A Plug-and-play Scheme to Adapt Image Saliency Deep Model for Video Data

Yunxiao Li¹ Shuai Li¹ Chenglizhao Chen¹,²∗ Aimin Hao¹,³ Hong Qin⁴
¹State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
²Qingdao University ³Peng Cheng Laboratory ⁴Stony Brook University

Code&Data: https://github.com/YunX17/AdaptSaliency

Abstract—With the rapid development of deep learning techniques, image saliency deep models trained solely by spatial information have occasionally achieved detection performance for video data comparable to that of the models trained by both spatial and temporal information. However, due to the lesser consideration of temporal information, the image saliency deep models may become fragile in the video sequences dominated by temporal information. Thus, the most recent video saliency detection approaches have adopted the network architecture starting with a spatial deep model that is followed by an elaborately designed temporal deep model. However, such methods easily encounter the performance bottleneck arising from the single stream learning methodology, so the overall detection performance is largely determined by the spatial deep model. In sharp contrast to the current mainstream methods, this paper proposes a novel plug-and-play scheme to weakly retrain a pretrained image saliency deep model for video data by using the newly sensed and coded temporal information. Thus, the retrained image saliency deep model will be able to maintain temporal saliency awareness, achieving much improved detection performance. Moreover, our method is simple yet effective for adapting any off-the-shelf pre-trained image saliency deep model to obtain high-quality video saliency detection. Additionally, both the data and source code of our method are publicly available.

Index Terms—Video Saliency Detection, Weakly Supervised Learning.

I. INTRODUCTION AND MOTIVATION

I

MAGE saliency detection aims at locating the most eye-catching regions in a given scene [1, 2]. As a pre-processing tool, its subsequent applications frequently include various computer vision applications, e.g., adaptive image retargeting [3], image compression [4], object tracking [5], and video surveillance [6, 7].

Since the human visual system is extremely sensitive to the distinct movement patterns [8, 9, 10], the solely spatial-information-trained image saliency deep models may become fragile in video data whose saliency should be simultaneously determined by both spatial and temporal information [11]. Thus, we may suppose that the current video saliency methods [12, 13, 14] that make full use of both the spatial and temporal information should significantly outperform the image saliency deep models [15, 16, 17, 18]. However, more often than not, the opposite result is obtained due to the variable nature of video data. For example, a salient object may occasionally be static for a long period showing no movement [19] so that the spatial information may become the only saliency cue and, as a result, the spatiotemporal video saliency models cannot perform well in such case. Thus, the overall performances of the up-to-date image saliency models [20, 21, 22, 23] are occasionally comparable to that of the state-of-the-art video saliency models.

In fact, there are two types of mainstream network architectures that are prevalent in the video saliency detection field, i.e., the early bi-stream network architecture (Fig. 1-B) and the current single-stream network architecture (Fig. 1-A). The conventional bi-stream network architecture consists of two independent sub-branches, of which one aims to conduct color saliency estimation from the spatial domain and the other attempts to reveal motion saliency clues over the temporal scale. Then, a fusion module is latterly applied to achieve a complementary status between the deep features of its precedent sub-branch, achieving spatial-temporal video saliency detection. However, the overall performance of the methods based on the bi-stream architecture is strongly limited by the temporal sub-branch because it is much more difficult to conduct temporal saliency estimation directly from the video frames with an extremely large problem domain.

As a result, the most recent video saliency work [24] has adopted the single stream network architecture (Fig. 1-A) that performs the spatial-temporal saliency estimation within the coarse-to-fine manner; i.e., the preceding color saliency deep model aims to coarsely locate the salient regions in the spatial domain, while its subsequent motion saliency deep model attempts to finely filter the non-salient nearby surroundings over the temporal perspective. Because the motion saliency deep model takes the output of its preceding color saliency deep model as the input (i.e., the spatial saliency deep features), its temporal saliency learning procedure can easily converge within a much simple problem domain. Nevertheless, the single stream network architecture also encounters a chicken-and-egg problem resulting in a performance bottleneck; i.e., the performance of the motion saliency deep model is heavily dependent on its preceding color deep saliency model, however, the deep features provided by color deep model generally show limited performance due to its lesser consideration of the temporal information.

To address the above-mentioned problems, in this paper, we attempt to follow the conventional bi-stream network architecture and devise a novel weakly supervised learning scheme to break the performance bottleneck of the temporal...
A. The current main stream stage-wise deep models

B. The conventional bi-stream deep models

C. Our novel method with Weakly Supervised Learning Module

Fig. 1. Differences between our novel method and the conventional methods.

sub-branch. We have shown the overall methodology of our method in Fig. 1-C, and the key ideas of our method include the following 2 aspects:

1) We have devised a novel supervised fine-tune scheme that uses tiny amounts of newly sensed and coded temporal information to rapidly adapt any off-the-shelf color saliency deep model for a high-quality temporal saliency estimation;
2) Based on the saliency detection results of the bi-stream network, we attempt to rapidly identify video frames with high quality video saliency detections that will be subsequently used as the pseudo ground truth for the weakly supervised learning, enabling the color sub-branch to perform saliency estimation over both spatial and temporal domains.

In particular, it is important to mention that the performance of our method can be easily improved further when we adopt a much stronger pre-trained color saliency deep model as the color sub-branch. Thus, with the rapid development of the color saliency deep learning techniques, our method will eventually be able to outperform the conventional video saliency detection methods.

II. BACKGROUND AND RELATED WORKS

A. Image Saliency Detection Methods

The conventional image saliency detection methods frequently adopt multiple discriminative hand-crafted saliency cues to obtain a robust detection result, e.g., the most representative regional contrast saliency cue [25] and the multi-level hierarchical saliency cue [26].

After entering the deep learning era, the classic deep learning techniques, e.g., convolutional neural networks (CNNs) and full convolutional networks (FCNs), have been widely applied for image saliency detection. Since the automatically computed deep features have much more discriminative semantic information, the deep learning based image saliency detection methods can significantly outperform the conventional hand-crafted methods. Li et al. [18] propose to simultaneously make full use of both the CNN-based deep features and the hand-crafted low-level features, achieving much improved detection performance. Furthermore, Li et al. [24] also propose a hybrid contrast-oriented deep neural networks method that takes full advantage of an attentional module to alleviate its computational burden.

The multi-level feature aggregation scheme has also been widely adopted in the image saliency detection field. Hou et al. [15] resort to the short connections to combine both deeper layers and shallower deep features in FCNs, which promotes the saliency detection accuracy effectively. Although the method [15] has achieved significant performance, it suffers from heavy computational burden. Hence, Chen et al. [16] propose the residual learning scheme to further refine the side layers’ deep features, shrinking the parameters of its convolutional layers effectively. Hu et al. [17] equip their network with recurrently aggregated deep features captured in the different layers of Fully Convolutional Networks (FCNs) to more accurately detect salient objects.

In [22], Wu et al. point out the wrong recognition toward the multi-level learning scheme; i.e., shallow deep features contribute less to the overall detection performance but have a higher computational cost. Therefore, [22] propose to directly ignore these deep layers from the shallow layers to further increase the computational speed.

Most recently, Liu et al. [23] propose a global guidance module from the bottom-up pathway and a feature aggregation module from the top-down pathway, fusing them in a multi-scale manner to enrich the details of the saliency detection. Wang et al. [20] propose a pyramid attention structure for salient object detection, enhancing the representation ability of its deep features effectively. Moreover, Feng et al. [21] further utilize an attentive feedback network with boundary-enhanced loss to further sharpen the detected salient object boundaries.

B. Hand-crafted Video Saliency Detection Methods

The conventional hand-crafted video saliency detection methods [28], [29], [1], [30], [31] usually utilize low-level cues to construct an elaborately designed optimization graph for maintaining saliency spatiotemporal coherency. Wang et al. [1] propose a spatial-temporal energy function with a gradient flow field to obtain spatiotemporally consistent saliency maps that are then further improved by using the newly designed appearance model and location model [32]. Similarly, by using multiple low-level features, Kim et al. [33] construct a probability framework consisting of spatial transition matrices with temporal restarting distribution in order to evaluate the video saliency via random walk with restart. Xi et al. [30] propose to feed multiple newly revealed spatiotemporal background priors into a dual-graph network, achieving much improved detection performance.

Unlike for the above-mentioned graph optimization based methods, the spatiotemporal coherency can also be sustained within a batch-wise manner. Li et al. [34] propose a kernel regression that includes three entity-models to exploit the spatiotemporal coherency of attentional regions. Chen
et al. propose to conduct low-rank guided batch-wise regional alignments, fulfilling the spatiotemporal coherency by constraining the saliency similarity between any aligned regional pairs. Further, Chen et al. introduce a bi-level feature learning scheme to further expand the spatiotemporal coherency sensing scope from the long-term perspective. Liu et al. perform temporal and spatial propagation via similarity matrices, obtaining saliency maps with strong spatiotemporal coherent. Zhou et al. employ localized estimation models with spatiotemporal refinement mechanism to further improve the detection performance. Guo et al. propose to integrate the conventional motion saliency cues into the object proposals. The temporal saliency cues are much more stable than the spatial saliency cues. Therefore, Guo et al. develop a rapid video saliency detection method using the principal motion vector and an appearance cue to achieve temporal consistency.

C. Video Saliency Deep Models

As previously mentioned, almost all of the deep learning based video saliency object detection methods can be divided into two types, namely, the bi-stream and the single stream methods. The most representative bi-stream method proposes a two-stream network, feeding static saliency into the module for dynamic saliency, obtaining saliency with a lower computational load. Addressing the same issue, Li et al. design a universal framework to increase the temporal coherence of the deep feature representation with ConvLSTM, achieving high speed. After attaching a fast-moving object edge map to the bi-stream based network, Sun et al. combine memory information to achieve robust detection. To deal with the lack of manually labeled data, Tang et al. train two cascade fully convolutional networks in a weakly supervised manner to predict saliency via both spatial and temporal cues. Le et al. propose to detect salient foregrounds by using conventional convolution in spatial branch and 3D convolution in the temporal branch over regions and consecutive frames, respectively. Similar to the method in, Fang et al. apply STSM and SSAM to estimate saliency over the time axis and spatial coordinate, respectively. And Wen et al. generate saliency via fusing multi-level deep features extracted by a symmetrical CNN composed of spatial and temporal branches.

The methods recently described in and follow the same bi-stream approach, introducing two similar yet effective architectures that are based on conventional bi-stream, employing two sub-networks for detecting saliency in still images and temporal data while using motion sub-network to enhance the sub-network for still images. In particular, Yan et al. present an effective spatial refinement network and then used a recurrent module to obtain both accurate contrast inference and coherence enhancement.

The bi-stream methods frequently lack relatively adequate yet gratifying ability to accomplish temporal saliency estimation in the video frames. Single stream methods shrink the problem domain via considering the output of the color saliency deep model as the input for the latter branch to estimate temporal saliency, achieving the best performance among the currently available methods. To the best of our knowledge, there are only several methods following the single stream approach. Specifically, observing the burden of considering multi-scale features in Region-based CNN, Song et al. use ConvLSTM with pyramid dilated convolution architecture to obtain consistency in the large space-time margin. Based on the method of, Fan et al. introduce a saliency-shift loss guided attention mechanism to strengthen the discrimination of the ConvLSTM network.

From the perspective of conventional LSTMs, the internal state of each memory cell contains the accumulated information about the spatiotemporal structure, while it may fail to capture these salient motions, because the gate units do not explicitly consider the impact of dynamic structures present in input sequences. To solve this limitation, Veeriah et al. introduce a differential RNN (dRNN) model that integrates the Derivative of States (DoS) into the conventional RNN, aiming to quantifies the change of information at each time thereby learning the evolution of action states. It is worthy mentioning that the concept of “salient motion” used in the dRNN usually relates to those motions that make the current action more discriminative than others towards the action recognition task. In fact, the key idea of our work is partially similar to the dRNN, which attempts to resort those frames containing salient motions to impact the upcoming new round of network training. However, there exists one critical aspect making the meaning of “salient motion” in our method different to that in the dRNN, i.e., the DoS adopted in the dRNN can only indicate whether the current frame contains ‘salient motion’—it may be the partial movements, belonging to different objects or even dynamic backgrounds. Though such kind of salient motions can benefit the action recognition task, it may be not suitable for the video salient object detection because of the non-quality-aware nature of the DoS. In sharp contrast, our method resorts the consistency between color saliency and motion saliency to measure the quality of frames, and only those frames containing high-quality motions will be used to impact the upcoming re-learning process, which is more suitable for the video salient object task.

Specifically, our idea is partially similar to in the video object segmentation task, of which the key idea of paper is to utilize keyframe strategy for fine-tuning. However, our method is different from them. BubbleNets learns to sort frames via a performance-based loss function and all the data for training the network derives from annotated dataset. After that, the method employs one selected keyframe with its accurate groundtruth for fine-tuning. Therefore, this method is a supervised method. On the contrary, our method is a semi-supervised one since we only need pseudo groundtruth where we directly employ those low-level saliency maps instead of accurate annotated data in our online training. As for the other paper, Li et al. attempt to re-train the model with the first frame and the selected keyframes which are obtained via several metrics of segmentation quality, e.g., average region number, temporal pixel change rate and segmentation compactness. On the one hand, the main task is different from ours because the paper needs to know an accurate groundtruth of the first frame in advance. On the
other hand, all the metrics of segmentation quality adopted in the paper [50] are based on the conventional hand-crafted rationale. In sharp contrast, our novel training scheme takes advantage of both deep model based motion saliency extracted from color-coded optical flow and color saliency extracted from original frame. In many cases, our method is more robust than simply applying hand-crafted features. In addition, our method has a relatively higher computing efficiency.

III. METHOD OVERVIEW

As shown in Fig. 2, our method consists of 6 steps: STEP 1. Color-coded optical flow computation; STEP 2. Motion saliency computation; STEP 3. Color saliency computation; STEP 4. Low-level saliency computation; STEP 5. Perform our keyframe strategy to locate the informative video frames and an online learning is proposed in the weakly supervised manner; STEP 6. Complementary fusion between the pre-computed low-level saliency and the newly computed saliency maps after using the weakly supervised online learning, ensuring the temporal consistency in the long-term manner to improve the overall detection performance.

Given an input video, we first compute its color-coded optical flow. According to the optical flow data, motion saliency maps are generated via our newly trained high performance motion saliency deep model, as will be detailed in Sec. IV-C. Furthermore, we use the pre-trained color deep saliency model to initially conduct color saliency estimation and then fuse it with the motion saliency map as the low-level saliency map, see details in Sec. IV-D.

Based on the computed low-level saliency maps, we use the newly designed keyframe strategy (Sec. IV-A) to locate the frames with relatively high-quality low-level saliency predictions that will subsequently be applied as the pseudo training ground-truth. Then, we adopt the newly self-paced online learning scheme (Sec. IV-B) to re-train the color deep model, in which the re-trained color deep model can simultaneously conduct saliency estimation from both the spatial and temporal perspectives. Finally, we fuse the original low-level saliency maps with the saliency predictions computed by the newly re-trained color deep model, achieving an optimal fusion status with much improved detection performance.

IV. MODEL ADAPTATION

A. Color Saliency Deep Model Preliminaries

Almost all of the current mainstream FCN-based image saliency deep models [27], [51] have enabled the end-to-end saliency detection, requiring much less computational cost than the conventional CNN-based methods. Considering feature map \( \mathbf{X} \) and feature map \( \mathbf{X}' \) after convolution operation, the convolutional operator can be formulated as given by Eq. (1)

\[
\mathbf{X}' = \mathbf{W} \ast \mathbf{X} + \mathbf{b},
\]

where \( \mathbf{W} \) and \( \mathbf{b} \) denote kernel and bias, respectively, in which the convolutional operator is frequently applied to downsample its input.

Although the down-sampled feature maps are valuable for coarsely locating the salient objects, they tends to lose the tiny details, generating the detected saliency map with obscured object boundary. To alleviate this problem, the FCN-based state-of-the-art methods frequently adopt multiple de-convolution layers, and the overall forward propagation of the standard FCNs can be formulated as Eq. (2)

\[
\hat{\mathbf{S}} = \text{DeConv}(\text{Conv}(\mathbf{I}; \alpha); \beta),
\]

where the function \( \text{DeConv}(\cdot) \) denotes a series of de-convolutional operators to ensure the feature map of last layer with resolution identical to the input image \( \mathbf{I} \), \( \text{Conv}(\cdot) \) denotes the convolutional operations in the adopted backbone network (e.g., VGG), the symbol \( \hat{\mathbf{S}} \) denotes the final saliency prediction, and symbols \( \alpha \) and \( \beta \) represent all of the learned parameters of the convolution and de-convolution layers, respectively.

To measure and minimize the error in the training stage, the FCN-based methods commonly use the cross-entropy loss given by Eq. (3)

\[
L(\alpha; \beta) = - \sum_{i,j} G_{i,j} \cdot \log \left( P(G_{i,j} = 1|\alpha; \beta; \mathbf{I}) \right) - \sum_{i,j} (1 - G_{i,j}) \cdot \log \left( P(G_{i,j} = 0|\alpha; \beta; \mathbf{I}) \right),
\]

where \( G_{i,j} = 1 \) denotes foreground, and \( G_{i,j} \) denotes the value of the ground-truth map at location \( (i, j) \), e.g., \( G_{i,j} = 0 \) denotes background; \( P \) denotes the probability of the final activation value at location \( (i, j) \).

B. Conventional Motion Saliency

As one of the most important saliency cues in video data, the motions between two consecutive video frames can be easily sensed by the optical flow algorithm within a pixel-wise manner. Given two consecutive video frames, the optical flow algorithm will output two directional (horizontal and vertical) gradient maps, i.e., \( \mathbf{VX} \in \mathbb{R}^{w \times h} \) and \( \mathbf{VY} \in \mathbb{R}^{w \times h} \), in which \( w \) and \( h \) denote the width and height, respectively, of the given image \( \mathbf{I} \). In fact, while the heavy computational cost is the major performance bottleneck of the conventional optical flow algorithm [52], we may choose to use the deep-learning-based optical flow method (e.g., LiteFlowNet [53]) to achieve an extremely low computational cost at the expense of a slight performance degeneration.

- **Fig. 2.** Architectural overview of our video saliency detection method in which the red arrows denote the color-related data flows, the green arrows denote the motion-related data flows, the blue arrows denote the data flows of our weakly supervised online learning scheme, and the black arrows denote the final bi-stream spatiotemporal saliency fusion.
where \( \omega \) is an empirically predefined weighting parameter, \( || \cdot ||_2 \) denotes the L2 Euclidean distance, and \( \Phi_{i,j} \) denotes the region in the vicinity of pixel \((i, j)\).

In fact, while Eq. (4) ensures that motion saliency computation is effective in most cases, it may still encounter cases with low-quality motion saliency that is mainly induced by either the anomaly object movements (e.g., the partial/intermittent movement in Fig. 3) or the fast view scale/angle change (e.g., camera jitter); see the qualitative demonstrations in Fig. 4.

C. A Novel Scheme to Adapt Color Saliency Deep Model for Motion Saliency Estimation

The deep-learning-based color saliency methods have received extensive research attentions with much significant performance improvements in the past five years. However, the up-to-date motion saliency computations still follow either the hand-crafted contrast computation mentioned in Sec. IV-B or the solely FCN-based simple networks [39] for which the core methodology has rarely benefited from the up-to-date color saliency related deep learning techniques. Moreover, the current mainstream deep-learning-based color saliency methods can easily assign large saliency values to the regions that are perceptually distinct from their surroundings.

Thus, all of the above issues motivate us to explore a feasible approach for converting the off-the-shelf color saliency deep models for the high-performance motion saliency estimation. In fact, the underlying rationale of saliency computation over temporal domain is in many ways identical to the classic color saliency computation, yet the difference between them is that the motion saliency is developed from the contrast computation over the optical flow spanned feature space rather than from the color-saliency-based spatial contrast. Therefore, we propose to use the optical flow data to fine-tune a pre-trained color saliency deep model (we choose ResDSS [15] in this paper) for motion saliency detection.

To handle the inconsistent data channel between optical flow data (i.e., 2-channel \( VX \), \( VY \)) and original color image (i.e., 3-channel RGB), we follow the coding scheme mentioned in [55] to convert the 2-channel optical flow data into the 3-channel color-coded version, in which it uses the 55 pre-defined colors with different hue and saturation to represent the flow orientation and magnitude, respectively; see the color coded optical flow data in the 3rd row of Fig. 4.

Next, we will use the newly coded optical flow data to fine-tune the color saliency deep model, where the key steps can be summarized as follows:
1) we use the widely adopted training set (30 sequences) of the Davis dataset to train our motion model;
2) for each video frame in the adopted training set, we use the optical flow method [52] to compute its optical flow data and then convert these data into 3-channel data using the aforementioned coding scheme;
3) we use the coded optical flow data to fine-tune the pre-trained ResDSS model with the learning rate of 1e-9 using the total loss function \( (L_{total}) \) given by Eq. (5).

\[
L_{total}(\alpha; \beta) = - \sum_k \sum_{i,j} G_{i,j} \log \left( P_k(G_{i,j} = 1|\alpha_k; \beta_k; F) \right) - \sum_k \sum_{i,j} (1 - G_{i,j}) \log \left( P_k(G_{i,j} = 0|\alpha_k; \beta_k; F) \right),
\]

where \( F \) represents the input color-coded optical flow image, and \( k \) denotes the index of the side-output layer or the fusion layer in ResDSS. Thus, the final motion saliency \( MS \) can be formulated as \( \sum_k \tilde{S}_k|k| \), where the \( \tilde{S}_k \) denotes the \( k \)-th motion saliency prediction, and \(|k| \) denotes the total number of the prediction maps. The overall qualitative demonstration of the computed motion saliency maps is shown in Fig. 4.

Our motion deep model (i.e., the re-trained ResDSS) can be trained fast (within 2 hours) by using relatively little training data (only 2K). Moreover, our method is flexible and can adapt any color saliency deep model for motion saliency detection. In particular, it is important to mention that the motion saliency detection performance can easily be further improved if we adopt a much stronger pre-trained color saliency deep model; for example, we have re-trained the RADF model [17] to achieve better motion saliency performance (see Ours RADF in Table I).

It is worthy mentioning that the overall performance of our method may vary with different optical flow sources, and we may well achieve better overall performance if we adopt more

![Fig. 3. An failure case demonstration that the conventional motion saliency (FL18) [11] toward the partial movements.]

![Table I Quantitative comparison between different motion deep model training strategies.](image-url)
accurate optical flow method, such as the PWCNet [56]. With regarding to this issue, please refer to the quantitative proofs in Table 1 where the ResDSS+ represents the results using the conventional optical flow data, the ResDSS∗ represents the re-trained motion model using the optical flow data of the LiteFlowNet [53], the ResDSSS represents the re-trained motion model using the optical flow data of the SPyNet [57], the ResDSSP represents the re-trained motion model using the optical flow data of the PWCNet [56].

D. Low-level Saliency Computation

We have obtained the newly learned motion saliency MS that can be used as the motion sub-branch in our bi-stream network mentioned in Sec. III. Moreover, in our bi-stream network, any off-the-shelf pre-trained color saliency deep model can be used as the color sub-branch and we represent its saliency prediction, namely color saliency map as CS. Thus far, we formulate the low-level saliency via the widely adopted multiplicative-based fusion as given by Eq. 6

\[ LS = MS \odot CS, \]

where \( \odot \) denotes the element-wise Hadamard product. Since the fusion procedure has simultaneously considered both the spatial and temporal saliency cues, its overall performance can be superior to either the motion saliency or the color saliency, as quantitatively demonstrated in Sec. VI-C.

V. WEAKLY SUPERVISED ONLINE LEARNING

A. Keyframe Strategy

Compared with the conventional contrast computation based motion saliency, our novel motion saliency deep model can output the motion saliency between the consecutive video frames correctly. However, due to its limited sensing scope (only 2 frames) over the temporal scale, the motion saliency produced by our motion saliency deep model may be perceptually different from the real motion saliency. Moreover, the sensing scope of the color branch in our bi-stream network is also limited within a single video frame, making the fused low-level saliency (LS) temporally inconsistent. To solve this issues, here, we propose a novel weakly supervised online learning to adapt the color branch for a long-term spatiotemporal saliency detection. It is also important to mention that we choose to leave the motion branch unchanged to avoid over-fitting.

The key rationale of our weakly supervised online learning is to use the high quality low-level saliency as the pseudo learning ground truth, where we propose a novel keyframe strategy to locate the video frames with high-quality low-level saliency predictions. Our keyframe strategy is inspired by the phenomenon that only those video frames with both high-quality color and motion saliency may correlate to the cases having high-quality fused low-level saliency. Thus, we may assume that the degree of consistency between the color and motion saliency maps has a positive relationship with the quality degree of the fused low-level saliency, which motivates us to use such consistency degree to locate those video frames with high-quality low-level saliency predictions.
To further validate this assumption, we demonstrate the correlation between the consistency of color and motion saliency, and the fused low-level saliency quality via multiple quantitative experiments as Fig. 5. In these experiments, all video frames are re-ordered according to the qualities of their fused low-level saliency maps, and these qualities are estimated by computing the structural similarity between each low-level saliency map and its saliency GT. Thus, the left side of each sub figure represents those video frames with high-quality fused low-level saliency maps, while the right side represents the low-quality cases. It should be noted that all video sequences are individually measured, normalized, and aligned to an identical form, containing 40 intervals in total.

We obtain the overall results by averaging all video sequences of each dataset, where the color of each interval from warm to cold represents the probability to be selected as keyframe. As shown in Fig. 5, we have demonstrated the quantitative results of our methods using two different baseline models (i.e. RADF and RAS) over five benchmark datasets, and all results have clearly suggested a positive relationship between the color and motion saliency consistency degree, and the fused low-level saliency quality, which also demonstrates the effectiveness of the proposed keyframe selection strategy.

Here we use the non-overlapping ratio NR to measure the degree of consistency between the motion saliency MS and the color saliency CS. Thus, for the i-th video frame, we formulate its non-overlapping ratio NR between the color and motion saliency predictions as Eq. 7 where the value of NR ∈ [0, 1] is inversely related to its degree of quality.

\[
NR_i = \frac{1}{\|T(MS_i + CS_i)\|_0} \cdot \frac{1}{T(MS_i) + T(CS_i) + C} \cdot \text{abs}(T(MS_i) - T(CS_i)) \cdot \text{size invariant factor},
\]

where C is a pre-defined constant value (0.001) to avoid division by zero, \(T(\cdot)\) represents the hard-threshold filter that assigns 0 to the elements with the value smaller than 0.1, \(\text{abs}(\cdot)\) denotes the absolute function, and \(\| \cdot \|_0\) and \(\| \cdot \|_1\) represent the L0 and L1 norms, respectively; the “scale invariant factor” is used to ensure that our quality assessment is insensitive to the salient region size, and we show its impact over the weakly supervised learning in Fig. 6. Once the non-overlapping ratio NR is obtained, we select the video frames with \(NR < 0.6\) as the keyframes, see quantitative proofs in Table V. Moreover, we show the results of keyframe selection strategy of a challenging video sequence in Fig. 7.

**B. Self-paced Online Learning**

Given an input video sequence, we use the low-level saliency predictions of the previously located keyframes as the learning pseudo ground truth to weakly fine-tune the color sub-branch. For the selected keyframes, the high-quality low-level saliency predictions provide a good representation of the long-term spatiotemporal information of the given video sequence. Furthermore, since the low-level saliency map LS ∈ [0, 1], our self-paced online learning uses the Euclidean loss to replace the conventional cross-entropy loss that is given by Eq. 8.

\[
L_E(\alpha; \beta) = \frac{1}{2N} \sum_{n \in N} \left| \left| LS_n - \hat{EM}(\alpha; \beta; I_n) \right| \right|^2,
\]

where \(L_E\) is the Euclidean loss, N is the training batch size, and \(\hat{EM}\) is the estimated saliency map of the color sub-branch.

It should also be noted that we name our weakly supervised learning scheme “an online manner” for the following reasons: 1) rather than directly conduct model fine-tuning over the
original color sub-branch, we fine-tune a “twin color sub-branch” with network weights identical to those of the original color sub-branch to achieve improved detection performance; 2) our method is data-driven; i.e., the newly fine-tuned color sub-branch is only available for the given video sequence; 3) due to a limited problem domain, our fine-tuning procedure for a given video sequence is extremely fast, and can be converged in a very short time.

In fact, our self-paced online learning scheme can enhance the deep feature distance between the salient regions and its non-salient nearby surroundings, ensuring its spatial saliency temporally smoothness, as shown in the qualitative demonstrations presented in Fig. 8.

Once the twin color sub-branch is computed, we attempt to fuse its saliency predictions with the original low-level saliency maps as the final video saliency detection results (Eq. 9), in which the fused saliency outperform its inputs slightly, as demonstrated quantitatively in Sec. VI-C.

\[ FS = \hat{EM}' \odot LS, \]

where FS denotes our final saliency map, and \( \hat{EM}' \) denotes the saliency prediction of our fine-tuned color sub-branch.

VI. EXPERIMENTS AND EVALUATIONS

A. Datasets

We evaluate our approach on 5 most widely used public available datasets, including DAVIS2016(480p) [54], SegV2 [59], MCL [33], UVSD [85], and FBMS [60]. All of the ground-truths of these datasets are well-annotated at the pixel level.

B. Evaluation Metrics

To better verify and validate the performance of our method, we use 3 widely adopted metrics, namely, the mean absolute error (MAE), the maximum F-measure value (maxF) and the average F-measure value (avgF).

We segment the video saliency detection results of different methods using the same integer threshold \( T \in [0, 255] \). Then, the regions are labeled as 1 when their saliency values are greater than \( T \) and the other regions are set to 0. Since the recall rate is inversely proportional to the precision, the tendency of the trade-off between precision and recall can provide an accurate indication of the overall video saliency detection performance. The F-measure can be computed via

\[ \text{F-measure} = \frac{(\beta^2 + 1) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}, \]

where Precision is the average precision rate, Recall is the average recall rate, and \( \beta^2 = 0.3 \) is used to bias toward the precision rate, as was first suggested in [61] and subsequently adopted by many significant studies.

MAE is defined as the average per-pixel difference between a saliency map \( S \) and its corresponding ground truth \( G \). Here, \( S \) and \( G \) are normalized to the range [0, 1].

\[ \text{MAE} = \frac{\sum \text{abs}(S - G)}{W \times H}, \]

where \( W \) and \( H \) are the width and height of the saliency map, respectively.

C. Adaptiveness Analysis

To validate the adaptiveness, we have tested our weakly supervised online learning scheme over five off-the-shelf color saliency deep models and one video saliency deep model, including RADF [17], ResDSS [15], RAS [16], CPD [22], PoolNet [23] and SSAV [24]. We have conducted the component evaluation to prove the effectiveness of our approach, with the results presented in Table III. Specifically, due to the use of our novel motion saliency, the fused low-level saliency exhibits a significant performance improvement. Furthermore, the overall performance of our novel models also persistently and remarkably outperforms its low-level saliency. In particular, as for the FBMS dataset, the color saliency deep models based solely on spatial information perform well for this dataset because the FBMS dataset is dominated by spatial information. Nevertheless, benefiting from the newly sensed temporal information, our method still outperforms the color saliency deep models apparently according to the import metric maxF.

The experimental results show that the experimental model SSAV performs well on two datasets, but finetuning SSAV on the target video suffers from the performance decrease...
TABLE II

**Quantitative comparison results between our method and 15 SOTA methods over 5 public available datasets.** The column-wise best are marked with **red** color, the 2ND-bests are marked with **green** color, and the 3RD-bests are marked with **blue** color.

| Dataset | Metric | Ours | SegV2 | UVSD | FBMS |
|---------|--------|------|-------|------|------|
|         | RAADF  | ResDSS | CAS   | PoolNet | SSAV |
|         | MAE    |       |       |       |      |
| DAVIS   | maxF   | 0.85  | 0.85  | 0.82  | 0.81 |
|         | avgF   | 0.79  | 0.78  | 0.77  | 0.79 |
|         | MAE    | 0.29  | 0.29  | 0.29  | 0.29 |
| SegV2   | maxF   | 0.84  | 0.83  | 0.82  | 0.80 |
|         | avgF   | 0.76  | 0.75  | 0.74  | 0.74 |
|         | MAE    | 0.26  | 0.26  | 0.26  | 0.26 |
| UVSD    | maxF   | 0.79  | 0.78  | 0.77  | 0.76 |
|         | avgF   | 0.69  | 0.68  | 0.67  | 0.67 |
|         | MAE    | 0.08  | 0.08  | 0.08  | 0.08 |
| FBMS    | maxF   | 0.80  | 0.79  | 0.78  | 0.77 |
|         | avgF   | 0.62  | 0.61  | 0.61  | 0.61 |
|         | MAE    | 0.10  | 0.10  | 0.10  | 0.10 |

**D. Implementation Details**

We implement our method using Matlab2016b with the popular Caffe platform. All of the input frames are resized to the spatial resolution of 300×300. All of the quantitative evaluations are conducted on a desktop computer with NVIDIA GTX 1080 GPU, Intel i7-6700k 4.00 GHz CPU (4 threads with 8 cores) and 32 GB RAM. We conduct the training procedure using the widely adopted settings, namely, stochastic gradient descent (SGD) with a momentum of 0.95 and weight decay of 0.0005. We reduce the learning rate of the chosen image model by a factor of 0.1. In our online training stage, assuming the number of the keyframe in the current video is \(N\), we set parameter \(\lambda = 10\). The number of the keyframe in the current video is \(N\), and we set parameter \(\lambda = 10\).
Table V

| Data Set | Metric | NR: 0.3 | NR: 0.5 | NR: 0.6 | NR: 0.75 | NR: 0.85 |
|----------|--------|---------|---------|---------|----------|---------|
| MCL [33] | maxF   | 0.791   | 0.792   | 0.793   | 0.789    | 0.791   |
|          | avgF   | 0.672   | 0.680   | 0.679   | 0.673    | 0.670   |
|          | MAE    | 0.042   | 0.042   | 0.041   | 0.043    | 0.042   |

For time analysis, since the keyframe number is determined by the total number of video frames in the given video sequence, our method may be somewhat time-consuming for adapting the color saliency deep model for video sequence with a large \( N \); i.e., we have tested the average time per frame over all of the benchmarks. The runtime (second per frame) of all the methods are shown in Table V1.

E. Comparison With the State of the Art

We have compared our method with 15 state-of-the-art methods, including SSAV19 [24], CPD19 [22], ResDSS19 [15], PoolNet19 [23], FL18 [11], DLVSD18 [39], RAS18 [10], RADF18 [17], FD17 [8], SGSP17 [35], RFCN16 [51], MDF16 [18], MC15 [33], GF15 [11], SA15 [32]. Ours-A, Ours-B, Ours-C, Ours-D, Ours-E, Ours-F denote the final saliency results after using our scheme over the model RADF, ResDSS, RAS, CPD, PoolNet, SSA V respectively.

As for the choice of the NR threshold mentioned in Sec. V-A we have conducted an ablation study on it over the challenging MCL [33] dataset, see the quantitative proofs in Table V. On the one hand, since those frames of challenge video sequence usually have low-quality color or motion saliency leading to high NR values, it may be difficult to ensure a high diversity in those selected keyframes if we choose an extremely small NR. On the other hand, a higher NR is more likely to result in more less-trustworthy keyframes. Moreover, the keyframe increase may also burden our online learning. Therefore, we decide to choose 0.6 as the threshold.

E. Comparison With the State of the Art

We have compared our method with 15 state-of-the-art methods, including SSAV19 [24], CPD19 [22], ResDSS19 [15], PoolNet19 [23], FL18 [11], DLVSD18 [39], RAS18 [10], RADF18 [17], FD17 [8], SGSP17 [35], RFCN16 [51], MDF16 [18], MC15 [33], GF15 [11], SA15 [32]. Ours-A, Ours-B, Ours-C, Ours-D, Ours-E, Ours-F denote the final saliency results after using our scheme over the model RADF, ResDSS, RAS, CPD, PoolNet, SSA V respectively.

Moreover, the keyframe increase may also burden our online learning. Therefore, we decide to choose 0.6 as the threshold.

As for the choice of the NR threshold mentioned in Sec. V-A we have conducted an ablation study on it over the challenging MCL [33] dataset, see the quantitative proofs in Table V. On the one hand, since those frames of challenge video sequence usually have low-quality color or motion saliency leading to high NR values, it may be difficult to ensure a high diversity in those selected keyframes if we choose an extremely small NR. On the other hand, a higher NR is more likely to result in more less-trustworthy keyframes. Moreover, the keyframe increase may also burden our online learning. Therefore, we decide to choose 0.6 as the threshold.

For time analysis, since the keyframe number is determined by the total number of video frames in the given video sequence, our method may be somewhat time-consuming for adapting the color saliency deep model for video sequence with a large \( N \); i.e., we have tested the average time per frame over all of the benchmarks. The runtime (second per frame) of all the methods are shown in Table V1.

E. Comparison With the State of the Art

We have compared our method with 15 state-of-the-art methods, including SSAV19 [24], CPD19 [22], ResDSS19 [15], PoolNet19 [23], FL18 [11], DLVSD18 [39], RAS18 [10], RADF18 [17], FD17 [8], SGSP17 [35], RFCN16 [51], MDF16 [18], MC15 [33], GF15 [11], SA15 [32]. The quantitative comparison results (the F-measure curves) are presented in Fig. 9. As shown in rows 1-2 of Fig. 10, our method handles these long-period motionless sequences well. Moreover, in such cases, almost all of the current state-of-the-art video saliency detection methods easily give massive failure detections. Furthermore, our methods can still handle the video scenes with complex backgrounds well; such video scenes are usually correlated with a challenging saliency estimation over the spatial domain, proving the effectiveness of our method for adapting the color saliency deep models for temporal saliency estimation, as shown in rows 3-10 of Fig. 10.

The quantitative results obtained by the most recent advanced video saliency detection method SSAV [24] are shown in Table II and it is observed that the performance of our method is comparable to that of SSAV. Specifically, Ours-F clearly outperforms SSAV by 4.1%, 6.5%, and 10.1% in terms of maxF over the DAVIS, SegV2 and MCL datasets, respectively. Furthermore, Ours-F achieved an MAE values that is very close to the MAE for the SSA V method on the challenging DA VIS, SegV2, MCL and UVSD datasets. However, our method fails to perform well over the FBMS dataset, which is mainly due to the fact that the FBMS dataset is dominated by spatial information with frequent intermittent movement, making the video saliency detection by using both the spatial and temporal saliency cues much more difficult. Benefiting from the long-term attribute of our method, our method still can achieve the top three maxF value for the FBMS dataset, as shown in Table II.

We qualitatively compare the results of the different methods in Fig. 10. As shown in rows 1-2 of Fig. 10, our method handles these long-period motionless sequences well. Moreover, in such cases, almost all of the current state-of-the-art video saliency detection methods easily give massive failure detections. Furthermore, our methods can still handle the video scenes with complex backgrounds well; such video scenes are usually correlated with a challenging saliency estimation over the spatial domain, proving the effectiveness of our method for adapting the color saliency deep models for temporal saliency estimation, as shown in rows 3-10 of Fig. 10.

The quantitative results obtained by the most recent advanced video saliency detection method SSAV [24] are shown in Table II and it is observed that the performance of our method is comparable to that of SSAV. Specifically, Ours-F clearly outperforms SSAV by 4.1%, 6.5%, and 10.1% in terms of maxF over the DAVIS, SegV2 and MCL datasets, respectively. Furthermore, Ours-F achieved an MAE values that is very close to the MAE for the SSA V method on the DAVIS and SegV2. It should also be noted that the SSAV method adopts an extremely large training dataset (with the additional eye fixation data), while by contrast, our deep models are trained using the Davis training set. With the rapid development of the color saliency deep models, we believe that our method will eventually outperform the SSAV method.

VII. Conclusion

This paper has proposed a novel weakly supervised scheme to adapt image saliency deep models for video data. Our method can generate a novel motion saliency sub-branch via
fine-tuning the off-the-shelf image saliency deep model using the color-coded optical flow data. Furthermore, we propose the newly-designed keyframe strategy to locate the frames with high-quality spatiotemporal saliency predictions. Then, we have used these high-quality predictions as the pseudo ground truth for the weakly supervised online training, enabling all of the off-the-shelf image saliency deep models to be adapted for the current video sequence as the new color sub-branch of our method. Our method is simple, flexible, and effective, and is likely to inspire future work even in the case that our color model adapted method is only comparable to the current leading state-of-the-art video saliency detection methods.

Acknowledgments. This research was supported in part by National Key R&D Program of China (No. 2017YFF0106407), National Natural Science Foundation of China (No. 61802215 and No. 61806106), Natural Science Foundation of Shandong Province (No. ZR201807120086) and National Science Foundation of USA (No. IIS-1715985, IIS0949467, IIS-1047715, and IIS-1049448).

REFERENCES
[1] W. Wang, J. Shen, and L. Shao, “Consistent video saliency using local gradient flow optimization and global refinement,” IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 4185–4196, 2015.
[2] C. Chen, S. Li, H. Qin, and A. Hao, “Structure-sensitive saliency detection via multilevel rank analysis in intrinsic feature space,” IEEE Transactions on Image Processing, vol. 24, no. 8, pp. 2303–2316, 2015.
[3] Y. Fang, Z. Chen, W. Lin, and C. Lin, “Saliency detection in the compressed domain for adaptive image retargeting,” IEEE Transactions on Multimedia, vol. 21, no. 9, pp. 3888–3901, 2012.
[4] S. Li, M. Xu, Y. Ren, and Z. Wang, “Closed-form optimization on saliency-guided image compression for hevc-msp,” IEEE Transactions on Multimedia, vol. 20, no. 1, pp. 155–170, 2018.
[5] C. Chen, S. Li, H. Qin, and A. Hao, “Real-time and robust object tracking in video via low-rank coherency analysis in feature space,” Pattern Recognition, vol. 48, no. 9, pp. 2885–2905, 2015.
[6] C. Peng, C. Chen, Z. Kang, J. Li, and Q. Cheng, “Res-pca: A scalable approach to recovering low-rank matrices,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7317–7325.
[7] C. Chen, S. Li, H. Qin, and A. Hao, “Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis,” Pattern Recognition, vol. 52, pp. 410–432, 2016.
[8] C. Chen, S. Li, Y. Wang, H. Qin, and A. Hao, “Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3156–3170, 2017.
[9] C. Chen, G. Wang, C. Peng, X. Zhang, and H. Qin, “Improved robust video saliency detection based on longterm spatial-temporal information,” IEEE Transactions on Image Processing, vol. 29, pp. 1090–1100, 2019.
[10] C. Chen, J. Wei, C. Peng, W. Zhang, and H. Qin, “Improved saliency detection in rgb-d images using two-phase depth estimation and selective deep fusion,” IEEE Transactions on Image Processing, vol. 29, pp. 4296–4307, 2020.
[11] C. Chen, S. Li, H. Qin, Z. Pan, and G. Yang, “Bi-level feature learning for video saliency detection,” IEEE Transactions on Multimedia, vol. 20, no. 12, pp. 3324–3336, 2018.
[12] G. Li, Y. Xie, T. Wei, K. Wang, and L. Lin, “Flow guided recurrent neural encoder for video salient object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3243–3252.
[13] H. Song, W. Wang, S. Zhao, J. Shen, and K. Lam, “Pyramid dilated deeper convlstm for video salient object detection,” in Proceedings of the European Conference on Computer Vision, 2018, pp. 715–731.
[14] X. Zhou, Z. Liu, C. Gong, G. Li, and M. Huang, “Video saliency detection using deep convolutional neural networks,” in Chinese Conference on Pattern Recognition and Computer Vision, 2018, pp. 308–319.
[15] Q. Hou, M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, “Deeply supervised salient object detection with short connections,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 4, pp. 815–828, 2019.
