Video Salient Object Detection via Fully Convolutional Networks

Wenguan Wang, Jianbing Shen , *Senior Member, IEEE*, and Ling Shao, *Senior Member, IEEE*

***Abstract*— This paper proposes a deep learning model to efficiently detect salient regions in videos. It addresses two important issues: 1) deep video saliency model training with the absence of sufficiently large and pixel-wise annotated video data and 2) fast video saliency training and detection. The proposed deep video saliency network consists of two modules, for capturing the spatial and temporal saliency information, respectively. The dynamic saliency model, explicitly incorporating saliency estimates from the static saliency model, directly produces spatiotemporal saliency inference without time-consuming optical flow computation. We further propose a novel data augmentation technique that simulates video training data from existing annotated image data sets, which enables our network to learn diverse saliency information and prevents overfitting with the limited number of training videos. Leveraging our synthetic video data (150K video sequences) and real videos, our deep video saliency model successfully learns both spatial and temporal saliency cues, thus producing accurate spatiotemporal saliency estimate. We advance the state-of-the-art on the densely annotated video segmentation data set (MAE of .06) and the Freiburg-Berkeley Motion Segmentation data set (MAE of .07), and do so with much improved speed (2 fps with all steps).**

***Index Terms*— Video saliency, deep learning, synthetic video data, salient object detection, fully convolutional network.**

I. Introduction

SALIENCY detection has recently attracted a great amount of research interest. The reason behind this growing popularity lies in the effective use of these models in various vision tasks, such as image segmentation, object detection, video summarization and compression, to name a few. Saliency models can be broadly classified into two categories: human eye fixation prediction or salient object detection. According to the type of input, they can be further categorized into static and dynamic saliency models. While static models take still images as input, dynamic models work on video sequences. In this paper, we focus on detecting distinctive regions in dynamic scenes. *Convolutional neural networks* (CNNs) have been successfully utilized in many fundamental areas of computer vision, including object detection [1], [4], semantic segmentation [5], and still saliency detection [7], [8]. Inspired by this, we investigate CNNs to another computer vision task, namely video saliency detection.

This work was supported in part by the National Basic Research Program of China 973 Pro- gram under Grant 2013CB328805, in part by the National Natural Science Foundation of China under Grant 61272359, in part by the Fok Ying-Tong Education Foundation for Young Teachers, and in part by the Joint Building Program of Beijing Municipal Education Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kalpana Seshadrinathan. *(Corresponding author: Jianbing Shen.)*

W. Wang and J. Shen are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: shenjianbing@bit.edu.cn).

L. Shao is with the School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, U.K. (e-mail: ling.shao@ieee.org).

TABLE I

STATISTICS FOR IMAGENET [9], FBMS [10], SEGTRACKV2 [11], VSB100 [12] AND DAVIS [13] DATASETS

Dataset	Ref	#Clips	#Annotations (frame/image)
ImageNet	[9]	-	$\sim 1.3 \times 10^6$
FBMS	[10]	59	$\sim$ 500
SegTrackV2	[11]	14	$\sim$ 1500
VSB100	[12]	100	$\sim$ 600
DAVIS	[13]	50	$\sim$ 4000

The first problem of applying CNNs to video saliency is the lack of sufficiently large, densely labelled video training data. As far as we know, the successes of CNNs in computer vision are largely attributed to the availability of large-scale annotated images (*e.g.*, ImageNet [9]). However, existing video datasets are too small to provide adequate training data for CNNs. In Table 1, we list the statistics of the ImageNet dataset and widely adopted video object segmentation datasets, including FBMS [10], SegTrackV2 [11], VSB100 [12] and DAVIS [13]. It can be observed that, the existing video datasets rarely match existing image datasets like ImageNet, in either quality or quantity. Besides, considering the high correlation between the frames from same video clip, existing video datasets are far unable to meet the needs of training CNNs for pixel-level video applications, like video salient object detection. On the other hand, for the moment, creating such a large-scale video dataset is usually infeasible, because annotating videos is complex and time-consuming. To this end, we propose a video data augmentation approach to synthetically generating labeled video training data, which explicitly leverages existing large-scale image segmentation datasets. The simulated video data are easily accessible and rapidly generated, close to realistic videos and present various motion patterns, deformations, companied with automatically generated annotations and optical flow. The experimental results via these automatically generated videos clearly demonstrate the practicability of our strategy.

Our video data synthesis approach clears the underlying challenge for learning CNNs for many applications in video processing, where dynamic saliency detection is of no exception. Another challenge for detecting saliency in dynamic scenarios derives from the natural demand of this task.

As suggested by human visual perception research [14], [15], when computing dynamic saliency maps, video saliency models need to consider both the spatial and the temporal characteristics of the scene. We propose a deep video saliency model for producing spatiotemporal saliency via fully exploring both the static and dynamic saliency information. The proposed model adopts fully convolutional networks (FCNs) [5] for pixel-wise saliency prediction. Associated with existing rich image saliency data, the static saliency is deeply exploited and explicitly encoded in the deep learning process via transferring and fine-tuning recent success in image classification [16]. For learning dynamic saliency cues, the proposed deep video saliency model learns from a large number of labelled videos, including both human-generated and natural video data, in a supervised learning mode. The static saliency is integrated into dynamic saliency detection process, thus for directly producing final spatiotemporal saliency estimation.

Another important contribution of this work is that our deep video saliency model is much more computationally efficient compared with existing video saliency models. Salient object detection is a key step in many image analysis tasks as it not only identifies relevant parts of a visual scene but may also reduce computational complexity by filtering out irrelevant segments of the scene. In recent years, some notable video saliency models have been proposed in many computer vision applications, such as video segmentation [17] and video re-timing [19]. However, time efficiency becomes the common major bottleneck for the applicability of existing video saliency algorithms; most computation time has been spent for optical flow computation. Additionally, from the perspective of learning deep networks in dynamic scenes, many schemes [20]–[22] take optical flow as input, causing high computational expenses.

In this work, we propose a both effective and efficient video saliency model, which frees itself from the computationally expensive optical flow estimation. One of the key insights of this paper is that, unlike high-level video applications such as action detection, video saliency can derive from short-term analysis of video frames. Thus we directly capture temporal saliency via learning deep networks from frame pairs, instead of using long-term video information, such as optical flows from multiple adjacent video frames.

We comprehensively evaluate our method on the FBMS dataset [10], where the proposed video saliency model produces more accurate saliency maps than state-of-the-arts. Meanwhile, it achieves a frame rate of 2fps (including all steps) on a GPU. Thus it is a practical video saliency detection model in terms of both speed and accuracy. We also report results on the newly released DAVIS dataset [13] and observe performance improvements over current competitors. Our source code will be available online.[1]

To summarize, the main contributions are threefold:

- We investigate convolutional neural networks for end- to-end training and pixel-wise saliency prediction in dynamic scenes. As far as we know, this is the first work for applying deep learning to video salient object detection.
- We propose a novel training scheme based on synthetically generated video data, which explicitly leverages existing rich image datasets; both static and dynamic saliency information are encoded into a unified deep learning model.
- Our methods are computationally efficient, much faster than traditional video saliency models and other deep networks in dynamic scenes.

[1] *http://github.com/shenjianbing/deepvideosaliency*

II. Related Work

In this section, we give a brief overview of recent works in two lines: saliency detection, and deep learning models in dynamic scenes.

A. Saliency Detection

Saliency detection has been extensively studied in computer vision, and saliency models in general can be categorized into visual attention prediction or salient object detection. The former methods [14], [23]–[25] try to predict scene locations where a human observer may fixate. Salient object detection [26]–[28] aims at uniformly highlighting the salient regions, which has been shown benefit to a wide range of computer vision applications. More detailed reviews of the saliency models can be found in [29] and [30]. Saliency models can be further divided into static and dynamic ones according to their input. In this work, we aim at detecting saliency object regions in videos.

Image saliency detection has been extensively studied for decades and most of the methods are driven by the well-known *bottom-up* strategy. Early bottom-up models [26], [27] are mainly based on detecting *contrast*, assuming salient regions in the visual field would first pop out from their surroundings and computing feature-based contrast followed by various mathematical principles. Meanwhile, some other mechanisms [28], [31], [32] have been proposed to adopt some prior knowledge, such as *background prior*, or global information, to detect salient objects in still images. More recently, deep learning techniques have been introduced to image saliency detection. These methods [7], [33] typically use CNNs to examine a large number of region proposals, from which the salient objects are selected. Currently, more and more methods [34], [36]–[38] tend to learn in an end-to-end manner and directly generate pixel-wise saliency maps via fully convolutional networks (FCNs) [5].

Compared with saliency detection in still images, detecting saliency in videos is a much more challenging problem due to the complication in the detection and utilization of temporal and motion information. So far, only a limited number of algorithms have been proposed for spatiotemporal saliency detection. Early models [52]–[54] can be viewed as simple extensions of exiting static saliency models with extra temporal dimension. Some more recent and notable approaches [2], [3], [6], [17], [19], [46] to this task have been proposed, showing inspired performance and good potentials in many computer vision applications [18], [47], [59], [68], [69]. However, the applicability of these approaches is

severely limited by their high-computational costs. The main computational bottleneck comes from optical flow estimation, which contributes much to the promising results.

In recent years, the border of saliency detection has been extend to capturing common saliency among related images/videos [41]–[43], [45], [48], inferring the salient event with video sequences [40] or scene understanding [44], [50], [51]. However, there are significant differences between above methods and traditional saliency detection, especially considering their goals and core difficulties.

B. Deep Learning Models in Dynamic Scenes

In this section, we mainly focus on famous, deep learning models for computer vision applications in dynamic scenes, including action recognition [20], [55], object segmentation [22], [56], object tracking [57], [58], [60]–[62], attention prediction [21] and semantic segmentation [63], and explore their architectures and training schemes. This will help to clarify how our approach differs from previous efforts and will help to highlight the important benefits in terms of effectiveness and efficiency.

Many approaches [57], [58], [63] directly feed single video frames into neural networks trained on image data and adopt various techniques for post-processing the results with temporal or motion information. Unfortunately, these neural networks give up learning the temporal information which is often very important in video processing applications. A famous architecture for training CNNs for action recognition in videos is proposed in [20], which incorporates two-stream convolutional networks for learning complementary information on appearance and motion. Other works [21], [56] adopt this architecture for dynamic attention prediction and video object segmentation. However, these methods train their models on multi-frame dense optical flow, which causes heavy computational burden.

In the areas of human pose estimation and video object processing, online learning strategy is introduced for improving performance [22], [55], [60]–[62]. Before processing an input video, these approaches generate various training samples for fine-tuning the neural networks learned from image data, thus enabling the models to be optimized towards the object of interest in the test video sequence. Obviously, these models are quite time-consuming and the fine-tuned models are only specialized for specific classes of objects.

In this work, we show the possibilities of learning to detect generic salient objects in dynamic scenes by training on videos and images via an entirely offline manner. We proposed a novel technique for synthesizing video data via leveraging large amounts of image training data. The CNNs model can be efficiently and entirely trained on rich videos and images, thus successfully learning both static and dynamic saliency features. Meanwhile, it directly learns inner relationship between frames, getting rid of time-consuming motion computation. Thus, our algorithm is significantly faster than traditional video saliency methods and the deep learning architectures that demand optical flow as input. In summary, our CNNs model learns to detect video saliency in a fast manner.

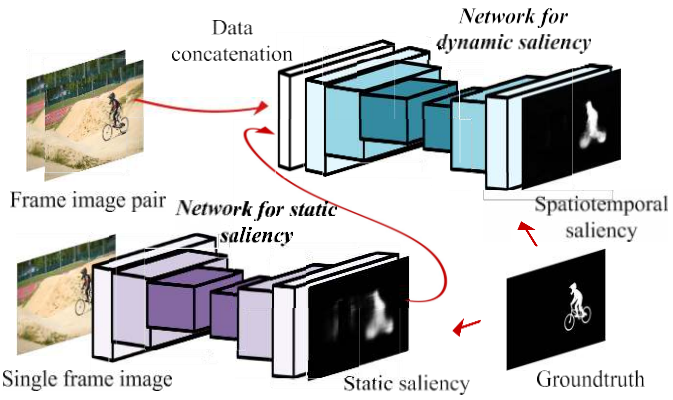


Fig. 1. A schematic representation of our proposed deep video saliency model. Our saliency model composes of two modules, which are designed for capturing the spatial and temporal saliency information simultaneously. The static saliency network (Sec. III-B) takes single frame as input and outputs static saliency estimates. The dynamic saliency network (Sec. III-C) learns dynamic saliency from frame pairs and takes static saliency generated by the first module as prior, thus producing the final spatiotemporal saliency maps.

III. Deep Networks for Video Saliency Detection

In this work, we describe a procedure for constructing and learning deep video saliency networks using a novel synthetic video data generation approach. Our approach generates a large amount of video data (150K paired frames) from existing image datasets, and associates these annotated video sequences with existing video data to learn deep video saliency networks. We first introduce the proposed CNNs based video saliency model in this section and then we describe our video synthesis approach in Sec. IV.

A. Architecture Overview

We start with an overview of our deep video saliency model before going into details below. At a high level, we feed frames of a video into a neural network, and the network successively outputs saliency maps where brighter pixels indicate higher saliency values. The network is trained with video sequences and images and learns spatiotemporal saliency in general dynamic scenes. Fig. 1 shows the architecture of proposed deep video saliency model. Inspired by classical human visual perception research [14], [15], which suggests both static and dynamic saliency cues contribute to video saliency, we design our model with two modules, simultaneously considering both the spatial and temporal characteristics of the scene.

The first module is for capturing static saliency, taking single frame image as input. It adopts fully convolutional networks (FCNs) for generating pixel-wise saliency estimate and utilizes previous excellent pre-trained models on large-scale image datasets. Boosted from rich image saliency benchmarks, this module is efficiently trained for capturing diverse static saliency information of interesting objects. This module is described in detail in Sec. III-B. The second module takes frame pairs and static saliency from the first module as input, and generates final dynamic saliency results. This network is trained from both synthetic and real labelled video data (see details in Sec. III-C).

B. Deep Networks for Static Saliency

A static saliency network takes a single frame image as input and produce a saliency map with the same size of

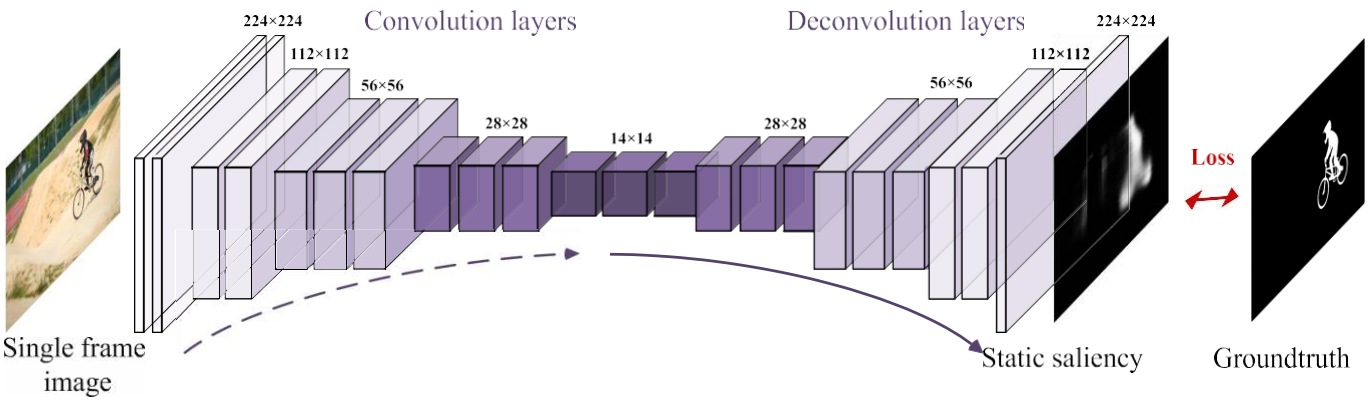


Fig. 2. Illustration of our network for static saliency detection. The network takes single frame image (for example, 224 × 224) as input, adopting multi-layer convolution networks that transforms the input image to multidimensional feature representation, then applying a stack of deconvolution networks for upsampling the feature extracted from the convolution networks. Finally, a fully convolution network with 1 × 1 kernel and *sigmoid* activity function is used to output of a probability map in the same size as input, in which larger values mean higher saliency values.

the input. We model this process with a fully convolutional network (FCN). The bottom of this network is a stack of convolutional layers. Convolutional layer is defined on shared parameters (weight vector and bias) architecture and has translation invariance characteristics. The input and output of each convolutional layer are a set of arrays, called feature maps, with size $h \times w \times c$, where h, w and c are height, width and the feature or channel dimensionality, respectively. For the first convolutional layer, the input is the color image, with pixel size h and w, and three channels. At the output, each feature map indicates a particular feature representation extracted at all locations on the input, which is obtained via convolving the input feature map with a trainable linear filter (or kernel) and adding a trainable bias parameter. If we denote the input feature map as X, whose convolution filters are determined by the kernel weights W and bias b, then the output feature map is obtained via:

$$f_s(X\,;\,W,\,b) = W *_s X + b, \tag{1}$$

where $*_s$ is the convolution operation with stride s. After each convolutional layer, point-wise nonlinearity (*e.g.*, ReLU) is applied for improving feature representation capability. Additionally, convolutional layers are often followed by some form of non-linear down-sampling (*e.g.*, max pooling). This results in robust feature representation which tolerates small variations in the location of input feature map.

Due to the stride of convolutional and feature pooling layers, the output feature maps are coarse and reduced-resolution. However, for saliency detection, we are more interested in pixel-wise saliency prediction. For upsampling the coarse feature map, multi-layer *deconvolution* (or *backwards convolution*) networks are put on the top of the convolution networks:

$$Y = D_S(F_S(I\,;\,\mathbb{C}_F)\,;\,\mathbb{C}_D), \tag{2}$$

where I is the input image; $F_S(\cdot)$ denotes the output feature map generated by the convolutional layers with total stride of S; $D_S(\cdot)$ denotes the deconvolution layers that upsample the input by a factor of S to ensure the same spatial size of the output Y and the input image I. The deconvolution operation is achieved via reversing the forward and backward passes of corresponding convolution layer. All the parameters $\mathbb{C}$s of convolution and deconvolution layers are learnable.

Finally, on the top of the network, a convolutional layer with a 1×1 kernel is adopted for mapping the feature maps Y into a precise saliency prediction map P through a sigmoid activation unit. We use the sigmoid layer for pred so that each entry in the output has a real value in the range of 0 and 1. Due to the utilization of FCN, the network is allowed to operate on input images of arbitrary sizes, and preserves spatial information. Fig. 2 illustrates the detailed configuration of our deep network for static saliency.

For training, all the parameters $\mathbb{C}$s are learned via minimizing a loss function, which is computed as the errors between the probability map and the ground truth. As demonstrated in [64], the use of an asymmetric weighted loss helps greatly in the case of unbalanced data. Considering the numbers of salient and non-salient pixels are usually imbalanced, we compute a weighted cross-entropy loss. Given a training sample (I, G) consisting of an image I with size $h \times w \times 3$, and groundtruth saliency map $G \in \{0, 1\}^{h \times w}$, the network produces saliency probability map $P \in [0, 1]^{h \times w}$. For any given training sample, the training loss on network prediction P is thus given by

$$L(P, G) = -\sum_{i=1}^{h \times w} (1-\alpha) g_i \log p_i + \alpha (1-g_i) \log(1-p_i), \tag{3}$$

where $g_i \in G$ and $p_i \in P$; α refers to ratio of salient pixels in ground truth G.

We train the proposed architecture in an end-to-end manner. It is commonplace to initialize systems for many of vision tasks with a prefix of a network trained for image classification. This has shown to substantially reduce training time and improve accuracy. During training, our convolutional layers are initialized with the weights in the first five convolutional blocks of VGGNet [16], which was originally trained over 1.3 million images of the ImageNet dataset [9]. The parameters of remaining layers are randomly initialized. Then we train

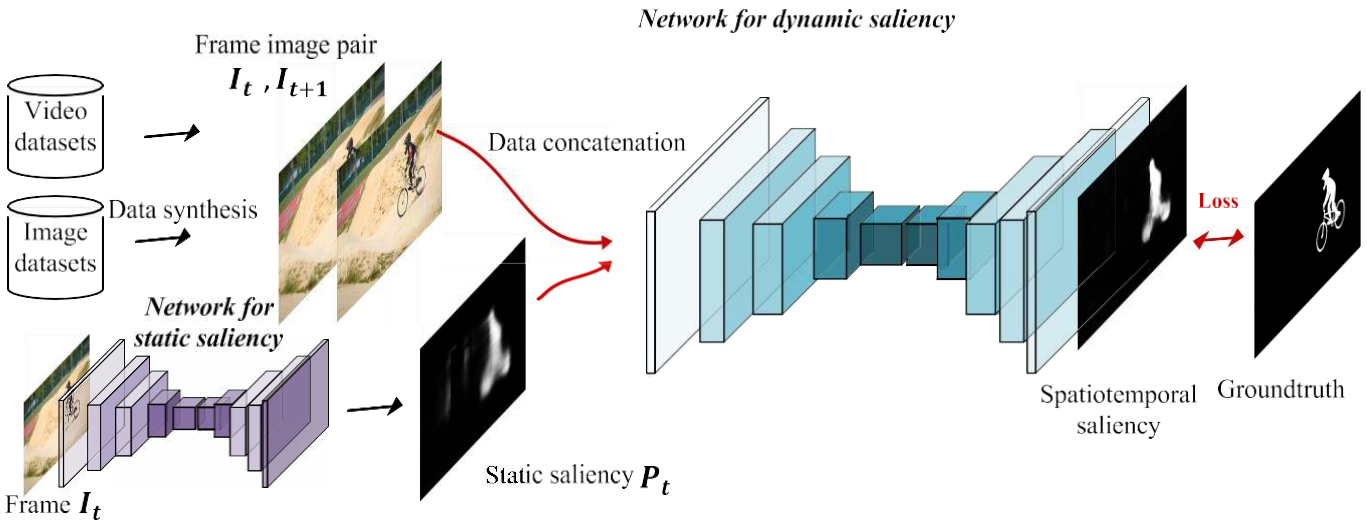


Fig. 3. Illustration of our network for dynamic saliency detection. Successive frame pairs (I_t, I_{t+1}) from real video data or synthesized from existing image datasets (described in Sec. IV), and static saliency information inferred from our static saliency network, are concatenated and fed into the dynamic network, which has a similar FCN architecture with the static network. The dynamic network captures dynamic saliency, and considers static saliency simultaneously, thus directly generating spatiotemporal saliency estimation.

our network with stochastic gradient descent (SGD) using backpropagation by minimizing the loss in (3). More details of implementation are described in Sec. V-A.

C. Deep Networks for Dynamic Saliency

Now we describe our spatiotemporal saliency network. As depicted in Fig. 3, the network has a similar structure as our static saliency network, which is based on FCN and includes multi-layer convolution and deconvolution nets. The dynamic network learns dynamic saliency information jointly with the static saliency results, thus directly generating spatiotemporal saliency estimates.

The training set consists of a collection of synthetic and real video data, which efficiently utilizes existing large-scale well-annotated image data (described in Sec. IV). More specifically, we feed successive pair of frames (I_t, I_{t+1}) and the groundtruth G_t of frame I_t in the training set into this network for capturing dynamic saliency. Meanwhile, since saliency in dynamic scenes is boosted by both static and dynamic saliency information, the network incorporates the saliency estimate P_t generated by static saliency network as saliency priors indicative of potential salient regions. Thus our dynamic saliency network directly generates final spatiotemporal saliency estimates for frame I_t , which is achieved via exploring dynamic saliency cues and leveraging static saliency prior from the static saliency network.

We concatenate frame pair (I_t, I_{t+1}) and static saliency P_t in the channel direction, thus generating a tensor $\mathbf{I}$ with size of $h \times w \times 7$. Then we feed $\mathbf{I}$ into our FCN based dynamic saliency network, which has similar architecture of static saliency network. Only the first convolution layer is modified accordingly:

$$f(\mathbf{I};\ W,\ b) = W_{I_t} * I_t + W_{I_{t+1}} * I_{t+1} + W_{P_t} * P_t + b, \quad (4)$$

where Ws represent corresponding convolution kernels; b is bias parameter. During training, stochastic gradient descent (SGD) is employed to minimize the weighted cross-entropy loss described before. After training, given a frame image pair and static saliency prior, the deep dynamic saliency model is able to output final spatiotemporal saliency estimate. For testing, we first detect the static saliency map P_t for frame I_t via our static saliency network. Then frame image pair (I_t, I_{t+1}) and the static saliency map P_t are fed into the dynamic saliency network for generating the final spatiotemporal saliency for frame I_t . After obtaining the video saliency estimate for frame I_t , we keep iterating this process for the next frame I_{k+1} until reaching the end of the video sequence. More implementation details can be found in Sec. V-A. Qualitative and quantitative study of the effectiveness of our dynamic saliency model is described in Sec. V-C.

Compared with the popular two-stream network structure used in [20], [21], and [57] we merge the output of the static network into the dynamic saliency model, which directly produces spatiotemporal saliency results. This architecture brings two advantages. Firstly, the fusion of dynamic and static saliency is explicitly inserted into the dynamic saliency network, rather than training two-stream networks for spatial and temporal features and specially designing a fusion network for spatial and temporal feature integration. Secondly, the proposed model directly infers the temporal information from two adjacent frames instead of previous methods [20], [56] using optical flow images, thus our model gaining higher computation efficiency.

IV. Synthetic Video Data Generation

So far, we have described our networks for video saliency detection. We discuss our approach for training our networks for dynamic saliency below. As discussed in Sec. I, existing video datasets [10]–[13] are insufficiently diverse and have very limited scales. As deep learning models are data-driven and have strong learning ability, directly learning deep networks on such video datasets would easily suffer overfitting. Noticing the gap between the requirement of learning neural networks for video processing and the lack of large-scale, high-quality annotated video data, we propose a technique for synthesizing video data from still frames.

Directly deriving video sequences from single image is also impossible. However, our video saliency network takes frame pairs as input, instead of the whole video sequence. That means we can simulates diverse but very short video sequences (only 2 frames in length) via fully utilizing well-labelled large-scale image datasets. Concretely, given a training sample (I, G) from existing image saliency datasets, we wish to generate a pair of frames (I, I^{j}), which present various motion patterns, diverse deformations and smooth transformation, thus being close to real video signal. We start at simulating the correspondence between I^{j} and I, which is easier than directly inferring adjacent frame I^{j}. Let $\mathbf{x} = (x, y)$ denote a point position, the correspondence between I^{j} and I can be represented as an optical flow field $\mathbf{v} = (u, v)$ via:

$$I^{j}(\mathbf{x}) = I(\mathbf{x} + \mathbf{v}(\mathbf{x})). \tag{5}$$

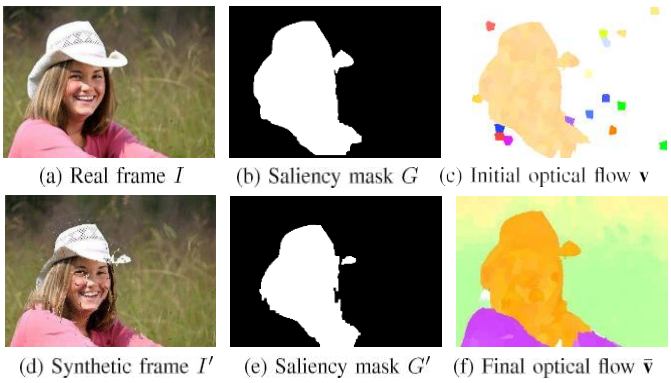
The optical flow field $\mathbf{v}$ directly represents the pixel-level motion information between two neighboring frames. Next we only introduce how to set the vertical displacement u, as the method of generating v is similar.

We model the optical flow on superpixel level [39], [49] as the motion of similar adjacent pixels should present *consistency*. We oversegment I into a group of superpixels $\mathcal{R}$. According to groundtruth label G, we further divide superpixels $\mathcal{R}$ into foreground superpixels $\mathcal{F}$ and background ones $\mathcal{B}$, where $\mathcal{R} = \mathcal{F} \cup \mathcal{B}$. For simulating the *diverse motion patterns* of background, we randomly select 10% background regions $\mathcal{S} \subset \mathcal{B}$ from $\mathcal{B}$ and randomly initialize their motion values us (vertical displacement) from $[-d, d]$, where $d = h/10$. The us of the other background regions are initialized as zero. The motion patterns of foreground are usually compactness, as the whole foreground regions move more regularly and purposefully compared with background. Beside, the motion between different foreground parts sometimes also present diverse. For example, the whole body of a person go an exact direction but his arms or legs may have different motions. For this, we first randomly set a value m (from $[-d, d]$) as the main motion patterns of the foreground regions. Then we randomly set vs of foreground regions from $[m - d/10, m + d/10]$ for representing the difference between foreground regions. This initialization process is visualized in Fig. 4-a.

A similar process is adopted for generating the initial horizontal motion displacement (v) and we are able to get an initial optical flow $\mathbf{v}$ for I. Next, we propose an energy function for smoothing and propagating the initial optical flow globally, yet preserving the difference between foreground and background in motion patterns. Let the initial motion vector of each superpixel $r_i \in \mathcal{R}$ be denoted as $\bar{\mathbf{v}}_i$, the final motion vector $\mathbf{v}_i$ is obtained via optimizing the energy function as follows[2]:

$$E(\mathbf{v}) = \underbrace{\sum_i \lambda_i (\mathbf{v}_i - \bar{\mathbf{v}}_i)^2}_{Unary\ Term} + \underbrace{\sum_{i,j \in \mathcal{N}} w_{i,j} (\mathbf{v}_i - \mathbf{v}_j)^2}_{Smooth\ Term}. \tag{6}$$

The first term is the unary constraint that each superpixel tends to have its initial motion, while the smooth term gives



(a) Real frame I (b) Saliency mask G (c) Initial optical flow $\bar{\mathbf{v}}$

(d) Synthetic frame I' (e) Saliency mask G' (f) Final optical flow $\mathbf{v}$

Fig. 4. Illustration of our synthetic video data generation. A synthetic optical flow filed (c) is first initialized with considering various motion characters in real video sequences. Via (6), final optical flow filed (f) is generated, which is more smooth and better simulates real motion patterns. According to (f), a synthetic frame image I^{j} and its saliency mask G^{j} are warped from (a) and (b), respectively.

the interactive constraint that neighboring superpixels have consistent motion patterns when their representative colors are similar. The superpixel neighborhood set $\mathcal{N}$ contains all the spatially adjacent superpixels.[3] The parameter λ is a positive coefficient measuring how much we want to fit the initial motion. Typically, $\lambda = +\infty$ imposes the hard constraint that each region definitely has the initial motion. We define λ:

$$\lambda_i = \begin{cases} 1 & \text{if } r_i \in \mathcal{F} \\ 1 & \text{if } r_i \in \mathcal{S} \\ 10^{-4} & \text{otherwise} \end{cases} \tag{7}$$

For the seed regions (selected background regions $\mathcal{S}$ and all the foreground regions $\mathcal{F}$), we expect that they tend to preserve their initial motions; however, for other regions $(\mathcal{B} \setminus \mathcal{S})$ we emphasize more influence on the smooth term thus we can propagate the initial motions from those seed regions. The weighting function $w_{i,j}$ in (6) defines a similarity measure for adjacent superpixels $(r_i, r_j \in \mathcal{N})$:

$$w_{i,j} = \begin{cases} exp^{-\|C(r_i) - C(r_j)\|^2} & \text{if } r_i, r_j \in \mathcal{F} \\ exp^{-\|C(r_i) - C(r_j)\|^2} & \text{if } r_i, r_j \in \mathcal{B} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $C(r)$ indicates the mean color vector of pixels in superpixel r. We set the weight $w_{i,j}$ as zero, when two adjacent superpixels are from foreground $\mathcal{F}$ and background $\mathcal{B}$, respectively. We consider motion consistency inside the foreground and background, while preserve motion difference between foreground and background. (6) can be efficiently solved by convex optimization and we can obtain a smooth optical flow field $\mathbf{v}$. As shown in Fig. 4, base on $\mathbf{v}$, we can generate a simulated frame I^{j} and its corresponding annotation G^{j} from (I, G).

The proposed method is very fast and outputs synthesized

[2]Here we slightly reuse $\mathbf{v}$ for representing the optical flow vector of superpixel without ambiguity.

video frame pair, optical flow, and pixel-wise annotations simultaneously. The number of samples in existing image segmentation/saliency datasets is ten or hundred order of

[3]For further encouraging the motion consistency of background regions, we consider all the selected background regions S are adjacent in neighboring system א.

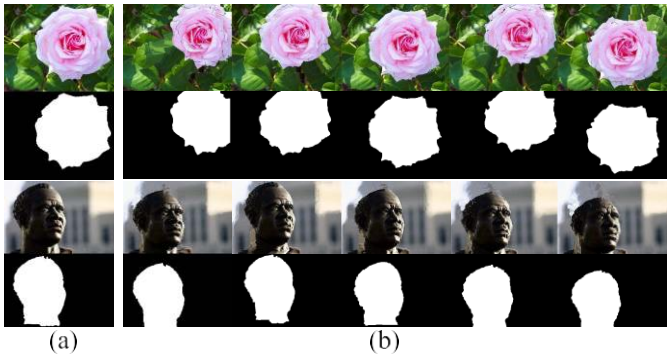


Fig. 5. (a) Real images and corresponding saliency groundtruth masks from existing image datasets. (b) Synthetic image examples and saliency masks generated via our method.

magnitude larger than in the video segmentation datasets, allowing us to generate enough scenes. For each image sample I of an image dataset, we generate ten simulated frames. Some simulated results can be observed in Fig. 5. In our experiments, we use two large image saliency datasets MSRA10K [65] and DUT-OMRON [66], generating more than 150K simulated videos associated with pixel-level annotations and optical flow within 3 hours (processing speed of 14 fps on one CPU). Those synthesized video data, combined with real video samples from existing video segmentation datasets, are fed into our model for learning general dynamic saliency information without over-fitting.

V. Experimental Results

In this section, we describe our evaluation protocol and implementation details (Sec. V-A), provide exhaustive comparison results over two large datasets (80 videos in total, Sec. V-B), study the quantitative importance of the different components of our system (Sec. V-C), and assess its computational load (Sec. V-D).

A. Experimental Setup

1) Datasets: We report our performance on two public benchmark datasets: Freiburg-Berkeley Motion Segmentation (FBMS) dataset [10], and Densely Annotated VIdeo Segmentation (DAVIS) dataset [13]. The FBMS dataset contains 59 natural video sequences, covering various challenges such as large foreground and background appearance variation, significant shape deformation, and large camera motion. This dataset is originally used for motion segmentation, where unsalient but moving objects are also labeled as foreground. We offer more precise annotations for this dataset via only labeling the main salient objects. The FBMS dataset comes with a split into a training set and a test set, where the training set includes 29 video sequences and the test set has 30 video sequences. We also report our performance on the newly developed DAVIS dataset, which is one of the most challenging video segmentation benchmarks. It consists of 50 video sequences in total, and fully-annotated pixel-level segmentation ground-truth for each frame is available. We report the performance of our method and other alternatives on the *test set* of FBMS dataset and the *whole* DAVIS dataset.

For training, we use two large image saliency datasets: MSRA10K [65] and DUT-OMRON [66]. The MSRA10K dataset comprising of 10K images, is widely used for saliency detection and covers a large variety of image contents – natural scenes, animals, indoor, outdoor, etc. Most of the images have a single salient object. The DUT-OMRON dataset is one of the most challenging image saliency datasets and contains 5172 images with multiple objects with complex structures and high background clutter. All the above datasets contain manually annotated groundtruth saliency. The video sequences of the whole SegTrackV2 dataset [11] and the training set of the FBMS dataset are also used for training the dynamic saliency network, which include about 3K frame pairs.[4]

2) Implementation: The proposed deep video saliency network has been implemented with the popular Caffe library [70], an open source framework for CNNs training and testing. For our static video saliency network, the weights of the first five convolutional blocks are initialized by the VGGNet model [16] trained on ImageNet [9], the other convolutional layers are initialized from zero mean Gaussian with a standard deviation of 0.01 and the biases are set to 0. Based on this, our network was trained on the MSRA10K [65] and the DUT-OMRON [66] datasets with 100K iterations for saliency detection in static scenes. Our dynamic video saliency network is also initialized from the VGGNet network. For the first convolutional layer, we use Gaussian initialization due to a different input channel from VGGNet. Benefiting from our video data synthesis approach, we can employ images and annotations from existing saliency segmentation datasets for training our video saliency model. The images and masks from MSRA10K and DUT-OMRON datasets are used to generate more than 150K video slits. Then we combine our simulated video data with real video data (∼3K frame pairs) from exiting video segmentation datasets [10], [11] for generating an aggregate video saliency training set. Our whole video saliency model is trained for 300K iterations. For both two networks, we use stochastic gradient descent (SGD) and a polynomial learning policy with initial learning rate of 10^{-7}. The momentum and weight decay are set to 0.9 and 0.0005.

B. Performance Comparison

To evaluate the quality of the proposed approach, we provide in this section quantitative comparison for performance of the proposed method against various top-performing alternatives: saliency via deep feature (MD) [33], saliency via absorbing markov chain (MC) [67], space-time saliency for time-mapping (TIMP) [19], gradient-flow filed based saliency (GAFL) [3], geodesic distance based video saliency (SAGE) [17], and saliency via random walk with restart (RWRV) [6], on test set (30 video sequences) of the FBMS dataset and the whole DAVIS dataset (50 video clips). The former two methods aim at image saliency while the latter four are designed for video saliency.

[4]Due to the number of annotations provided by FBMS is very limited (only 4∼6 frames are labeled for each video sequence), we provide extra ∼500 annotations.

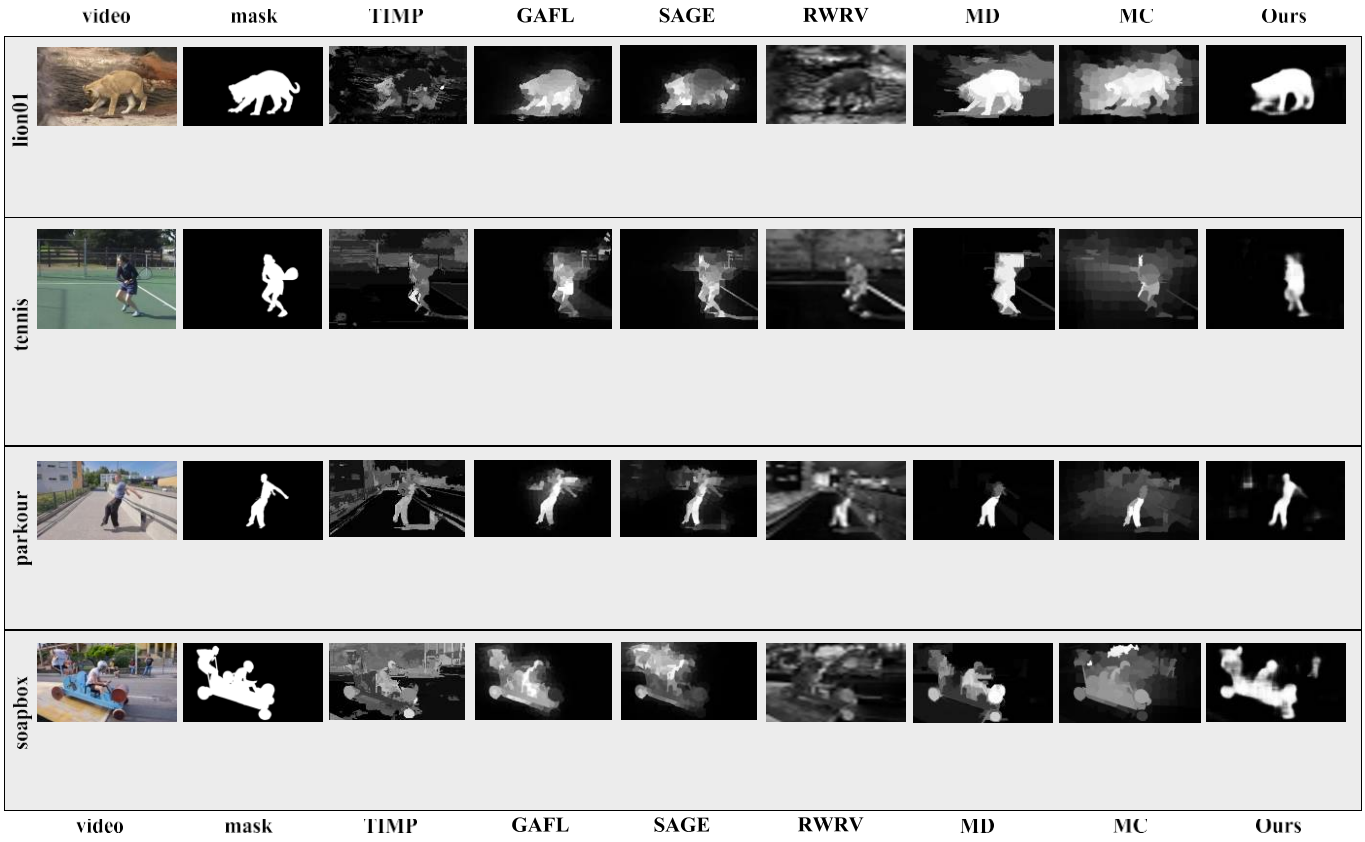


Fig. 6. Qualitative comparison against the state-of-the-art methods on the FBMS dataset [10] (*lion01* and *tennis*), and DAVIS dataset [13] (*parkour* and *soapbox*) with pixel-level ground-truth labels. Our saliency method yields continuous saliency maps that are most similar to the ground-truth.

1) Qualitative Results: Qualitative comparisons are presented in Fig. 6, where the top line shows example video frames and the second line shows the ground truth detection results of salient objects. As seen, the image saliency method [67] without deep learning, unsurprisingly, faces difficulties in dynamic scenes, due to the lack of inter-frame information and utilization of hand-crafted features. The video saliency methods [3], [17] generate more visually promising results, but suffer higher computation load (which will be detailed in Sec. V-D) and show relatively weak performance with complex background. As for [33], it's an image saliency model but exhibits competitive performance with above bottom-up video saliency approaches, which demonstrates the power of deep learning model in saliency detection. However, we can observe the proposed algorithm captures foreground salient objects more faithfully in most test cases. In particular, the proposed algorithm yields good performance on some challenging scenarios, even for blurred backgrounds (*lion01*), various object motion patterns (*parkour*) or large shape deformation (*soapbox*). This can be attributed to our video data synthesis, which offers diverse scene information and rich motion patterns. Based on this, our method is able to learn both static and dynamic saliency information and detects salient moving objects accurately despite similar appearance to the background.

2) Quantitative Results: We report quantitative evaluation results on three widely used performance measures: precision-recall (PR) curves, F-measure and MAE.

We first employ precision-recall (PR) curves for performance evaluation. Precision corresponds to the percentage of salient pixels correctly assigned, while recall corresponds to the fraction of detected salient pixels in relation to the ground truth number of salient pixels. For each saliency map, we vary the cutoff threshold from 0 to 255 to generate 256 precision and recall pairs, which are used to plot a PR curve.

The F-measure is the overall performance measurement computed by the weighted harmonic of precision and recall:

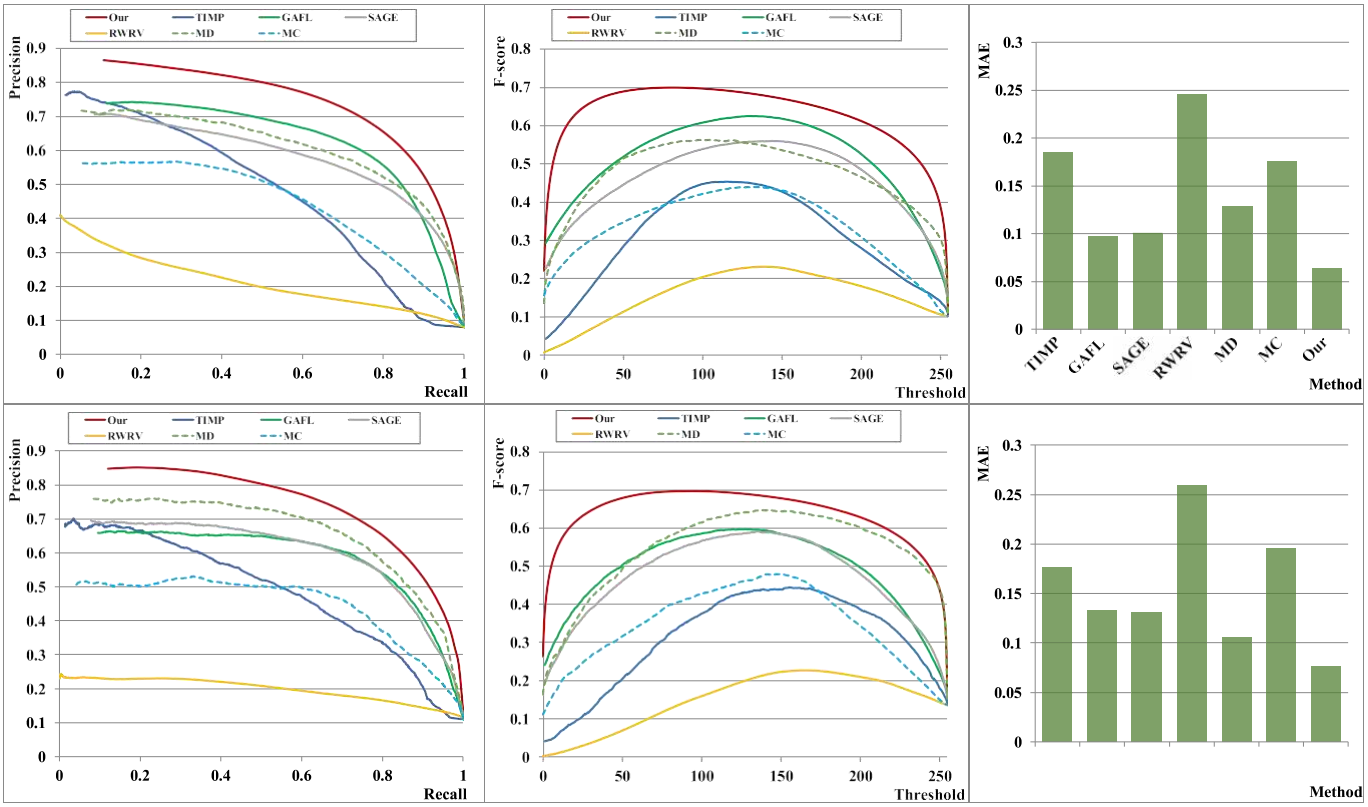
$$\text{F-measure} = \frac{(1 + \beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}, \tag{9}$$

where we set $\beta^2 = 0.3$ to weigh precision more than recall as suggested in [71]. For each saliency map, we derive a sequence of F-measure values along the PR-curve with the threshold varying from 0 to 255.

As neither precision nor recall considers the true negative saliency assignments, the mean absolute error (MAE) is also introduced as a complementary measure. MAE is defined as the average per-pixel difference between an estimated saliency probability map P and its corresponding ground truth G. Here, P and G are normalized to the interval [0, 1]. MAE is computed as

$$\text{MAE} = \frac{\sum_{i=1}^{h \times w} |P(\mathbf{x}_i) - G(\mathbf{x}_i)|}{h \times w}, \tag{10}$$

where h and w refer to the height and width of the input frame image. MAE is meaningful in evaluating the



(a) (b) (c)

Fig. 7. Comparison with 8 alternative saliency detection methods using the DAVIS dataset [13] (**top**), and the test set of the FBMS dataset [10] (**bottom**) with pixel-level ground-truth: (a) average precision recall curve by segmenting saliency maps using fixed thresholds, (b) F-score, (c) average MAE. Notice that, our algorithm consistently outperforms other methods across different metrics.

applicability of a saliency model in a task such as object segmentation.

The precision-recall curves of all methods are reported in Fig. 7-a. As shown, our method significantly outperforms the state-of-the-art both on the FBMS dataset [10], and the DAVIS dataset [13]. Our saliency method achieves the best precision rates, which demonstrates our saliency maps are more precise and responsive to the actual salient information. The F-scores are depicted in Fig. 7-b, in which our model achieves better scores than other methods. Similar conclusions can be drawn from the MAE. In Fig. 7-c, our method achieves the lowest MAE among all compared methods.

C. Validation of the Proposed Method

To exhibit more details of our algorithm and objectively evaluate the contribution of different phases in the proposed saliency model, we report the evaluation of each of the components described in Sec. III and different variants of the proposed saliency model. We experiment on the test set of the FBMS dataset [10], and the DAVIS dataset [13] and measure the performance using precision recall curve and MAE.

1) Ablation Study: We first study the effect of each module of our deep saliency model. In Fig. 8, we present qualitative comparison between static saliency from our static network (in Sec. III-B) and final spatiotemporal saliency results from our whole model (in Sec. III-C). It can be observed, due to the lack of dynamic information, the static saliency model faces difficulties distinguishing salient objects from clutter background in dynamic scenes. Via comprehensively utilizing static and dynamic saliency stimuli, our deep video saliency model is able to estimate more accurate spatiotemporal saliency maps. For quantitatively examining the performance of our static saliency network, we directly use the static saliency maps generated by the static network as final saliency estimates. From Table II, we can observe decreased performance (7.65→ 8.19 on FBMS, 6.36→ 7.17 on DAVIS), due to the lack of dynamic saliency information. Similarly, we train a dynamic network without considering static saliency as prior using the same training data. We attribute this to the difficulty of directly capturing dynamic saliency information from two successive frames without any saliency prior or extra motion information. We can draw two important conclusions. First, the fusion of static model and dynamic model improves on both. Second, taking static saliency as prior information makes training the dynamic model easier and yield more accurate prediction.

2) Training Strategy: We also explore the effect of different training strategies. We first study the influence of our synthetic video data generation strategy in Sec. IV. We train our deep saliency model only using the synthetics from image data. Although the real video data occupy a small percentage of the training, we can still see a decrease in MAE (7.65→9.27 on FBMS, 6.36→7.53 on DAVIS) when we only use synthetic data. The small performance decrease verifies the effectiveness

TABLE II

ASSESSMENT OF INDIVIDUAL MODULES AND VARIANTS OF OUR DEEP SALIENCY MODEL ON THE TEST SET OF FBMS DATASET [10] AND THE DAVIS DATASET [13] USING MAE. LOWER VALUES ARE BETTER

aspect	variant	FBMS		DAVIS	
		MAE(%)	ΔMAE(%)	MAE(%)	ΔMAE(%)
	whole model	**7.65**	-	**6.36**	-
module	Static model in Sec. III-B	8.19	+0.54	7.17	+0.81
	Dynamic model in Sec. III-C	9.43	+1.78	8.32	+1.96
Training	Training set *i*: only using image data (1.50×10^5)	9.27	+1.62	7.53	+1.17
	Training set *ii*: only using video data (0.03×10^5)	24.5	+16.8	23.9	+17.5
	Training set *iii*: reduced training data (1.00×10^5)	9.14	+1.48	7.54	+1.18
	Training set *iv*: reduced training data (0.50×10^5)	10.7	+3.08	9.13	+2.77
	Training set *v*: reduced training data (0.10×10^5)	12.8	+5.18	10.9	+4.58
	Training set *vi*: reduced training data (0.05×10^5)	13.5	+5.83	12.7	+6.39

Fig. 8. Qualitative comparison between our static saliency results and final spatiotemporal saliency results. From top to bottom: input frame images, saliency results via our static saliency network, and spatiotemporal saliency results via our whole video saliency model.

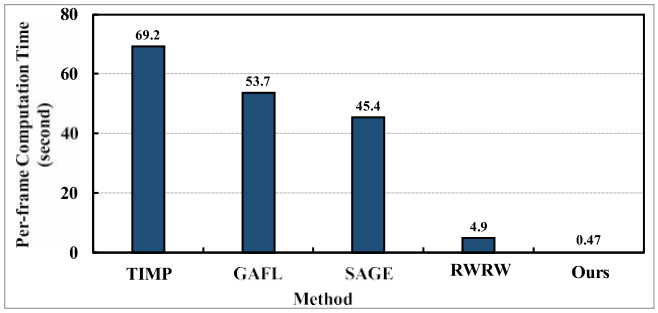


Fig. 9. Computational load of our method and the state-of-the-art video saliency methods for processing a 480p video.

of our data augmentation technique; on the other hand, it suggests the synthetics should not completely replace the real video data. We further explore the performance of our model only using video data (0.03×10^5 frame pairs). Unfortunately, our model suffers over-fitting due to the high similarities of scenes within same video. This also demonstrates the importance of our synthetic video data generation.

D. Runtime Analysis

Here we consider the speed of our saliency method. Our computing platform includes Intel Xeon E7 CPU (12 cores) with 64 GB memory and Nvidia Geforce TITAN X GPU. We do not count I/O time, and do not allow processing multiple images in parallel. The time consumption, of our method compared against other video saliency methods [3], [6], [17], [19] are presented in Fig. 9. From Fig. 9 we can learn that, run time efficiency is the major bottleneck for the usability of previous video saliency algorithms, as a substantial amount of time is spent computing motion or edge information. In contrast, our method computes 480p saliency masks in as little as 0.47 seconds, which is much faster than traditional video saliency methods. Our method does not rely on optical flow, edge maps or other pre-computed information, resulting in roughly an order of magnitude faster processing speed.

VI. CONCLUSION

In this work, we have presented a deep learning method for fast video saliency detection using convolutional neural networks. The proposed deep video saliency model has two modules, namely static saliency network and dynamic saliency network, which are designed for capturing spatial and temporal statistics of dynamic scenes. The saliency estimates from the static saliency network is incorporated in the dynamic saliency network, which enables our method to automatically learn the way of fusing static saliency into dynamic saliency detection and directly produce final spatiotemporal saliency results with less computation load. Furthermore, we proposed a novel data augmentation technique for synthesizing video data from still images, which enables our deep saliency model to learn generic spatial and temporal saliency and prevents overfitting. Experimental results have shown that our methods generate high-quality salience maps. Additionally, our model is very efficient with a frame rate of 2fps on a GPU.

REFERENCES

[1] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[2] Y. Fang, Z. Wang, W. Lin, and Z. Fang, "Video saliency incorporating spatiotemporal cues and uncertainty weighting," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3910–3921, Sep. 2014.

[3] W. Wang, J. Shen, and L. Shao, "Consistent video saliency using local gradient flow optimization and global refinement," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4185–4196, Nov. 2015.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[6] H. Kim, Y. Kim, J.-Y. Sim, and C.-S. Kim, "Spatiotemporal saliency detection for video sequences based on random walk with restart," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2552–2564, Aug. 2015.

[7] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1265–1274.

[8] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 478–487.

[9] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[10] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 282–295.

[11] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2192–2199.

[12] F. Galasso, N. S. Nagaraja, T. Cardenas, T. Brox, and B. Schiele, "A unified video segmentation benchmark: Annotation, metrics and analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3527–3534.

[13] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 724–732.

[14] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognit. Psychol.*, vol. 12, no. 1, pp. 97–136, Jan. 1980.

[15] P. Mital, T. J. Smith, S. Luke, and J. Henderson, "Do low-level visual features have a causal influence on gaze during dynamic scene viewing?" *J. Vis.*, vol. 13, no. 9, p. 144, 2013.

[16] K. Simonyan and A. Zisserman. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[17] W. Wang, J. Shen, and F. Porikli, "Saliency-aware geodesic video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3395–3402.

[18] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[19] F. Zhou, S. B. Kang, and M. F. Cohen, "Time-mapping using space-time saliency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3358–3365.

[20] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[21] C. C. Bak, A. Erdem, and E. Erdem. (Jul. 2016). "Two-stream convolutional networks for dynamic saliency prediction." [Online]. Available: https://arxiv.org/abs/1607.04730

[22] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. (Dec. 2016). "Learning video object segmentation from static images." [Online]. Available: https://arxiv.org/abs/1612.02646

[23] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[24] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 545–552.

[25] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2009, pp. 2106–2113.

[26] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 733–740.

[27] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[28] W. Wang, J. Shen, L. Shao, and F. Porikli, "Correspondence driven saliency transfer," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5025–5034, Nov. 2016.

[29] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.

[30] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Dec. 2015.

[31] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 29–42.

[32] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2814–2821.

[33] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5455–5463.

[34] Y. Tang and X. Wu, "Saliency detection via combining region-level and pixel-level predictions with CNNs," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 809–825.

[35] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3183–3192.

[36] N. Liu and J. Han, "DHSnet: Deep hierarchical saliency network for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 678–686.

[37] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 825–841.

[38] X. Li *et al.*, "DeepSaliency: Multi-task deep neural network model for salient object detection," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3919–3930, Aug. 2016.

[39] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao, "Real-time superpixel segmentation by DBSCAN clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5933–5942, Dec. 2016.

[40] D. Zhang, J. Han, L. Jiang, S. Ye, and X. Chang, "Revealing event saliency in unconstrained video collection," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1746–1758, Apr. 2017.

[41] W. Wang, J. Shen, X. Li, and F. Porikli, "Robust video object cosegmentation," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3137–3148, Oct. 2015.

[42] W. Wang and J. Shen, "Higher-order image co-segmentation," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1011–1021, Jun. 2016.

[43] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 865–878, May 2017.

[44] B. X. Nie, P. Wei, and S.-C. Zhu, "Monocular 3D human pose estimation by predicting depth on joints," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3447–3455.

[45] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 215–232, 2016.

[46] S.-H. Zhong, Y. Liu, F. Ren, J. Zhang, and T. Ren, "Video saliency detection via dynamic consistent spatio-temporal attention modelling," in *Proc. AAAI Conf. Artif. Intell.*, 2013, pp. 1063–1069.

[47] W. Wang, J. Shen, H. Sun, and L. Shao, "ViCoS2: Video co-saliency guided co-segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[48] D. Zhang, J. Han, J. Han, and L. Shao, "Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1163–1176, Jun. 2016.

[49] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451–1462, Apr. 2014.

[50] X. Lu, Y. Yuan, and X. Zheng, "Joint dictionary learning for multispectral change detection," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 884–897, Apr. 2017.

[51] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[52] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[53] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *J. Vis.*, vol. 9, no. 12, p. 15, 2009.
[54] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171–177, Jan. 2010.
[55] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman, "Personalizing human video pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3063–3072.
[56] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik, "Learning to segment moving objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4083–4090.
[57] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.
[58] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3119–3127.
[59] W. Wang, J. Shen, Y. Yu, and K.-L. Ma, "Stereoscopic thumbnail creation via efficient stereo saliency detection," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 8, pp. 2014–2027, Aug. 2017.
[60] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 809–817.
[61] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.
[62] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.
[63] Y.-H. Tsai, G. Zhong, and M.-H. Yang, "Semantic co-segmentation in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 760–775.
[64] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3376–3385.
[65] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[66] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3166–3173.
[67] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency detection via absorbing Markov chain," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1665–1672.
[68] W. Wang, J. Shen, J. Xie, and F. Porikli, "Super-trajecotry for video segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, to be published.
[69] W. Wang and J. Shen, "Deep cropping via attention box prediction and aesthetics assessment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2186–2194.
[70] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[71] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1597–1604.

Wenguan Wang received the B.S. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree with the School of Computer Science. His current research interests include deep learning for salient object segmentation in videos.

Jianbing Shen (M'11–SM'12) is a Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. He has authored or co-authored about 60 journal and conference papers, such as the IEEE TPAMI, the IEEE TIP, the IEEE TVCG, the IEEE CVPR, and the IEEE ICCV. His research interests include computer vision and multimedia processing. He serves as an Associate Editor of *Neurocomputing*.

Ling Shao (M'09–SM'10) received the B.Eng. degree in electronic and information engineering from the University of Science and Technology of China, and the M.Sc. degree in medical image analysis and the Ph.D. (D.Phil.) degree in computer vision from the University of Oxford. He is a Professor with the School of Computing Sciences, University of East Anglia, Norwich, U.K. His research interests include computer vision, image/video processing, pattern recognition, and machine learning. He is a fellow of the British Computer Society, a fellow of the IET, and a Life Member of the ACM. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and other journals.