 33, 21]. With the help of large-scale eye-tracking labeled datasets for training, numerous DL-based static saliency models for VSP [30, 13, 12, 10, 31, 14] have been proposed

and achieved remarkable performance boosting compared to the traditional approaches. eDN [30] follows an entirely automatic data-driven approach to perform a large-scale search for an optimal ensemble of deep CNN features, and then trains an SVM classifier to predict the saliency maps. DeepFix [13], DeepNet [12] and SALICON [10] leverage large-scale eye-tracking data to fine-tune the classical image classification networks to generate the corresponding eye-fixation maps. Mr-CNN [31] employs a multi-resolution CNN guided by both bottom-up visual saliency and top-down visual cues to predict visual saliency. DVA [14] is based on a skip-layer network structure, which estimates eye-fixation from multiple convolutional layers with various reception fields.

Another research branch of VSP focuses on simulating eye fixation behavior in dynamic scenes [32, 22, 33, 21]. The traditional dynamic approaches [15, 16, 17] leverage hand-crafted spatiotemporal features to model visual saliency, which cannot capture rich semantic information from the attention targets that is essential for accurate VSP. To address this issue, numerous DL-based dynamic saliency models [32, 22, 33, 21] have been developed with promising performance. SalGan [32] proposes a data-driven metric based VSP method that is trained with an adversarial loss function, yielding saliency maps that resemble the ground-truth. ACLNet [22] releases a benchmark dataset for predicting human eye movements during dynamic scene free viewing and proposes a CNN-LSTM network with an attention mechanism for VSP. SalEMA [23] introduces a conceptually simple exponential moving average of an internal convolutional state to modify existing network architectures for VSP. TASED-Net [33] designs a 3D fully-convolutional network structure and decodes the encoded features spatially while aggregating all the temporal information for VSP. STRA-Net [21] develops a residual attentive learning network architecture, which enhances the spatiotemporal features by a composite attention mechanism for VSP.

# 2.2. Deformable Convolutional Networks

The deformable convolutional network (DCN) proposed by [34] aims to enhance the capability of regular convolutions by learning additional offsets from

its local neighborhood, and allows the network to adaptively capture more contextual information in a larger receptive field. DCNv2 [27] reformulates Dconv and introduces a modulation mechanism that expands the scope of deformation modeling through a more comprehensive integration of deformable convolution within the network. The superior performance of Dconv has been demonstrated in some other computer vision tasks including video super-resolution [25, 26], object detection [35], image classification [36] and crowd understanding [37]. TDAN and EDVR [25, 26] use Doonv to align features between the reference frame and its corresponding supporting frames for motion compensation in video restoration task. RepPoints [35] leverages Dconv to develop a flexible object representation for accurate geometric localization as well as semantic feature extraction. DHCNet [36] achieves better classification performance for hyperspectral image classification by applying the regular convolutions on the Dconv feature maps. ADCrowdNet [37] designs an attention-injective Dconv to address the accuracy degradation issue in highly-congested noisy scenes for crowd understanding task.

#### 3. Proposed Approach

# 3.1. Architecture Overview

Figure 1 shows an overview of the proposed model for VSP. Given N consecutive frames  $\mathcal{I} = \{I_t\}_{t=1}^N$  in a video sequence, our aim is to learn a deep CNN  $f_{\text{CNN}}$  that outputs a set of corresponding visual saliency maps  $\mathcal{M} = \{M_t\}_{t=1}^N$ :

$$\mathcal{M} = f_{\text{CNN}}(\mathcal{I}; \boldsymbol{\theta}_{\text{CNN}}), \tag{1}$$

where  $\theta_{\text{CNN}}$  denotes the whole network parameters to be optimized.  $f_{\text{CNN}}$  consists of three sub-networks: the MDAN  $f_{\text{align}}$ , the Bi-ConvLSTM  $f_{\text{Bi-ConvLSTM}}$  and the decoder  $f_{\text{decoder}}$ . Specifically, the design of  $f_{\text{align}}$  is inspired by the renowned DCN [34, 27]. We align each neighboring frame to the reference one at feature level by progressively aligning and aggregating the multi-level features from top to down. By fusing the spatiotemporal cues across frames at

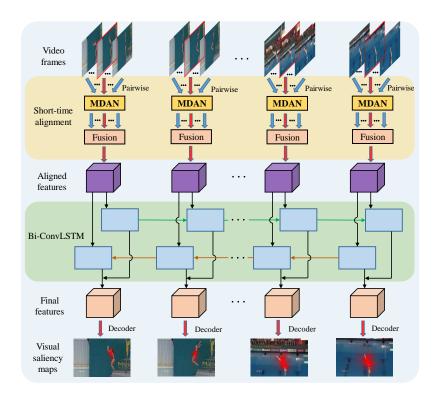


Figure 1: Overview of the proposed VSP framework. Firstly, a sequence of video frames are fed into the MDAN to align the reference and its neighboring frames, yielding the aligned features that can well capture short-term spatiotemporal information. Then, the aligned features are sent to the Bi-ConvLSTM, generating the features that encodes long-term spatiotemporal information. Finally, the output features are decoded to generate the predicted saliency map.

different semantic levels,  $f_{\text{align}}$  enables to well handle diverse motions across frames that can severely affect accurately predicting attention shifts in VSP. Given a sequential of 2T + 1 frames with the reference frame t at the center,  $f_{\text{align}}$  aligns the left-and the right-side T neighboring frames to the reference frame t, respectively and then fuses them to generate the enhanced reference frame features  $A_t$ :

$$\mathbf{A}_t = f_{\text{align}}(\{I_n\}_{n=t-T}^{t+T}; \boldsymbol{\theta}_{\text{align}}), \tag{2}$$

where  $\theta_{\rm align}$  denotes the corresponding network parameters of MDAN to be optimized.

Although the features  $A_t$  in (2) are strengthen by the features of the neigh-

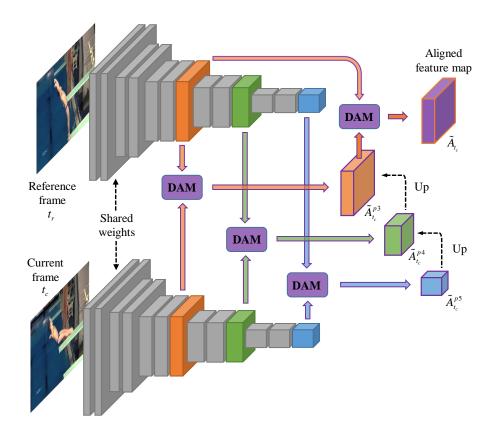


Figure 2: Architecture of MDAN.

bouring frames, their representative capability will be severely affected when the attention targets suffer from severe distortions caused by long-term occlusions or large motions. To address this issue, we further design the Bi-ConvLSTM  $f_{\rm Bi-ConvLSTM}$  to maintain long-term visual attention stability, generating the enhanced spatiotemporal representation as:

$$\{\boldsymbol{H}_{t}^{f}, \boldsymbol{H}_{t}^{b}\} = f_{\text{Bi-ConvLSTM}}(\{\boldsymbol{A}_{t}, \boldsymbol{H}_{t-1}^{f}, \boldsymbol{H}_{t+1}^{b}\}; \boldsymbol{\theta}_{\text{Bi-ConvLSTM}}),$$
 (3)

where  $\boldsymbol{H}_{t}^{f}$  and  $\boldsymbol{H}_{t}^{b}$  denotes the forwardly and backwardly estimated hidden states for frame t, respectively.

Finally, the spatiotemporal representations  $\{\boldsymbol{H}_t^f, \boldsymbol{H}_t^b\}$  in (3) are fused to  $\boldsymbol{Y}_t$  using (11) and fed into the decoder network  $f_{\text{decoder}}$  that is composed of a few convolutional layers and a bilinear upsamping layer, yielding the finally

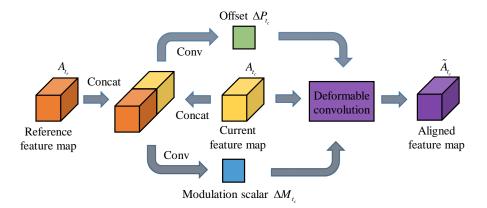


Figure 3: Architecture of DAM.

predicted saliency map of frame t:

$$M_t = f_{\text{decoder}}(Y_t; \boldsymbol{\theta}_{\text{decoder}}),$$
 (4)

where  $\theta_{\text{decoder}}$  is the parameters of the decoder sub-network to be learned.

# 3.2. Multi-scale Deformable Convolutional Alignment Network (MDAN)

Figure 2 illustrates the architecture of MDAN, which progressively makes feature alignment in a coarse-to-fine manner through a set of deformable convolutional alignment modules (DAMs). MDAN can well capture large and complex motion information by adaptively sampling at multiple feature levels in a coarse-to-fine fashion, and does not need to explicitly estimate the motion fields as optical flow [19, 20, 21, 21], thereby greatly reducing computational cost.

Figure 3 shows the architecture of DAM that is based on Dconv [27]. In [27], the Dconv that maps the input feature  $\boldsymbol{A}$  to the output features maps  $\widetilde{\boldsymbol{A}}$  is defined as

$$\widetilde{\boldsymbol{A}}(\boldsymbol{p}) = \sum_{k=1}^{K} \boldsymbol{w}_k \cdot \boldsymbol{A}(\boldsymbol{p} + \boldsymbol{p}_k + \Delta \boldsymbol{p}_k) \cdot \Delta \boldsymbol{m}_k,$$
 (5)

where K denotes the number of sampling locations in a convolutional kernel. For instance, if K = 9,  $p_k \in \{(-1, -1), (-1, 0), \dots, (1, 1)\}$  defines a  $3 \times 3$  convolutional kernel of dilation 1.  $\Delta p_k$  and  $\Delta m_k$  denote the learnable offset and modulation scalar at the k-th location, respectively.  $\mathbf{w}_k$  and  $\mathbf{p}_k$  denote the weight and pre-specified offset of the k-th position, respectively

We employ the Dconv (5) for feature alignment. Given the feature maps  $\mathbf{A}_{t_r}$  and  $\mathbf{A}_{t_c}$  at the reference frame  $t_r$  and the current frame  $t_c$ , respectively, we concatenate the features  $[\mathbf{A}_{t_r}, \mathbf{A}_{t_c}]$  as input, and learn the offset and modulation scalar by

$$\begin{cases}
\Delta \mathbf{P}_{t_c} = f_{\text{offset}}([\mathbf{A}_{t_r}, \mathbf{A}_{t_c}]; \boldsymbol{\theta}_{\text{offset}}), \\
\Delta \mathbf{M}_{t_c} = f_{\text{modulation}}([\mathbf{A}_{t_r}, \mathbf{A}_{t_c}]; \boldsymbol{\theta}_{\text{modulation}}),
\end{cases}$$
(6)

where  $\Delta P = \{\Delta p\}$  and  $\Delta M = \{\Delta m\}$ .  $f_{\text{offset}}$  and  $f_{\text{modulation}}$  are two networks consisting a few convolution layers with parameters  $\boldsymbol{\theta}_{\text{offset}}$  and  $\boldsymbol{\theta}_{\text{modulation}}$ , respectively. Afterwards, we replace  $\boldsymbol{A}$  in (5) by  $\boldsymbol{A}_{t_c}$ , generating the corresponding aligned feature maps  $\boldsymbol{A}_{t_c}$ :

$$\widetilde{\boldsymbol{A}}_{t_c}(\boldsymbol{p}) = \sum_{k=1}^{K} \boldsymbol{w}_k \cdot \boldsymbol{A}_{t_c}(\boldsymbol{p} + \boldsymbol{p}_k + \Delta \boldsymbol{p}_{t_c k}) \cdot \Delta \boldsymbol{m}_{t_c k}, \tag{7}$$

where  $\Delta \mathbf{p}_{t_c} \in \Delta \mathbf{P}_{t_c}$  and  $\Delta \mathbf{m}_{t_c} \in \Delta \mathbf{M}_{t_c}$  in (6).

After introducing the DAM, we give the details of how to design the MDAN. As shown in Figure 2, given the reference frame  $t_r$  and the current frame  $t_c$  as input, we utilize a Siamese network architecture with the VGG16 [38] backbone network to extract their features. Specifically, we first select three different-level feature maps  $\{A_{t_r}^{p3}, A_{t_r}^{p4}, A_{t_r}^{p5}\}$  and  $\{A_{t_c}^{p3}, A_{t_c}^{p4}, A_{t_c}^{p5}\}$  of frames  $t_r$  and  $t_c$ , respectively, which correspond to their pool3, pool4 and pool5 layers in VGG16, respectively. Then, we progressively align the features of frame  $t_c$  to those of frame  $t_r$  in a coarse-to-fine manner. First, we feed the pairs  $\{A_{t_r}^{pi}, A_{t_c}^{pi}\}$ , i=3,4,5 into the formula of DAM (7), outputting the corresponding multi-level aligned features  $\tilde{A}_{t_c}^{pi}$ , i=3,4,5. Then, we fuse the multi-level aligned features progressively from top to bottom, yielding the enhanced aligned features as

$$\boldsymbol{A}_{t_c} = [[\widetilde{\boldsymbol{A}}_{t_c}^{p5} \uparrow, \widetilde{\boldsymbol{A}}_{t_c}^{p4}] \uparrow, \widetilde{\boldsymbol{A}}_{t_c}^{p3}]. \tag{8}$$

Afterwards, we put  $A_{t_c}$  (8) and  $A_{t_r} \leftarrow A_{t_r}^{p3}$  into (7), yielding the output of MDAN  $\widetilde{A}_{t_c}$  for frame  $t_c$ . Finally, we concatenate all the aligned features

 $\{\widetilde{A}_{t_c}\}_{t_c=t-T}^{t+T}$  and fuse them through a  $1\times 1$  convolution layer to yield the enhanced spatiotemporal features for frame t

$$\mathbf{A}_{t} = f_{\text{fusion}}([\widetilde{\mathbf{A}}_{t_{c}=t-T}, \dots, \widetilde{\mathbf{A}}_{t_{c}=t+T}]; \boldsymbol{\theta}_{\text{fusion}}), \tag{9}$$

where  $\theta_{\text{fusion}}$  denotes the weight parameters of the 1 × 1 convolutional layer.

## 3.3. Bidirectional ConvLSTM (Bi-ConvLSTM)

The aforementioned MDAN aligns the features between the reference frame and its left-and right-side neighbouring frames that can be viewed as a short-time bidirectional spatial alignment process. However, video sequence may have large scene transformation and attention shift due to long-term occlusions or large motions, resulting in difficulty by only using short-term information from adjacent frames. We further strengthen the fused features  $A_t$  (9) generated by MDAN through encoding long-term information across more frames, and leverage Bi-ConvLSTM to fully capture long-term spatiotemporal context information in bi-directions. In [39], the ConvLSTM is formulated as:

$$\begin{cases}
\mathbf{I}_{t} = \sigma(\mathbf{W}_{I}^{A} * \mathbf{A}_{t} + \mathbf{W}_{I}^{H} * \mathbf{H}_{t-1}), \\
\mathbf{F}_{t} = \sigma(\mathbf{W}_{F}^{A} * \mathbf{A}_{t} + \mathbf{W}_{F}^{H} * \mathbf{H}_{t-1}), \\
\mathbf{O}_{t} = \sigma(\mathbf{W}_{O}^{A} * \mathbf{A}_{t} + \mathbf{W}_{O}^{H} * \mathbf{H}_{t-1}), \\
\mathbf{C}_{t} = \mathbf{F}_{t} \circ \mathbf{C}_{t-1} + \mathbf{I}_{t} \circ \tanh(\mathbf{W}_{C}^{A} * \mathbf{A}_{t} + \mathbf{W}_{C}^{H} * \mathbf{H}_{t-1}), \\
\mathbf{H}_{t} = \mathbf{O}_{t} \circ \tanh(\mathbf{C}_{t}),
\end{cases}$$
(10)

where \* is the convolution operator,  $\circ$  is the Hadamard product,  $\sigma$  denotes the Sigmoid function, and tanh denotes the hyperbolic tangent function. For different learnable parameters W, we do convolutions with input feature maps  $A_t$  and hidden state  $H_{t-1}$ , respectively, and then sum them and feed into the Sigmoid function to obtain an input gate  $I_t$ , an output gate  $O_t$  and a forget gate  $F_t$ . The memory cell  $C_t$  plays the role of an accumulator of the state information by updating the ratio of memory and forgetting between the current and the previous moments, respectively. Finally, the hidden state  $H_t$  is generated by

pixel-wise multiplying the output gate  $O_t$  by the memory cell  $C_t$  rescaled by a tanh activation function.

The ConvLSTM can capture long-term information from the past frames well, but does not consider the rich information from the future frames that is helpful to further boost the performance of VSP. To this end, we design Bi-ConvLSTM that captures both forward and backward long-range context information, yielding the final spatiotemporal feature representation for VSP:

$$Y_t = H_t^f + H_t^b, \tag{11}$$

where  $\boldsymbol{H}_{t}^{f}$  and  $\boldsymbol{H}_{t}^{b}$  denote the hidden states from forward and backward ConvL-STM units. Afterwards, the output features  $\boldsymbol{Y}_{t}$  are sent to the decoder network  $f_{\text{decoder}}$  (4) to generate the predicted saliency map  $\boldsymbol{M}_{t}$ .

#### 3.4. Loss Function

We leverage the loss function similar to that proposed by [10, 21], which combines four loss terms related to saliency evaluation metrics. The loss function is formulated as:

$$\mathcal{L}(P, Q, G) = \mathcal{L}_{NSS}(P, Q) + \mathcal{L}_{SIM}(P, G) + \mathcal{L}_{CC}(P, G) + \mathcal{L}_{KL}(P, G), \quad (12)$$

where P denotes the predicted attention map, Q is the ground-truth binary fixation map, and G indicates the continuous ground-truth attention map.

 $\mathcal{L}_{\mathrm{NSS}}$  originates from normalized scanpath saliency (NSS), which is introduced to the visual saliency field as a simple correspondence measure between saliency maps and ground-truth [40].  $\mathcal{L}_{\mathrm{NSS}}$  computes the average normalized saliency at fixated locations:

$$\mathcal{L}_{\text{NSS}}(\boldsymbol{P}, \boldsymbol{Q}) = -\frac{1}{N} \sum_{i} \frac{\boldsymbol{P} - \mu(\boldsymbol{P})}{\sigma(\boldsymbol{P})} \cdot \boldsymbol{Q}_{i}, \tag{13}$$

where  $\mu(\cdot)$  and  $\sigma(\cdot)$  denote the mean and the standard deviation, respectively, N is the number of positive pixels belonging to Q.

 $\mathcal{L}_{\text{SIM}}$  is derived from the similarity (SIM) metric, which measures the similarity between two distributions and computes the sum of the minimum values

at each pixel:

$$\mathcal{L}_{SIM}(\boldsymbol{P}, \boldsymbol{G}) = -\sum_{i} \min(\boldsymbol{P}_{i}, \boldsymbol{G}_{i}), \tag{14}$$

where P and Q are normalized to  $\sum_{i} P_{i} = \sum_{i} G_{i} = 1$ .  $\mathcal{L}_{\text{CC}}$  measures the correlation or dependence of two variables by linear correlation coefficient (CC):

$$\mathcal{L}_{CC}(\mathbf{P}, \mathbf{G}) = -\frac{cov(\mathbf{P}, \mathbf{G})}{\sigma(\mathbf{P}) \times \sigma(\mathbf{G})},$$
(15)

where cov() means the covariance and  $\sigma(\cdot)$  is the standard deviation.

 $\mathcal{L}_{\mathrm{KL}}$  is from Kullback-Leibler (KL) divergence metric, which measures the difference between two probability distributions:

$$\mathcal{L}_{KL}(\boldsymbol{P}, \boldsymbol{G}) = \sum_{i} \boldsymbol{G}_{i} \log(\frac{\boldsymbol{G}_{i}}{\boldsymbol{P}_{i}}). \tag{16}$$

# 4. Experiments

## 4.1. Implementation Details

We adopt the frequently-used VGG16 [38] pre-trained on ImageNet [42] as the backbone network to extract three feature maps with different resolutions. All the other parameters are trained from scratch except for the backbone network. The neighbor size of the reference frame is set to 2T + 1 = 5. Each video training batch contains 15 consecutive frames from the same video with batch size 4. We randomly select the video and its starting frame for each training sample. All the training frames are scaled to  $256 \times 320$  pixels, and the surrounding pixels of the image are padded with 0 if its size does not match to  $256 \times 320$  pixels. The ground-truth binary fixation mask and its continuous attention map are scaled to  $128 \times 160$  pixels. We use the Adam optimizer [43] to learn the whole network parameters  $\theta_{\text{CNN}}$  in an end-to-end manner without any post processing. The learning rate is set to 1e-7 and the model converges at about 20,000 steps. The proposed model is implemented in PyTorch and one Nvidia RTX 2080Ti GPU is used for acceleration. The whole training process takes about 15 hours.

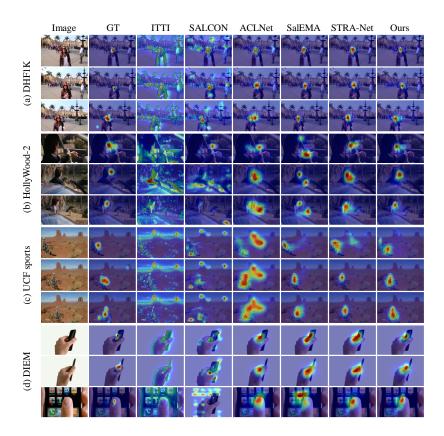


Figure 4: Qualitative comparisons with other methods including ITTI [41], SALICON [10], ACLNet [22], SalEMA [23], STRA-Net [21] on (a) DHF1K, (b) HollyWood-2, (c) UCF sports, (d) DIEM. We show one example video with three frames for demonstration purpose on each dataset.

# 4.2. Evaluation Datasets

We conduct extensive evaluations on four widely-used eye-tracking benchmark datasets.

**DHF1K** [53]: It consists of 1,000 high-quality elaborately-selected video sequences that are annotated by 17 observers using an eye tracker device. The videos therein have diverse contents, varied motion patterns, various objects, large scale and high quality. The videos are divided into a training set of 600 videos and a validation set of 100 videos that are publicly available, but the fixation labels of the remaining 300 videos are not released, which are used to

Table 1: Quantitative results on DHF1K, HollyWood-2 and UCF sports dataset. The best-performing scores are marked in **bold**, and the second-best performing scores are underlined. \* represents the DL-based model.

	Dataset	DHF1K				HollyWood-2				UCF sports						
	Methods	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
Static Models	ITTI [41]	0.774	0.162	0.553	0.233	1.207	0.788	0.221	0.607	0.257	1.076	0.847	0.251	0.725	0.356	1.640
	GBVS [44]	0.828	0.186	0.554	0.283	1.474	0.837	0.257	0.633	0.308	1.336	0.859	0.274	0.697	0.396	1.818
	*SALICON [10]	0.857	0.232	0.590	0.327	1.901	0.856	0.321	0.711	0.425	2.013	0.848	0.304	0.738	0.375	1.838
	*Shallow-Net [12]	0.833	0.182	0.529	0.295	1.509	0.851	0.276	0.694	0.423	1.680	0.846	0.276	0.691	0.382	1.789
	*Deep-Net [12]	0.855	0.201	0.592	0.331	1.775	0.884	0.300	0.736	0.451	2.066	0.861	0.282	0.719	0.414	1.903
	*DVA [14]	0.860	0.262	0.595	0.358	2.013	0.886	0.372	0.727	0.482	2.459	0.872	0.339	0.725	0.439	2.311
Dynamic Models	PQFT [45]	0.699	0.139	0.562	0.137	0.749	0.723	0.201	0.621	0.153	0.755	0.825	0.250	0.722	0.338	1.780
	Seo et al. [46]	0.635	0.142	0.499	0.070	0.334	0.652	0.155	0.530	0.076	0.346	0.831	0.308	0.666	0.336	1.690
	Rudoy et al. [47]	0.769	0.214	0.501	0.285	1.498	0.783	0.315	0.536	0.302	1.570	0.763	0.271	0.637	0.344	1.619
	Hou et al. [48]	0.726	0.167	0.545	0.150	0.847	0.731	0.202	0.580	0.146	0.684	0.819	0.276	0.674	0.292	1.399
	Fang et al. [49]	0.819	0.198	0.537	0.273	1.539	0.859	0.272	0.659	0.358	1.667	0.845	0.307	0.674	0.395	1.787
	OBDL [50]	0.638	0.171	0.500	0.117	0.495	0.640	0.170	0.541	0.106	0.462	0.759	0.193	0.634	0.234	1.382
mic	AWS-D [51]	0.703	0.157	0.513	0.174	0.940	0.694	0.175	0.637	0.146	0.742	0.823	0.228	0.750	0.306	1.631
yna	*OM-CNN [52]	0.856	0.256	0.583	0.344	1.911	0.887	0.356	0.693	0.446	2.313	0.870	0.321	0.691	0.405	2.089
Д	*Two-stream [19]	0.834	0.197	0.581	0.325	1.632	0.863	0.276	0.710	0.382	1.748	0.832	0.264	0.685	0.343	1.753
	*ACLNet [22]	0.890	0.315	0.601	0.434	2.354	0.913	0.542	0.757	0.623	3.086	0.905	0.496	0.767	0.603	3.200
	*SalEMA [23]	0.890	0.465	0.667	0.449	2.573	0.919	0.487	0.708	0.613	3.186	0.906	0.431	0.740	0.544	2.638
	*TASED-Net [33]	0.895	0.361	0.712	0.470	2.667	0.918	0.507	0.768	0.646	3.302	0.899	0.469	0.752	0.582	2.920
	*STRA-Net [21]	0.895	0.355	0.663	0.458	2.558	0.923	0.536	0.774	0.662	3.478	0.914	0.535	0.790	0.645	3.472
Ours	Training setting (i)	0.896	0.390	0.679	0.479	2.758	0.915	0.504	0.786	0.613	3.461	0.896	0.455	0.760	0.558	2.985
	Training setting (ii)	0.888	0.309	0.670	0.438	2.479	0.934	0.529	0.806	0.672	3.936	0.913	0.418	0.753	0.566	3.039
	Training setting (iii)	0.851	0.260	0.664	0.327	1.876	0.893	0.402	0.752	0.481	2.627	0.921	0.497	0.799	0.612	3.676
	Training setting (iv)	0.900	0.353	0.680	0.476	2.685	0.928	0.537	0.800	0.661	3.804	0.917	0.494	0.785	0.599	3.406

validate the generalization capability of the model fairly.

HollyWood-2 [28]: It contains 1,707 videos selected from the HollyWood-2 action recognition dataset [54] that is collected from a set of 69 HollyWood movies. 12 action classes are included such as *hugging*, *kissing*, *running*, *etc*. The whole data consists of a training set of 823 sequences and a test set of 884 sequences. It is one of the largest and most challenging available dataset for VSP.

**UCF Sports** [28]: It includes 150 videos from the UCF sports action datasets [55], which covers 9 sports action classes such as *swinging*, *lifting*, *skate-boarding*, *etc*. The dataset consists of 103 videos for training and 47 videos for testing.

**DIEM** [29]: It has 84 videos that are collected from 50 participants which are widely used for studying human-eye fixation attention. Following [50, 21], we select the same 20 videos as the testing set.

For fair comparison, we leverage the standard training strategy in [22],

Table 2: Quantitative results on DIEM dataset (without training). The best-performing scores are marked in **bold**. The second best-performing scores are underlined. \* represents the DL-based model.

	Methods	AUC-J↑	SIM↑	s-AUC↑	CC↑	NSS↑
Models	ITTI [41]	0.791	0.132	0.653	0.196	1.103
	GBVS [44]	0.813	0.156	0.633	0.214	1.198
	*SALICON [10]	0.793	0.171	0.674	0.270	1.650
ic ]	*Shallow-Net [12]	0.838	0.188	0.620	0.297	1.646
Static	*Deep-Net [12]	0.849	0.164	0.697	0.291	1.650
	*DVA [14]	0.868	0.237	0.721	0.386	2.347
Dynamic Models	PQFT [45]	0.724	0.126	0.649	0.144	0.856
	Seo et al. [46]	0.724	0.120	0.568	0.116	0.665
	Rudoy et al. [47]	0.725	0.150	0.618	0.260	1.390
	Hou et al. [48]	0.735	0.130 $0.142$	0.589	0.128	0.735
	Fang et al. [49]	0.823	0.142	0.636	0.120 $0.251$	1.423
	OBDL [50]	0.762	0.165	0.694	0.221	1.289
	AWS-D [51]	0.702 $0.774$	0.150	0.695	0.221	1.252
	*OM-CNN [52]	0.857	0.130	0.693	0.210 $0.371$	$\frac{1.232}{2.235}$
	*Two-stream [19]	0.859	0.256 $0.256$	0.682	0.366	2.235 $2.171$
	*ACLNet [22]	0.881	0.230 $0.277$	0.693	0.396	2.368
	*STRA-Net [21]	0.870	0.306	0.678	0.390	2.300 $2.452$
Ours	Training setting (i)	0.880	0.391	0.718	$\frac{0.472}{0.433}$	$\frac{2.346}{2.376}$
	Training setting (ii)	0.870	0.355	0.698	0.468	2.359
	Training setting (iii)	0.831	0.296	0.676	0.351	1.828
	Training setting (iv)	0.889	0.396	0.711	0.490	2.346

which consists of 4 training settings with the training sets of (i) DHF1K, (ii) HollyWood-2, (iii) UCF sports, (iv) DHF1K + HollyWood-2 + UCF sports. Meanwhile, we use the testing sets of DHF1K, HollyWood-2 and UCF sports to evaluate the performance. Furthermore, to further assess the generalization capability of our model, we evaluate it on the testing set of DIEM which has no training set available.

# 4.3. Comparison Results

As [22, 23, 33, 21], we use the widely-used evaluation metrics [56] to evaluate the comparative methods, including Normalized Scanpath Saliency (NSS), Similarity (SIM), shuffled AUC (s-AUC), linear Correlation Coefficient (CC) and

# AUC-J.

We compare the proposed approach with 19 saliency models, including 6 static models (ITTI [41], GBVS [44], SALICON [10], Shallow-Net [12], Deep-Net [12], DVA [14]) and 13 dynamic models (PQFT [45], Seo et al. [46], Rudoy et al. [47], Hou et al. [48], Fang et al. [49], OBDL [50], AWS-D [51], OM-CNN [52], Two-stream [19], ACLNet [22], SalEMA [23], TASED-Net [33], STRA-Net [21]). Among them, SALICON [10], Shallow-Net [12], Deep-Net [12], DVA [14], OM-CNN [52], Two-stream [19], ACLNet [22] and STRA-Net [21] are the DL-based models while the others are the traditional models.

Results on DHF1K. We test our model on the testing set of DHF1K, which contains 300 videos without publicly released ground-truth annotations of human eye-tracking maps available. A public server is used to report the results on the test set. It is a fair and large test set for verifying the generalization capability of our model. Table 1 lists the results of our model in terms of AUC-J, SIM, s-AUC, CC and NSS. Among them, our model achieves the best CC and NSS scores and the second-best SIM and AUC-J scores with training setting (i). Meanwhile, for training setting (iv), our model achieves the best AUC-J score and the second-best s-AUC, CC and NSS scores. Besides, the results of our model with all the settings are much better than the statistic models. For the settings (ii) and (iii), our model achieves competing performance among the dynamic models, following the top-performing methods including ACLNet, SalEMA, TASED-Net and STRA-Net.

Figure 4 (a) shows the qualitative comparison results, where the visual attention maps generated by our model well approach to the ground-truth, which more focus on the saliency targets and are not disturbed by the transitions of the backgrounds.

Results on HollyWood-2. As listed by Table 1, the performance of our approach is superior to the other methods, especially in training setting (ii), our model achieves the best scores in terms of s-AUC, CC and NSS. The dataset mainly

focuses on task-driven viewing mode whose contents are limited to human actions and movie scenes. Therefore, training with the same distribution of the training setting yields much better results. Figure 4 (b) shows a man opening the door from the car and going out, where our predicted saliency maps can more accurately track the salient objects than the saliency maps generated by the other methods.

Results on UCF sports. Compared to all the other models, the proposed method achieves the best or second-best performance in terms of all metrics under training setting (iii). For the other training settings, our method achieves comparative results compared to the top-performing methods such as ACLNet, SalEMA, TASED-Net and STRA-Net. Figure 4 (c) shows the scene where a man rides a horse in the desert. The salient targets we focus on at different times will change, and our predicted visual maps enable to precisely move the attention shift from horse to man compared to the other methods.

Results on DIEM. For evaluating the generalization capability of our model, we do not use any data of DIEM to train our model. Following [50, 21], we evaluate our model on the testing set of DIEM, containing the first 300 frames of each video. Table 1 lists the quantitative results, which achieves competitive results compared to the other methods, especially under training settings (i) and (iv) where the proposed method achieves the best or second-best performance in terms of almost all metrics.

# 4.4. Ablative Study

To further show our main contributions, we compare different variants of our model including those without Bi-ConvLSTM and replacing all Dconv in MDAN to regular convolution. Table 3 lists the results of ablative experiments on the UCF sports testing set under training setting (iii). We can observe that without Bi-ConvLSTM, the scores of all the metrics decline to some degree such as the AUC-J score drops from 0.921 to 0.9034 and the CC score reduces by

Table 3: Results of ablative experiments of the proposed model on the UCF sports testing set. The best scores are marked in **bold**. The symbol w/o denotes removing a specific module.

UCF sports Dataset	AUC-J↑	SIM↑	s-AUC↑	$CC\uparrow$	NSS↑
Ours	0.921	0.497	0.799	0.612	3.676
w/o. Bi-ConvLSTM	0.903	0.470	0.788	0.571	3.467
w/o. forward ConvLSTM	0.907	0.467	0.778	0.590	3.293
w/o. backward ConvLSTM	0.894	0.457	0.789	0.536	3.393
replace DConv to Conv in MDAN	0.905	0.452	0.775	0.549	3.265
w/o. p5 in MDAN	0.894	0.442	0.781	0.565	3.483
w/o. $p4$ and $p5$ in MDAN	0.895	0.425	0.776	0.548	3.168
w/o. p3 in MDAN	0.900	0.451	0.783	0.543	3.236
w/o. $p3$ and $p4$ in MDAN	0.874	0.423	0.780	0.516	3.357

0.04 from 0.612 to 0.571, verifying the effectiveness of Bi-ConvLSTM. Moreover, replacing Dconv to regular convolution in MDAN causes the performance degradation. For example, the SIM score drops from 0.497 to 0.452 and the NSS score decreases from 3.676 to 3.265, demonstrating the effectiveness of using DConv for feature alignment across frames. Finally, to make further efforts to verify the effectiveness of other modules, we conduct some ablative experiments including: removing forward ConvLSTM, backward ConvLSTM, p3, p4, p5 in MDAN, respectively. The extra results are also listed in the bottom row of Table 3, which confidently validate that each component in Bi-ConvLSTM and MDAN has a positive effect to boost the performance of the proposed approach.

# 5. Conclusions

In this paper, we have presented an effective enhanced spatiotemporal alignment network for VSP, mainly including two novel module designs: the MDAN and the Bi-ConvLSTM. The MDAN makes multi-resolution feature alignment between the reference and its neighboring frames in a coarse-to-fine manner, which is good at handling various motions. Afterwards, the output features of MDAN are further enhanced by the Bi-ConvLSTM that fully captures the long-time temporal information in both forward and backward timing directions. Extensive experiments on four VSP benchmark datasets have clearly

demonstrated superiority of the proposed method to state-of-the-art methods in terms of five metrics.

# References

- [1] T. Deng, K. Yang, Y. Li, H. Yan, Where does the driver look? top-down-based saliency detection in a traffic driving environment, IEEE Transactions on Intelligent Transportation Systems 17 (2016) 2051–2062.
- [2] W. G. Aguilar, M. A. Luna, J. F. Moya, V. Abad, H. Ruiz, H. Parra, C. Angulo, Pedestrian detection for uavs using cascade classifiers and saliency maps, in: International Work-Conference on Artificial Neural Networks, Springer, 2017, pp. 563–574.
- [3] C. F. Flores, A. Gonzalez-Garcia, J. van de Weijer, B. Raducanu, Saliency for fine-grained object recognition in domains with scarce training data, Pattern Recognition 94 (2019) 62–73.
- [4] W. Wang, J. Shen, R. Yang, F. Porikli, Saliency-aware video object segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (2017) 20–33.
- [5] Y.-T. Hu, J.-B. Huang, A. G. Schwing, Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation, in: European Conference on Computer Vision, 2018, pp. 786–802.
- [6] P. Zhang, W. Liu, D. Wang, Y. Lei, H. Wang, C. Shen, H. Lu, Non-rigid object tracking via deep multi-scale spatial-temporal discriminative saliency maps, Pattern Recognition (2019) 107130.
- [7] L. Gao, Z. Guo, H. Zhang, X. Xu, H. T. Shen, Video captioning with attention-based lstm and semantic consistency, IEEE Transactions on Multimedia 19 (2017) 2045–2055.

- [8] G. Schillaci, S. Bodiroža, V. V. Hafner, Evaluating the effect of saliency detection and attention manipulation in human-robot interaction, International Journal of Social Robotics 5 (2013) 139–152.
- [9] S. Marat, M. Guironnet, D. Pellerin, Video summarization using a visual attention model, in: European Signal Processing Conference, IEEE, 2007, pp. 1784–1788.
- [10] X. Huang, C. Shen, X. Boix, Q. Zhao, Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks, in: International Conference on Computer Vision, 2015, pp. 262–270.
- [11] N. Liu, J. Han, D. Zhang, S. Wen, T. Liu, Predicting eye fixations using convolutional neural networks, in: Computer Vision and Pattern Recognition, 2015, pp. 362–370.
- [12] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, N. E. O'Connor, Shallow and deep convolutional networks for saliency prediction, in: Computer Vision and Pattern Recognition, 2016, pp. 598–606.
- [13] S. S. Kruthiventi, K. Ayush, R. V. Babu, Deepfix: A fully convolutional neural network for predicting human eye fixations, IEEE Transactions on Image Processing 26 (2017) 4446–4456.
- [14] W. Wang, J. Shen, Deep visual attention prediction, IEEE Transactions on Image Processing 27 (2017) 2368–2378.
- [15] L. Itti, N. Dhavale, F. Pighin, 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, volume 5200, International Society for Optics and Photonics, 2003, pp. 64–78.
- [16] L. Zhang, M. H. Tong, G. W. Cottrell, Sunday: Saliency using natural statistics for dynamic analysis of scenes, in: Annual Cognitive Science Conference, AAAI Press Cambridge, MA, 2009, pp. 2944–2949.

- [17] Z. Ren, S. Gao, L.-T. Chia, D. Rajan, Regularized feature reconstruction for spatio-temporal saliency detection, IEEE Transactions on Image Processing 22 (2013) 3120–3132.
- [18] Y. Yan, J. Ren, G. Sun, H. Zhao, J. Han, X. Li, S. Marshall, J. Zhan, Unsupervised image saliency detection with gestalt-laws guided optimization and visual attention based refinement, Pattern Recognition 79 (2018) 65–78.
- [19] C. Bak, A. Kocak, E. Erdem, A. Erdem, Spatio-temporal saliency networks for dynamic saliency prediction, IEEE Transactions on Multimedia 20 (2017) 1688–1698.
- [20] L. Jiang, M. Xu, Z. Wang, Predicting video saliency with object-to-motion cnn and two-layer convolutional lstm, arXiv preprint arXiv:1709.06316 (2017).
- [21] Q. Lai, W. Wang, H. Sun, J. Shen, Video saliency prediction using spatiotemporal residual attentive networks, IEEE Transactions on Image Processing 29 (2019) 1113–1126.
- [22] W. Wang, J. Shen, F. Guo, M.-M. Cheng, A. Borji, Revisiting video saliency: A large-scale benchmark and a new model, in: Computer Vision and Pattern Recognition, 2018, pp. 4894–4903.
- [23] P. Linardos, E. Mohedano, J. J. Nieto, N. E. O'Connor, X. Giro-i Nieto, K. McGuinness, Simple vs complex temporal recurrences for video saliency prediction, arXiv preprint arXiv:1907.01869 (2019).
- [24] Y. Jo, S. Wug Oh, J. Kang, S. Joo Kim, Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation, in: Computer Vision and Pattern Recognition, 2018, pp. 3224–3232.
- [25] Y. Tian, Y. Zhang, Y. Fu, C. Xu, Tdan: Temporally deformable alignment network for video super-resolution, arXiv preprint arXiv:1812.02898 (2018).

- [26] X. Wang, K. C. Chan, K. Yu, C. Dong, C. Change Loy, Edvr: Video restoration with enhanced deformable convolutional networks, in: Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.
- [27] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: Computer Vision and Pattern Recognition, 2019, pp. 9308–9316.
- [28] S. Mathe, C. Sminchisescu, Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2014) 1408–1424.
- [29] P. K. Mital, T. J. Smith, R. L. Hill, J. M. Henderson, Clustering of gaze during dynamic scene viewing is predicted by motion, Cognitive Computation 3 (2011) 5–24.
- [30] E. Vig, M. Dorr, D. Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, in: Computer Vision and Pattern Recognition, 2014, pp. 2798–2805.
- [31] N. Liu, J. Han, T. Liu, X. Li, Learning to predict eye fixations via multiresolution convolutional neural networks, IEEE Transactions on Neural Networks and Learning Systems 29 (2016) 392–404.
- [32] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, X. Giro-i Nieto, Salgan: Visual saliency prediction with generative adversarial networks, arXiv preprint arXiv:1701.01081 (2017).
- [33] K. Min, J. J. Corso, Tased-net: Temporally-aggregating spatial encoderdecoder network for video saliency detection, in: International Conference on Computer Vision, 2019, pp. 2394–2403.
- [34] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: International Conference on Computer Vision, 2017, pp. 764–773.

- [35] Z. Yang, S. Liu, H. Hu, L. Wang, S. Lin, Reppoints: Point set representation for object detection, in: International Conference on Computer Vision, 2019.
- [36] J. Zhu, L. Fang, P. Ghamisi, Deformable convolutional neural networks for hyperspectral image classification, IEEE Geoscience and Remote Sensing Letters 15 (2018) 1254–1258.
- [37] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, H. Wu, Addrowdnet: An attention-injective deformable convolutional network for crowd understanding, in: Computer Vision and Pattern Recognition, 2019, pp. 3225–3234.
- [38] K. Simonyan, A. Zisserman, Very deep convolutional networks for largescale image recognition, arXiv preprint arXiv:1409.1556 (2014).
- [39] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, in: Advances in Neural Information Processing Systems, 2015, pp. 802–810.
- [40] R. J. Peters, A. Iyer, L. Itti, C. Koch, Components of bottom-up gaze allocation in natural images, Vision Research 45 (2005) 2397–2416.
- [41] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence (1998) 1254–1259.
- [42] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
- [43] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
- [44] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Advances in Neural Information Processing Systems, 2007, pp. 545–552.

- [45] C. Guo, L. Zhang, A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression, IEEE Transactions on Image Processing 19 (2009) 185–198.
- [46] H. J. Seo, P. Milanfar, Static and space-time visual saliency detection by self-resemblance, Journal of vision 9 (2009) 15–15.
- [47] D. Rudoy, D. B. Goldman, E. Shechtman, L. Zelnik-Manor, Learning video saliency from human gaze using candidate selection, in: Computer Vision and Pattern Recognition, 2013, pp. 1147–1154.
- [48] X. Hou, L. Zhang, Dynamic visual attention: Searching for coding length increments, in: Advances in Neural Information Processing Systems, 2009, pp. 681–688.
- [49] Y. Fang, Z. Wang, W. Lin, Z. Fang, Video saliency incorporating spatiotemporal cues and uncertainty weighting, IEEE Transactions on Image Processing 23 (2014) 3910–3921.
- [50] S. Hossein Khatoonabadi, N. Vasconcelos, I. V. Bajic, Y. Shan, How many bits does it take for a stimulus to be salient?, in: Computer Vision and Pattern Recognition, 2015, pp. 5501–5510.
- [51] V. Leboran, A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, Dynamic whitening saliency, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2016) 893–907.
- [52] L. Jiang, M. Xu, T. Liu, M. Qiao, Z. Wang, Deepvs: A deep learning based video saliency prediction approach, in: European Conference on Computer Vision, 2018, pp. 602–617.
- [53] W. Wang, J. Shen, J. Xie, M.-M. Cheng, H. Ling, A. Borji, Revisiting video saliency prediction in the deep learning era, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019).

- [54] M. Marszałek, I. Laptev, C. Schmid, Actions in context, in: Computer Vision and Pattern Recognition, IEEE Computer Society, 2009, pp. 2929– 2936.
- [55] M. D. Rodriguez, J. Ahmed, M. Shah, Action mach a spatio-temporal maximum average correlation height filter for action recognition., in: Computer Vision and Pattern Recognition, volume 1, 2008, p. 6.
- [56] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, F. Durand, What do different evaluation metrics tell us about saliency models?, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (2018) 740–757.