Multimed Tools Appl
https://doi.org/10.1007/s11042-018-5953-1

Foveated convolutional neural networks for video summarization

Jiaxin Wu[1] · Sheng-hua Zhong[1] · Zheng Ma[2] · Stephen J. Heinen[2] · Jianmin Jiang[1]

Received: 12 December 2017 / Revised: 11 March 2018 / Accepted: 27 March 2018

Abstract With the proliferation of video data, video summarization is an ideal tool for users to browse video content rapidly. In this paper, we propose a novel foveated convolutional neural networks for dynamic video summarization. We are the first to integrate gaze information into a deep learning network for video summarization. Foveated images are constructed based on subjects' eye movements to represent the spatial information of the input video. Multi-frame motion vectors are stacked across several adjacent frames to convey the motion clues. To evaluate the proposed method, experiments are conducted on two video summarization benchmark datasets. The experimental results validate the effectiveness of the gaze information for video summarization despite the fact that the eye movements are collected from different subjects from those who generated

Jiaxin Wu and Sheng-hua Zhong contributed equally to this work.

✉ Jianmin Jiang
jianmin.jiang@szu.edu.cn

Jiaxin Wu
jiaxin.wu@email.szu.edu.cn

Sheng-hua Zhong
csshzhong@szu.edu.cn

Zheng Ma
zma@ski.org

Stephen J. Heinen
heinen@ski.org

[1] The College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong, China

[2] Smith-Kettlewell Eye Research Institute, San Francisco, CA, USA

Published online: 30 April 2018

summaries. Empirical validations also demonstrate that our proposed foveated convolutional neural networks for video summarization can achieve state-of-the-art performances on these benchmark datasets.

Keywords Video summarization · Convolutional neural networks · Eye movement · Foveated image

1 Introduction

As video data grows explosively in recent years, browsing such a huge quantity of videos is time-consuming and tedious. As a matter of fact, video summarization is an ideal tool for people to watch the video in a rapid way, which provides a compact form of the input video [10, 33, 34]. Generally, video summarization can be divided into two groups: static video summarization and dynamic video summarization [34]. Static video summarization provides a static storyboard comprising of representative individual frames while dynamic video summarization is a kind of video skimming, which is consist of attracted and brief video shots. In this paper, we proposed a dynamic video summarization model.

Recently, many research work have been proposed to tackle the dynamic video summarization task. At the very beginning, many methods were proposed based on low-level features [18, 33]. Li et al. used the scale-invariant feature transform descriptor [20] to measure the similarity of two frames [18]. Then, they selected final summary based on the improved maximal marginal relevance algorithm. Song et al. concatenated four classical features: a pyramid of HoG [1], GIST [25], SIFT [20] and color histogram [41] to represent the video frame [33]. They tried to generate the final summary by using title-based image search results based on the fusion features. Later, some dynamic video summarization methods focused on mid-level semantic features [10, 21]. Ma et al. proposed an audio-visual computational attention model for video summarization [21]. They constructed the user attention model by integrating visual, audio and linguistic information of the video. Finally, the video summarization result was generated based on a user attention curve. Gygli et al. computed the interestingness score for each frame based on a combination of some features such as spatial, temporal saliency and detecting faces [10]. Then the model selected an optimal subset of video shots to create an informative summary based on the interestingness scores. In recent years, many researchers started to develop dynamic video summarization models relied on deep learning features [22, 43, 48]. Yao et al. proposed a pairwise ranking two-stream deep model for video summarization [43]. They utilized the appearances of video frames as spatial information and temporal dynamics across frames to represent each video. At the end, they selected the highlight segments based on the highlight scores. Zhang et al. devised their model based on long short-term memory (LSTM) architecture [48], which was used to model the variable-range temporal dependency among video frames. The experimental results showed that the proposed sequential structure was effective to create a meaningful video summary. Moreover, Mahasseni et al. proposed [22] a generative adversarial neural network (GAN) for video summarization. They conducted the experiments on several benchmark datasets and the experimental results demonstrated the proposed method achieved competitive performance in comparison to the state-of-the-art methods.

Deep learning networks have achieved great success in computer vision [5, 31, 50]. In this paper, we propose our dynamic video summarization model via a two-stream

convolutional neural networks. The spatial-temporal architecture has been demonstrated to be an effective network for action recognition task [31, 37] and video summarization task [43]. In the spatial stream, the previous work tended to use RGB images to represent the visual cues [31, 37, 43]. And in the temporal stream, most of the deep learning methods for video content analysis used optical flow [31] or dense trajectories [37] to convey the motion information. Besides the spatial and temporal information, eye movements directly tracked from users are also useful for video related task. In fact, eye movement indicates a strong sense of user's interest and the importance of content [44]. It also reflexes how a person's attention is distributed in the spatial and temporal dimensions. Previous work have succeeded in demonstrating the effectiveness of gaze in many computer vision tasks, such as images classification [16], activity recognition [49], object detection [44] and so on. Although eye movement is thought as an important indicator of user's interest, only a few studies for video summarization tried to integrate this information [30, 42]. Xu et al. proposed a gaze-enabled video summarization model for egocentric videos [42]. They first divided the video into subshots. Then, they used aggregate fixation counts to measure the attention score of each shot and extracted features around the gaze region by using R-CNN for each subshot keyframe. Finally, they formulated the video summarization task as a submodular function maximization problem by selecting an optimal subset to obtain the final summary. They conducted the experiments on two egocentric datasets with gaze data. The experimental results showed the proposed model could generate good summaries for egocentric videos. Salehin et al. designed a framework based on the smooth pursuit to detect important events for the input video [30]. They first proposed a method to distinguish smooth pursuit from other types of eye movement: fixation and saccade. Later, they used smooth pursuit information to calculate the important score for each frame. At the end, they generated the final summaries based on these important scores. The experiments were conducted on Office video dataset [8], which contains videos with camera movement/shaking and illumination changes. They collected the eye tracker data for each video of this Office dataset. The experimental results showed that the proposed method could achieve a satisfactory summary result for a video with camera movement, low contrast, and significant illumination changes. However, both of these existing approaches tended to directly utilize eye movement information alone. They did not incorporate the gaze information with existing sources.

Instead, we try to combine eye movement with video content by imitating the visual processing in human cortex. We proposed a foveated two-stream ConvNets for video summarization (FVS). In the spatial stream, instead of using RGB images directly, we construct foveated images based on subjects' gaze information as the input of the spatial channel for the networks. In the temporal stream, we replace optical flow (or dense trajectories) with motion vectors which can be extracted from the compressed video directly. To the best of our knowledge, we are the first to integrate eye movement into deep learning architecture for video summarization. We apply our proposed model on two dynamic video summarization benchmark datasets (SumMe [10] and TVSum [33]). The experimental results confirm the effectiveness of the foveated images and motion vectors.

The rest of this paper is organized as follows. Section 2 briefly reviews the applications of eye movement. In Section 3, we illustrate our proposed dynamic video summarization model via two-stream architecture. In Section 4, we conduct several experiments on two standard datasets. Finally, we will conclude our method and talk about the future work in Section 5.

2 Related work

Eye movement is a very important cue to indicate a person's interest and purpose [2, 24, 44]. In other words, eye movement always shows how a person's attention is distributed in the spatial and temporal dimensions [44]. We can find out what is attracting people and the relative importance of content by using gaze information.

Gaze information plays an important role in various tasks. Pereira et al. tried to analyze the eye movement patterns to detect Alzheimer's disease (AD) [27]. They found that AD patients tended to have increased latencies in reflexive saccades and altered saccadic inhibition, which might suggest impairments in executive functioning. Holmberg et al. tried to use eye movement data to find out the relation of children's visual attention on advertisements and advert saliency features [13]. They recorded the gaze data of children when they were surfing their favorite websites. The experimental results showed that all low-level saliency features such as motion, luminance and edge density had effects on children' s visual attention, but these effects relied on children' s individual level of gaze control. Meanwhile, there are lots of gaze-related work on computer vision. Mishra et al. [23] proposed a novel object segmentation approach based on gaze information. They utilized fixation points to find an optimal closed contour for the object of interest. Their results showed that, with the help of gaze data, the proposed method could make promising segmentation performances. Recently, Karessli et al. tackled zero-shot image classification based on gaze information [16]. They introduced three kinds of gaze embedding features, including gaze histograms, gaze features with grid and gaze features with the sequence. They conducted their experiments on two gaze-annotated image classification datasets. The results demonstrated that human eye movement data was indeed class discriminative and could be a competitive alternative to the expert annotation.

There is also another kind of gaze-enabled task. Instead of using eye movement information alone, they tended to combine gaze information with images contents by constructing foveated images [9, 36]. Foveated image has spatially varying resolution according to one or more fixation points [39]. The gaze regions have the higher resolution corresponding to the center of the eye' retina (the fovea) while the rest of the image tend to be blurred. Guenter et al. proposed a foveated 3D graphics model to accelerate graphics computation [9]. They tracked the participant' s gaze point and rendered three images layers around it at progressively higher angular size but lower sampling rate. The experiments showed that the proposed foveated rendering method improved graphics performance and achieved competitive performance to standard rendering. Wang et al. introduced a novel image quality measurement based on the foveated image in the wavelet transform domain [36]. They suggested that the proposed foveated image quality metric could be used for image coding and quality enhancement. They applied the proposed method for a foveated image coding system. The results showed that it could have a good coding performance.

3 Video summarization based on foveated two-stream ConvNets

As we know, video can be naturally separated into spatial and temporal parts [31]. The spatial stream treats individual frame appearance as input, which carries information about scenes and objects contained in the video. The temporal stream uses motion across the frames as input, which conveys the movements of the objects and the camera. Thus, we design our video summarization model based on the two-stream architecture. Figure 1 outlines

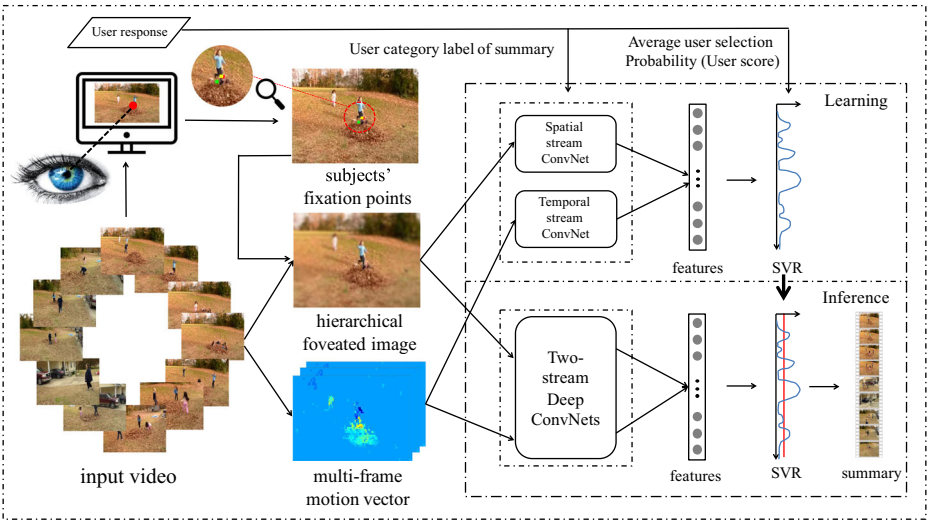


Fig. 1 Overview of the proposed foveated two-stream ConvNets for video summarization

the overview of our proposed model. In the learning stage, we first construct a two-class classification model based on a foveated two-stream deep ConvNets. Then, the training data with their corresponding user category labels are input to train a discriminative two-stream ConvNets. Next, the combined features extracted from the ConvNets with their corresponding summary probabilities are input to support vector regression (SVR) to train an effective regression model. Finally, the regression model is used to predict the highlight score/probability for each frame. In the inference stage, firstly, the learnt ConvNets are used to extract features from the input test data. Secondly, the learnt SVR is utilized to predict the highlight score for each frame based on the combined feature. Lastly, the final summary is generated based on their predicted scores/probabilities.

In the following, we first describe how we construct foveated RGB images based on subject's fixation points for the spatial stream. Next, we introduce how to build up multi-frame motion vectors for the temporal stream. Finally, we will talk about the video summarization generation based on our proposed foveated two-stream deep ConvNets. Besides, to distinguish two groups of subjects/users in this paper, we use different names to represent them. As we know, the publishers of SumMe [10] and TVSum [33] datasets have provided multiple summaries for each video. In SumMe, Gygli et al. asked at least 15 participants to summarize the video content for each video [10]. For each video in TVSum, Song et al. asked 20 participants to generate their summarizations for the video [33]. To the persons who provided their personal summaries for videos in SumMe and TVSum datasets, we call them "users" in the following paragraphs and sections. We recruited several subjects to collect their gaze data for each video in these standard datasets. We use "subjects" or "subject" with their individual ID number to represent them.

3.1 Foveated images construction

In this paper, we propose to use foveated images to represent the visual appearance of video frames as the input of the spatial stream. It is well known that people have noisy

representations of spatial locations [4]. When the eye is observing a visual stimulus (such as a still image or a video clip), only the fixation region is perceived by the human visual system with maximum resolution, and the perceived resolution decreases progressively for regions that are projected away from the fovea. Rovamo et al. suggested that the spatial resolution at a certain eccentricity location could be predicted accurately by the following equation $\sigma_z^2(E) = c \times (1 + 0.42 \times E)$. The standard deviation of a two-dimensional Gaussian distribution centered at eccentricity E could also be predicted by the above formulation [29]. E is the visual angle of the current location and the fixation point. As a matter of fact, foveated imaging simulates the visual process of the optical system of the eye. The foveated image has spatially various resolution according to one or more fixation points [39]. The gaze regions have higher resolution corresponding to the center of the eye' retina (the fovea) while the rest of the image have relatively lower resolution. We believe that constructing foveated RGB images based on subjects' eye movements as the input of our model can well convey the users' interests to the current video.

The foveated image is generated by the convolution of the input video frame. The convolutional kernel size depends on subjects' eye movements. The foveated image can be seen as a hierarchical blur version of the input video frame. We develop a foveated 2-D convolution algorithm for the input video frame to generate the corresponding foveated image. Generally, the foveated image can be divided into two regions (gaze regions and non-gaze regions) depending on their distances with the fixation points. In the following, we will describe how we construct a foveated image for the input video frame. Firstly, we split the foveated image into gaze regions and non-gaze regions. For each fixation point, we construct a 2-D Gaussian distribution $W \sim N(1, 0)$ centered in it. Then we define the non-zero value areas of these 2-D distributions as the gaze regions. The rest of areas are defined as non-gaze regions. Secondly, we assign the convolutional result values to the foveated image. Actually, each value is the convolutional result of the neighbor region ($Z \in \mathbb{Z}^{s \times s}$) around the corresponding point in the input frame and a Gaussian kernel map ($M \in \mathbb{R}^{s \times s}$). For s and M, we have different strategies for different regions. For the gaze regions, the convolution kernel size s and the values of the kernel map M are equal to 1, which means that the gaze regions keep the original values of the input video frame. For the non-gaze regions, we define an adaptive kernel size s as follow:

$$s(d) = 2 \times \lceil \sigma(d) \rceil - 1, \tag{1}$$

$$\sigma(d) = c \times \left(1 + 0.42 \times \frac{d}{p}\right). \tag{2}$$

where d is the distance between the current point (x, y) and the subjects' gaze points $G(u, v)$. p is a constant value, representing pixel per degree in the experiment. The kernel size s for the non-gaze region depends on the distance d of the current point and the fixation points. This definition is based on the predicted spatial resolution formulation proposed by Rovamo et al. [29]. We follow the existing work [35] to set the parameter $c = 0.08$. For the Gaussian kernel map M in the non-gaze regions, we first construct a $s \times s$ Gaussian distribution $W \sim N(\sigma(d), 0)$. Then M is equal to the normalized W. Finally, each pixel value of the foveated image is equal to the sum of convolution results of the Z and M.

The detailed procedure of the proposed foveated 2-D convolution algorithm is described in Algorithm 1. On average, it costs about 0.12 ± 0.028 seconds to generate a foveated

image (320×240). The experiment is conducted on a Linux server with 48 Intel(R) Xeon(R) E5-2690 2.60GH CPUs and 256GB RAM.

Algorithm 1 Foveated 2-D convolution algorithm for the input video frame.

Input:

The input video frame, $I(x, y) \in \mathbb{Z}^{w \times h}$, w is the width and h is the height of input frame; Subjects' gaze points in the input frame, $G(u, v) \in \mathbb{R}^{\alpha \times 2}$, α is the number of subjects;

Output:

The foveated image for the input frame, $J(x, y) \in \mathbb{Z}^{w \times h}$;

for each point (x, y) **do**
  if (x, y) is in the gaze regions **then**
    $J(x, y) = I(x, y)$;
  else
    for each subject i **do**
      Calculate the temporary distance q between the current point (x, y) and the gaze point $G(u_i, v_i)$ of subject i , $q(i) = \sqrt{(x - u_i)^2 + (y - v_i)^2}$.
    end for
    The final distance d of current point (x, y) is equal to the minimum value of q, $d = min(q)$.
    Compute the σ, $\sigma(d) = 0.08 \times (1 + 0.42 \times \frac{d}{p})$, p is constant value, which denotes pixel per degree in the experiment.
    Calculate the kernel size s, $s(d) = 2 \times \lceil \sigma(d) \rceil - 1$.
    Extract a $s \times s$ region Z surrounding point (x, y) from the input frame I, $Z = I(x - \frac{s-1}{2} : x + \frac{s-1}{2}, y - \frac{s-1}{2} : y + \frac{s-1}{2})$
    Construct a $s \times s$ 2-D Gaussian kernel map M, $W(x, y) = \frac{1}{2\pi\sigma(d)^2} e^{-\frac{x^2+y^2}{2\sigma(d)^2}}$, $x, y \in [-\frac{s-1}{2}, \frac{s-1}{2}]$, $M = \text{norm}(W)$.
    Sum up the convolution results between Z and M. $conv = Z * M$; $J(x, y) = sum(conv(:))$;
  end if
end for
return $J(x, y)$;

Here, we also show two groups of sample foveated images in Fig. 2. The first group is with gaze locations which are separated, and the second group is with gaze locations which are gathered. Figure 2a and c show the original RGB along with three subjects' gaze points represented as different colored dots (red, green and blue). And Fig. 2b and d are the foveated images generated based on Fig. 2a and c respectively. It is clear that the gaze regions have higher resolution while the rest of the regions tend to be blurred. For example, in Fig. 2a, three subjects are looking at three different locations and it results in three gaze regions in Fig. 2b are relatively clear and the rest regions like the top-right mountain of Fig. 2b are blurred. In addition, the resolutions of regions decrease when they are farther away from fixation points. For example, in Fig. 2c, three subjects are all looking at the girl in pink in the foreground. Thus, in its resulting foveated image Fig. 2d, the resolution of the girl in light blue is higher than that of the boy in black because of the relatively near location from the girl in pink (fixation points).

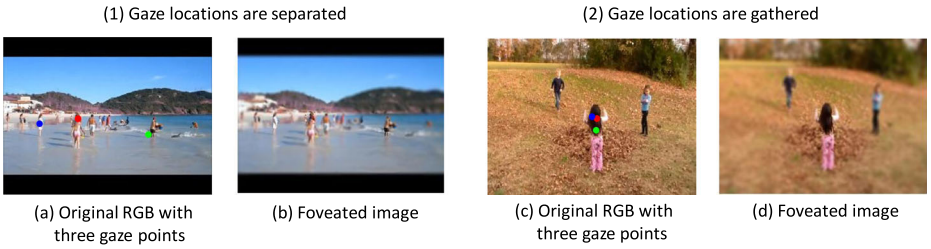


Fig. 2 Samples of original RGB images (along with fixations points) and their corresponding foveated images

3.2 Multi-frame motion vectors construction

In this paper, we construct multi-frame motion vectors to express the movement information for the temporal stream. Motion vectors are typically utilized by video compression to storage the changes to an image from one frame to the next. It represents the movement of the block, which is usually a 16×16 pixel region. Motion vectors are a key element for motion estimation. Although motion vectors are not as accurate as the optical flow, they have been proved to contain useful movement information for action recognition [15]. Besides, as motion vectors are already calculated and encoded in compressed videos, we can directly extract them at very low computational cost.

In this section, we describe how to stack multiple motion vectors to express the movement across several consecutive frames. We assume that the motion vectors of frame t is denoted as $\mathbf{D}_t$. It can be divided into horizontal and vertical components ($\mathbf{D}_\mathbf{t}^\mathbf{x}$ and $\mathbf{D}_\mathbf{t}^\mathbf{y}$). $\mathbf{D}_\mathbf{t}^\mathbf{x}$ represents the displacement between the pairs of adjacent frames t and $t+1$ in horizontal dimension while $\mathbf{D}_\mathbf{t}^\mathbf{y}$ contains the movement in the vertical dimension. To convey the movement across a sequence of video frames, we construct a $2L$ input component by stacking motion vectors of the current frame t and next $L-1$ consecutive frames (we call this input multi-frame motion vectors). L denotes the staking length of the multi-frame motion vectors. Let w and h be the width and height of the input video, then the multi-frame motion vectors representation $\mathbf{T}_t \in \mathbb{R}^{w \times h \times 2L}$ for current frame t is constructed as follows:

$$\begin{cases} \mathbf{T}_t(2i-1) = \mathbf{D}_{t+i-1}^x \\ \mathbf{T}_t(2i) = \mathbf{D}_{t+i-1}^y \end{cases}, 1 \le i \le L. \tag{3}$$

In this equation, the horizontal and vertical components of $\mathbf{D}_t$ are stacked crossly. Finally, $\mathbf{T}_t$ is input to the temporal stream for frame t.

3.3 Foveated two-stream ConvNets for video summarization

Figure 1 shows our proposed model for dynamic video summarization. Given the input video, we first construct foveated images and multi-frame motion vectors for it. Then, the foveated images are input to the spatial stream and multi-frame motion vectors are input to the temporal stream to extract discriminative features for each frame. Each stream is implemented using a deep ConvNet. In this paper, we prefer to use a deep ConvNet VGG-16 [32]. The architecture of this ConvNet is $C64-C64-C128-C128-C256-C256-C256-C512-C512-C512-C512-C512-C512-F4096-F4096-F2$. It has 13 convolution layers (represented by C with the number of neurons) and 3 fully-connected

layers (represented by F with the number of neurons). Figure 3 shows the structure of the two-stream deep ConvNets. All convolutional layers are with stride 1. Also, considering that the VGG-16 model has a fixed size input, we sample the multi-frame motion vectors representation $\mathbf{T}_t$ to be a $224 \times 224 \times 2L$ sub-volume and input it to temporal steam.

In the learning stage, the training data with their corresponding labels are input to train each stream. In the inference procedure, we fuse the output of the 15^{th} layer in each stream as the features for each video frame. We suppose the outputs of the 15^{th} layer in the spatial stream and temporal stream for frame t are denoted as f_t^s and f_t^p. They are two 4,096-dimensional vectors. Then the output feature f_t for frame t is constructed based on these two vectors:

$$f_t = \left[f_t^s, f_t^p\right] \tag{4}$$

The resulting 8,192-dimensional vector f_t is then input to the subsequent SVR algorithm to predict the highlight score of the current frame.

In this paper, we use the support vector regression (SVR) model proposed by Drucker et al. [7] to predict the highlight score for each frame. SVR has been widely reported to achieve good performances in many computer vision and machine learning problems. As shown in Fig. 1, in the learning stage, the fused features with the corresponding average probability of user selection (we can also call it as user score) are input to SVR. In the inference scheme, the learnt SVR is used to predict the highlight score for each frame relied on its feature. Finally, we construct the final video summary according to their predicted highlight scores. The final summary is comprised of those video frames with highest K percentage of the predicted highlight scores. For SVR, we use the standard toolbox LIBSVM [3]. For the kernel function, we utilize the Radial Basis Function (RBF). In addition, a grid search is run to find the optimal parameter settings.

4 Experiments

In this section, we conduct several experiments to demonstrate the effectiveness of our proposed method. Firstly, we will introduce how we collect the gaze data, the evaluating metric and our implementation details in Section 4.1. Secondly, some experiments conducted in SumMe dataset will be illustrated in Section 4.2. The experiments include the consistency of subjects' eye movements, the comparison between our proposed model and several state-of-the-art methods, the effectiveness comparison based on the collected gaze locations or the locations predicted by an attention method, the visualization of our predicted summary

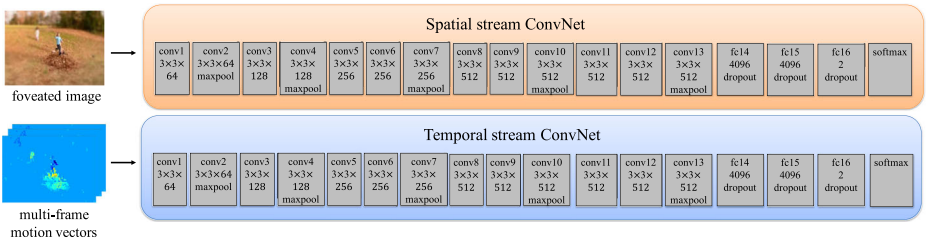


Fig. 3 The structure of the two-stream deep ConvNets. Each ConvNet has 13 convolutional layers and 3 fully-connected layers. All convolutional layers are with stride 1 and the ReLU activation function is not shown for brevity

results and the performance with different summary lengths on SumMe dataset. Finally, the experimental results on another video summarization benchmark dataset TVSum are shown in Section 4.3. Although Xu et al. [42] and Salehin et al. [30] have proposed dynamic video summarization models based on gaze data, they conducted experiments on unpublished datasets. We are not able to compare our proposed method with them.

4.1 Experimental setting

Gaze data collection The Gaze data was collected on two video summarization benchmark datasets: SumMe [10] and TVSum [33]. SumMe contains 25 videos and TVSum is comprised of 50 videos. These videos were properly resized to provide a clear view for observers. We asked participants to freely watch the entire video and muted the audio to ensure that there was only visual stimuli. The visual stimuli was presented on a screen (resolution: 2560×1440, refresh rate: 60 Hz) and generated by Psychtoolbox-3 toolbox [17] in Matlab on a MacBook Pro computer. Observers were stabilized by a chin and forehead rest. They were also maintained a constant viewing distance of 57 cm, resulting in a display with a visual angle of 51.2° × 33.3°. We collected the gaze data with a SR Research EyeLink 1000 video-based eye tracker. Horizontal and vertical positions of the right eye were recorded at 1000 Hz. The calibration and validation were conducted with the standard nine-point method included with the system. We used three participants (one male and two females) on SumMe and two participants (two females) on TVSum in our task. All observers had normal or corrected-to-normal vision.

Evaluation metric In our experiments, we compare automatic summarization (A) with the human-created summaries (B) and report the F-measure score to measure the performance of compared methods for evaluation. Many existing work on video summarization utilized this metric to demonstrate their methods [10, 22, 33], which is defined as follows:

$$F = \frac{2 \times p_r \times r_e}{p_r + r_e}, \tag{5}$$

$$r_e = \frac{\#matched \quad pairs}{\#frames \quad in \quad B} \times 100\%, \tag{6}$$

$$p_r = \frac{\#matched \quad pairs}{\#frames \quad in \quad A} \times 100\%. \tag{7}$$

where r_e is the recall and p_r is the precision. In our experiments, we report the mean F-measure and the nearest-neighbor F-measure (NN-F-measure) by comparing the predicted summaries with the user summaries. The mean F-measure is the average value of the F-measures for all users. It is given by:

$$\bar{F} = \frac{1}{N} \sum_{i=1}^{N} F_i \tag{8}$$

where N denotes the number of users, and F_i is the F-measure for users i. The NN-F-measure represents the maximal value of F_i, and it is given by:

$$F_{max} = \max_i(F_i) \tag{9}$$

This metric is used to evaluate the performance of the proposed method based on the most similar summary from all users. We use the standard toolbox proposed by Gygli et al. [10] to

evaluate our performance on SumMe and we utilize the evaluation code provided by Zhang et al. [48] on TVSum dataset.

Implementation details We follow the existing work [22, 47, 48] to randomly select 80% of the videos for training and utilize the rest of videos for testing in each dataset. For the final summary, the statistic from Gygli et al. [10] showed that the length of the summarization should be about 15% of the input video. Therefore, in most of our experiments, we set the summary length $K = 15$. For the parameters in ConvNets, we follow the general setting in [37] and we pre-train the ConvNets on the ImageNet dataset [5] to avoid the over-fitting. Based on the gaze data collection setting, the pixel per degree $p = 44$ in (2). Our method is implemented with Caffe [14] on the Tesla K80 GPU. We use the algorithm proposed by Zhang et al. [46] to extract motion vectors of the input videos. To demonstrate the effectiveness of motion vectors for video summarization, we compare the video summarization performance of using motion vectors and optical flow. We select the widely used toolbox to obtain optical flow [38, 45]. The stacking length L is set to 10 by following the existing work [31].

4.2 Video summarization on SumMe dataset

The SumMe dataset [10] contains 25 user videos depicting several events such as sports and cooking. The length of videos varies from 1 to 7 minutes. Each video has at least 15 user summary annotations as well as frame-level important scores. The annotation was collected in the controlled environment. For a given video, the users were asked to generate a summary that comprises most of its meaningful content, in other words, that best summarizes the input video. We find that users' responses vary from each others' even within the same video. The diversity and variety of the video contents and the users' responses make the dataset a challenging benchmark for video summarization.

To verify the consistency of the gaze information of various subjects, we first compute the distances of subjects' gaze points and compare them with the random baseline. For the random baseline, we calculate the distance between two random points in each frame. To calculate the similarity of subjects' eye movements, we compute the gaze-points distance of each pair of subjects in each frame. And then we report the average distances of each pair for each video. We resize the videos properly during the gaze data collection. The width of each video is resized to 1920 pixels and the height is resized proportionately (ranges from 1080 to 1440 pixels). This experiment is conducted on resized videos. Figure 4 shows the

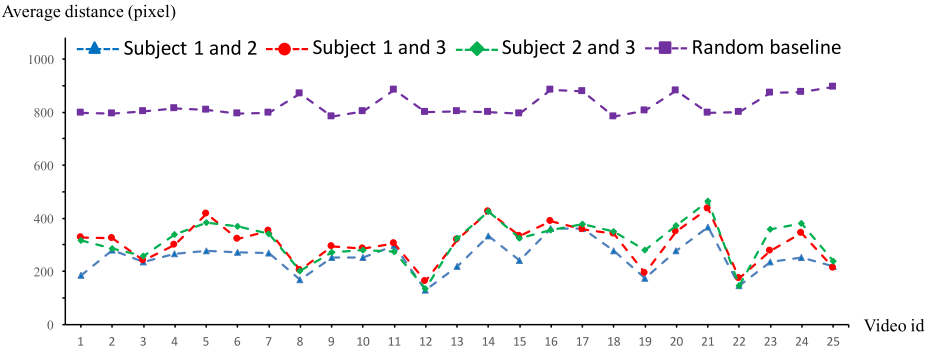


Fig. 4 The distance comparisons of three pairs of subjects' gaze points and the random baseline in 25 videos of SumMe

compared results. It is obvious that the distance of each pair of subjects' eye movements is significantly smaller than the random baseline. In addition, the distance of each pair is also quite similar to each other. These indicate that the subjects' gaze information are quite consistent. Specially, the broken line in blue locates lower than other cases. It shows that the eye movements of the subject 1 is more similar to subject 2 than subject 3.

We also show the video summarization performances of using different subjects' eye movements on SumMe dataset in Table 1. Firstly, the performances of three individual subjects are shown in the table. These three models generate foveated RGB image only based on single subject' s eye movement. Secondly, we provide a multiple subjects version of our proposed models which utilized the combination fixation points of subject 1, subject 2 and subject 3 into the construction of the foveated video frames. We applied these models based on two versions of our proposed method (FVS-FRGB and FVS-FRGB&MV). FVS-FRGB is our proposed one-stream ConvNet for video summarization, which utilizes foveated images as spatial stream input. FVS-FRGB&MV is our proposed two-stream ConvNets, which uses foveated images as the input of spatial stream and multi-frame motion vectors as the input of temporal stream. They are all evaluated on average mean F-measure (AMF) and average NN-F-measure (ANF). From Table 1, the average values and the standard deviation values (STD) of the performance among subject 1, subject 2 and subject 3 are given at the end of the table. It is clear that the multiple subjects model nearly outperforms all individual subject models as well as the average performances of them. In addition, the performances of the individual subject are quite consistent with each other and standard deviations are quite small.

Next, in order to evaluate the effectiveness of our proposed method, we compare our proposed method with other existing methods. As SumMe is a widely used standard benchmark dataset for video summarization, many work [10, 11, 18, 22, 33, 47, 48] have been validated in this dataset. We compare our method with static video summarization as well as several state-of-the-art dynamic video summarization methods: Video Representation Clustering based Video Summarization (VRCVS) [40], Creating Summaries from User Videos (CSUV) [10], Video MMR [18], Exemplar-based Subset Selection (ESS) [47], Learning Submodular Mixtures of Objectives (LSMO) [11], Summarizing Web Videos using Titles (SWVT) [33], Video Summarization with Long Short-term Memory (dppLSTM) [48] and Unsupervised Video Summarization with Adversarial LSTM Networks (SUM-GAN) [22].

Table 1 The performance comparisons of using different subjects' gaze information. We have the eye movement data of three subjects on SumMe. The table shows the performance of our proposed models (FVS-FRGB and FVS-FRGB&MV) by using different subjects' eye movements

Subject	Method			
	FVS-FRGB		FVS-FRGB&MV	
	AMF	ANF	AMF	ANF
subject1	34.6%	53.7%	35.5%	55.0%
subject2	34.1%	53.9%	36.4%	57.3%
subject3	34.3%	**56.1%**	36.3%	56.5%
multiple	**34.8%**	55.5%	**36.7%**	**57.7%**
average	34.3%	54.6%	36.1%	56.3%
STD	0.21%	1.08%	0.39%	0.94%

The best performance is marked in bold

VRCVS is a recent cluster-based static video summarization model which utilized a density-based clustering algorithm to generate a static storyboard for the input video [40]. While the rest of compared methods are all focused on dynamic video summarization task [10, 11, 18, 22, 33, 47, 48]. Basically, CSUV [10], Video MMR [18] and SWVT [33] are unsupervised methods based on hand-craft features. Gygli et al. selected an optimal segment subset based on low-, mid- and high-level visual features [10]. Li et al. proposed a dynamic video summarization based on a classical algorithm of text summarization, Maximal Marginal Relevance, which rewarded relevant keyframes and penalized redundant keyframes to generate the best summary result for each video [18]. Song et al. solved video skimming task by using title-based image search results based on standard image descriptors (color histograms, GIST and SIFT) [33]. While ESS, LSMO, dppLSTM and SUM-GAN are supervised methods based on deep learning features. Zhang et al. used human-created summaries to help the subset selection based on deep features [47]. Gygli et al. represented video segment in term of deep features trained on ImageNet to generate interesting and representative summary [11]. The dppLSTM and SUM-GAN used more latest deep learning architectures to formulate their methods. Zhang et al. proposed a supervised method based on long shot-term memory (LSTM) [48]. Mahasseni et al. introduced a novel generative adversarial network (GAN) consisting of the summarizer and discriminator for video summarization [22]. The compared methods can also be divided into frame-level and shot-level. Wu et al. evaluated the proposed method VRSCS on frame-level [40]. Gygli et al. provided the performance of the proposed method CSUV on both frame-level and shot-level [10]. The rest of the compared methods (Video MMR [18], SWVT [33], ESS [47], LSMO [11], dppLSTM [48], and SUM-GAN [22]) were all evaluated on shot-level. For most of the compared methods, we report the results published in their paper. Specifically, for dppLSTM [48], we calculate the F-measure score by using the evaluation code and summary results provided by Zhang et al. [48]. We report the best performance of their proposed method on SumMe dataset, which utilized other datasets, i.e. TVSum [33] for training. For SUM-GAN [22], we show the results of the supervised version, which also obtained the training data augmented with videos from other datasets (such as TVSum [33]). We implement VRSCS method [40] and report two versions of the summary results. For frame-based results of this static video summarization approach, we simply regard the static summary (storyboard) generated by the proposed method as the final summaries. For shot-based results, we first segment video into shots by using superframe algorithm proposed by Gygli et al. [10] and then select the shots which contain those frames in the storyboard as final selection results.

For the comparisons, we also provide different versions of our proposed methods based on foveated two-stream ConvNets for video summarization (FVS). There are one-stream deep architectures (FVS-OP, FVS-MV, FVS-RGB and FVS-FRGB) and two-stream deep networks (FVS-RGB&MV and FVS-FRGB&MV). In detail, FVS-OP, FVS-MV, FVS-RGB and FVS-FRGB indicate the methods that use optical flow, multi-frame motion vectors, RGB images and foveated RGB images as the input of the one-stream ConvNet, respectively. FVS-RGB&MV and FVS-FRGB&MV are models with two-stream learning structure. FVS-RGB&MV treats the RGB images as the input of the spatial stream and multi-frame motion vectors as the input of the temporal stream. While FVS-FRGB&MV uses the foveated RGB images as spatial stream input and multi-frame motion vectors as temporal stream input. We report our results on two levels: frame-level and shot-level. For frame-level version, we generate the final summary comprised of those video frames with highest 15 percentage of the predicted scores. As we discussed before, most of the compared existing methods are evaluated on shot-level. For fair comparisons, we follow the existing work [22, 48] to split the video into disjoint intervals by using kernel temporal segmentation (KTS)

[28]. Then, the final summary is comprising of those segments with highest predicted scores. The predicted score of a segment is equal to the average score of the frames in that interval. The total duration of the final summary segments is less than 15 percent of the length of the input video. To make the total duration of keyshots be below 15 percent of the original video, we also utilize the knapsack algorithm by following the existing work [10, 33, 48].

The comparison results are shown in Table 2 with the average mean F-measure (AMF) and the average NN-F-measure (ANF). From Table 2, it is obvious that human annotations are conducive to the performance of supervised methods, which allow most of them to achieve higher F-measure score than those unsupervised methods. Since most of the compared methods are based on shot-level, we report two versions of our proposed method on Table 2. Focusing on shot-level methods, we find that all of our proposed methods (FVS-MV, FVS-RGB, FVS-FRGB, FVS-RGB&MV and FVS-FRGB&MV) can generate better summary than the compared static video summarization methods, state-of-the-art dynamic video summarization methods, and even those deep learning based methods (dppLSTM and SUM-GAN) using latest deep learning architectures with augmented data. All of our proposed models have higher AMF and ANF than the compared methods, which demonstrates the effectiveness of our proposed foveated two-stream ConvNets for video summarization.

In the comparison of our proposed methods, it is clear that our proposed two-stream (FVS-RGB&MV and FVS-FRGB&MV) models gain higher F-measure score than those one-stream models (FVS-OP, FVS-MV, FVS-RGB and FVS-FRGB), which verifies that the essentials of two-stream architecture for video summarization. In the comparison of two kinds of motion input, the AMF and ANF of FVS-MV significantly outperform FVS-OP. We believe that motion vectors are more skilled at extracting those consistent motions of the prominent object with sufficient level of amplitude, which might be able to arouse strong emotional responses in viewers [6, 12], than optical flow. In the comparison of the

Table 2 The performance comparisons of our proposed methods with other models on SumMe dataset

		Method	Frame-level		Shot-level	
			AMF	ANF	AMF	ANF
Unsupervised methods	Existing static methods	VRCVS [40]	1.0%	0.5%	14.9%	40.4%
	Existing dynamic methods	CSUV [10]	23.4%	----	----	39.4%
		Video MMR [18]	----	----	----	26.6%
		SWVT [33]	----	----	26.6%	----
Supervised methods		ESS [47]	----	----	----	40.9%
		LSMO [11]	----	----	----	39.7%
		dppLSTM [48]	----	----	17.72%	42.9%
		SUM-GAN$_{sup}$ [22]	----	----	----	43.6%
	Proposed methods	FVS-OP	23.0%	39.9%	21.2%	47.6%
		FVS-MV	35.2%	53.8%	**27.6**%	55.3%
		FVS-RGB	32.0%	53.4%	25.2%	50.8%
		FVS-FRGB	34.8%	55.5%	27.2%	49.9%
		FVS-RGB&MV	35.4%	56.3%	26.1%	54.9%
		FVS-FRGB&MV	**36.7**%	**57.7**%	27.0%	**55.5**%

'----' denotes that the result is not reported in existing papers

The best performance is marked in bold

RGB image (FVS-RGB) and the foveated RGB image (FVS-FRGB), the results show that FVS-FRGB outperforms FVS-RGB in most cases which confirms the importance of gaze information for video summarization. For those two-stream architectures, the model which utilizes the foveated images as spatial stream input shows better performance than the model using the original RGB frames as input. We believe that foveated image provides user interesting regions for ConvNets. This enables the ConvNets to generate discriminative features, which are more related to users' scores.

We also provide the effectiveness comparison based on the collected gaze locations or the locations predicted by an attention method [51] on SumMe dataset. Zhou et al. proposed a class activation mapping technique (CAM) which was able to expose the implicit attention of convolutional neural networks on the image and highlight the most informative image regions [51]. In the experiment, we compare the summarized performances of two versions of FVB-FRGB on SumMe dataset. One utilizes the collected gaze locations to generate foveated images (denoted as FVB-FRGB) while the other uses highlighted regions detected by CAM to construct foveated images (denoted as FVB-FRGB-AttentionModel). The detailed processing is described as follow. Firstly, we utilize the class activation mapping technique proposed by Zhou et al. [51] to extract the attention regions of each video frame, which results in an activation map. The higher values of the activation map correspond to the regions which need to pay more attention to. Next, we extract three points of each activation map which have highest values to replace the gaze points of three subjects we collected in each video frame. Finally, each foveated image is generated based on previous three points extracted by CAM. The remaining processing is as same as the original FVB-FRGB. The experimental results are shown in Table 3. From the table, we can see that our proposed FVS-FRGB which generates foveated images based on locations from the real gaze data outperforms FVS-FRGB-AttentionModel which generates foveated images based on locations predicted by an attention method [51] by about 2%. Compared with the summarized results in Table 2, FVS-FRGB-AttentionModel is still able to exceed all compared methods, even though the real gaze locations are replaced by the predicted locations generated by an attention method. It is also proved the feasibility of integrating attention method into our proposed model in the future.

In Fig. 5, we visualize a sample of the predicted results. The displayed video is mainly about kids playing in the leaves around the house. At the beginning, three kids were on the top of the hill and then they jumped to the leaves. They gathered around and started a fight by using leaves as the weapon. After a while, the viewpoint was changed to catch the kids running around the house. Finally, they were back to the leaves to continue the battle. The first row of Fig. 5 shows the average possibility of each frame whether it would be the final summary based on all users' selections. We can also regard the probability as the score for each video frame. In the following three rows, we provide the predicted scores generated by our proposed methods: FVS-RGB&MV, FVS-FRGB&MV-shot and FVS-FRGB&MV. The

Table 3 The effectiveness comparison based on the collected gaze locations or the locations predicted by an attention method on SumMe dataset

Method	Frame-level		Shot-level	
	AMF	ANF	AMF	ANF
FVS-FRGB	**34.8%**	**55.5%**	**27.2%**	**49.9%**
FVS-FRGB-AttentionModel	33.7%	53.8%	24.5%	47.1%

The best performance is marked in bold

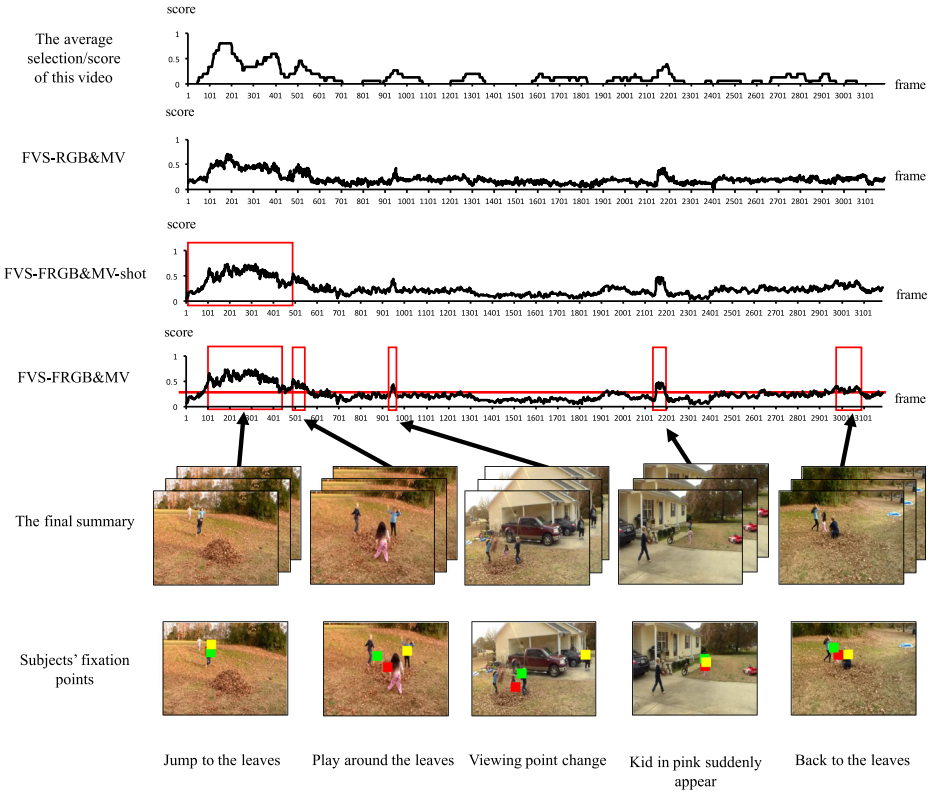


Fig. 5 A sample of the predicted results on video "Kids playing in the leaves" of SumMe database

fifth row describes the corresponding video content of the selected intervals. Finally, the last row indicates the subjects' fixation points on the video frames with each color representing a subject. From the example, it is clear that our proposed model can effectively extract the main content of the given video. It is able to capture the important segments about the kids playing in the leaves, which are similar with all the users' selections (roughly from 100 to 650 frame). It could also succeed in detecting several peaks of the average score of this video such as the interval begun at 2150 frame and ended at 2200 frame. In addition, by observing the frame-level performances of our proposed models in Table 2, we could find that even we do not segment video into shots in advance, our frame-level proposed methods are still capable to achieve competitive performance with those shot-level state-of-the-art methods. Interestingly, our frame-level based models gain higher F-measure score than our shot-level models. We believe that the shot segmentation algorithm restricts the structure of final summary. From Fig. 5, in the comparison of the third and fourth rows, our proposed foveated two-stream architecture not only can extract representative and interesting content of the input video, but can also learn the intrinsic connection of video frames, which help frame-level models achieve higher performance. Finally, by exploring the subjects' fixation points in the final summary, we find that the more aggregative the positions of the gazing point on a frame are, the more likely that the frame tends to be selected as final summary.

We also explore the influence of different summary lengths K on our proposed models. According to the statistics collected by Gygli et al. [10], the length of the final summary

tends to be 15 percent of the original video. In our models, we set $K = 15$ to generate video summary. Here, we provide the performance with different summary lengths K in Fig. 6. The comparison contains four methods, including the shot version of the static summarization method VRCVS-shot [40], our proposed one-stream architecture FVS-OP and our proposed two-stream models: FVS-RGB&MV and FVS-FRGB&MV. Figure 6a describes the average mean-F-measure, while Fig. 6b shows the average NN-F-measure on $K = 5, 10, 15, 20, 25$. It is clear that the two-stream method FVS-FRGB&MV, which utilizes foveated RGB image as the spatial steam input and multi-frame motion vectors as the temporal stream input, exceeds other models in all compared values of summary length. It achieves the best average mean-F-measure and the average NN-F measure when $K = 15$.

4.3 Video summarization on TVSum dataset

The TVSum dataset [33] consists of 50 videos collected from YouTube. These videos are from 10 categories (five videos per category), such as parade, bee keeping and grooming an animal. All categories are defined in the TRECVid Multimedia Event Detection (MED). The video length varies from 1 to 11 minutes. The dataset also provides 20 human annotations in term of shot-level important scores of 1 (not important) to 5 (very important) for each video. Each shot is with a uniform length of 2 seconds. Therefore, for SVR stage in our proposed method, we just uniformly subsample all the videos in TVSum to 2 frames per second by following the setting of previous work [48] and then each interval could have a shot-level important score itself for training. Finally, each test frame in the same shot would have the equal predicted score after SVR process.

In Fig. 7, we show a sample video with average important scores of TVSum dataset. The video captured the Chinese New Year parade in Chinatown. At the beginning, the title of this video appeared. Most of the users agreed this interval should have relatively higher important scores. Then the video started to record a man' s talking. The parade began at 1750^{th} frame. When the dancing lion appeared in the video, the average important scores were relatively high. At the end of the parade, users showed less interest in the video content. Finally, the video recorded the man talking about the parade.

Next, we compare our proposed methods with state-of-the-art approaches. Table 4 shows the comparison results. The compared methods include the static video summarization model VRSCS [40] and three dynamic video summarization models SWVT [33], dppLSTM

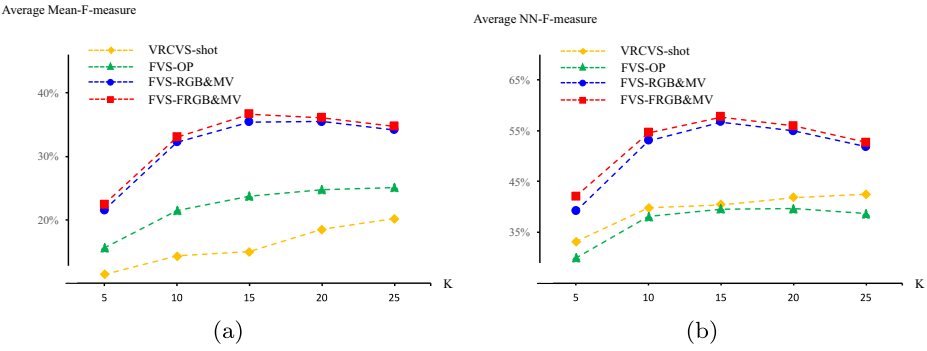


Fig. 6 The performance comparison vs. K on SumMe dataset. (**a**): The average mean-F-measure with different summary lengths K. (**b**): The average NN-F-measure with different summary lengths K

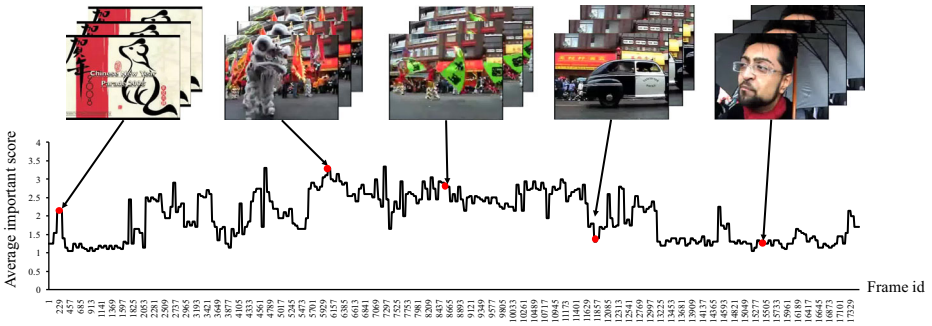


Fig. 7 A sample video "Chinatown parade" (video id: fWutDQy1nnY) with average important scores of TVSum database

[48] and SUM-GAN$_{sup}$ [22]. VRSCS and SWVT are unsupervised methods based on hand-craft features while dppLSTM and SUM-GAN are supervised methods based on advanced deep learning models LSTM and GAN. Here, we report the best performances of dppLSTM and SUM-GAN in their paper, which augmented the training data using other video summarization dataset (such as SumMe). As most of the compared methods (SWVT, dppLSTM and SUM-GAN) were evaluated on shot-level, we provide VRSCS-shot with the same setting in SumMe. We also provide several versions of our proposed method evaluated on shot-level. FVS-MV, FVS-RGB and FVS-FRGB are one-stream architectures which utilizes multi-frame motion vectors, RGB images and foveated images as input respectively. FVS-RGB&MV and FVS-FRGB&MV are based on two-stream deep ConvNets. FVS-RGB&MV uses RGB images as the input of spatial stream and multi-frame motion vectors as the input of temporal stream. While FVS-FRGB&MV uses foveated RGB images as the spatial stream input and multi-frame motion vectors as the temporal stream input.

From Table 4, we can find that the dynamic video summarization approaches achieve better performance than the static video summarization method on TVSum dataset. The deep learning based methods outperform the hand-crafted based methods. It is also clear that our proposed models gain significantly better summary performances than the compared models.

Table 4 The performance comparisons using the average F-measure on TVSum dataset

		Method	AMF	ANF
Unsupervised methods	Existing static methods	VRCVS-shot [40]	24.7%	34.0%
	Existing dynamic methods	SWVT [33]	50.0%	----
Supervised methods		dppLSTM [48]	58.7%	78.6%
		SUM-GAN$_{sup}$ [22]	61.2%	----
	Proposed methods	FVS-MV	58.2%	81.0%
		FVS-RGB	62.0%	**83.9**%
		FVS-FRGB	62.2%	83.5%
		FVS-RGB&MV	**62.8**%	83.8%
		FVS-FRGB&MV	62.2%	83.5%

'----' denotes that the result is not reported in existing papers

The best performance is marked in bold

5 Conclusions and discussion

In this paper, we propose a novel dynamic video summarization model based on foveated two-stream deep ConvNets. In the spatial stream, the foveated images are constructed based on subjects' fixation points to convey the visual appearance of the video. In the temporal stream, multi-frame motion vectors are built up to extract movement information of the input video.

In empirical validation, we evaluate our proposed method on two video summarization benchmark datasets. The experimental results demonstrate that the proposed methods can generate better video summary in comparison with the baseline methods as well as the state-of-the-art models. Meanwhile, extensive experimental results also show that the effectiveness of using foveated image and motion vectors. With the help of gaze information, our proposed foveated images achieve better performance than the original RGB images. Multi-frame motion vectors outperform the optical flow in video summarization task.

Although our model with an additional information: the eye tracking information, to obtain better summarized results, the subjects who were tracked eye movements were not the users who selected video summaries. We also find different subjects' eye movements are similar with each other. It means that eye movement of any subject could be helpful to generate a good video summary. On the other hand, eye movements become more and more convenient to obtain in recent years. Many portable equipments are produced. Papoutsaki et al. even proposed a webcam eye tracking method on the browser [26]. Moreover, some existing work seek to predict eye movements on multimedia data and they achieve not bad performances [19, 49]. In future, we will investigate how to integrate eye movement prediction stage into our model. Furthermore, we will propose an end to end architecture for video summarization task.

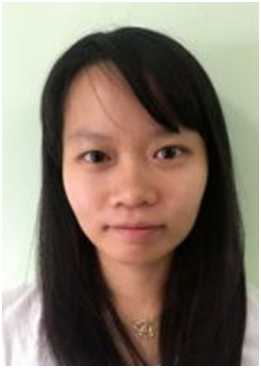
Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 61502311, No. 61620106008), the Natural Science Foundation of Guangdong Province (No. 2016A030310053), the Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase) under Grant (No.U1501501), the Shenzhen Emerging Industries of the Strategic Basic Research Project under Grant (No. JCYJ20160226191842793), the Shenzhen high-level overseas talents program, and the Tencent "Rhinoceros Birds" - Scientific Research Foundation for Young Teachers of Shenzhen University.

References

1. Bosch A, Zisserman A, Munoz X (2007) Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on image and video retrieval, pp 401–408. https://doi.org/10.1145/1282280.1282340
2. Bradley MM, Lang PJ (2015) Memory, emotion, and pupil diameter: repetition of natural scenes. Psychophysiology 52(9):1186–1193. https://doi.org/10.1111/psyp.12442
3. Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27,1–27,27
4. Daniel P, Whitteridge D (1961) The representation of the visual field on the cerebral cortex in monkeys. J Physiol 159(2):203–221
5. Deng J, Dong W, Socher R, Li JL, Li K, Li FF (2009) Imagenet: a large-scale hierarchical image database. In: Proceedings of the 2009 IEEE computer society conference on computer vision and pattern recognition, pp 248–255
6. Detenber B, Simons R, Bennett GG Jr (1998) Roll 'em!: the effects of picture motion on emotional responses. J Broadcast Electron Media 42:113–127
7. Drucker H, Burges CJC, Kaufman L, Smola AJ, Vapnik V (1997) Support vector regression machines. In: Mozer MC, Jordan MI, Petsche T (eds) Advances in neural information processing systems, vol 9, pp 155–161

8. Fu Y, Guo Y, Zhu Y, Liu F, Song C, Zhou ZH (2010) Multi-view video summarization. IEEE Trans Multimed 12(7):717–729. https://doi.org/10.1109/TMM.2010.2052025
9. Guenter B, Finch M, Drucker S, Tan D, Snyder J (2012) Foveated 3d graphics. ACM Trans Graph 31(6):164,1–164,10. https://doi.org/10.1145/2366145.2366183
10. Gygli M, Grabner H, Riemenschneider H, Van L (2014) Creating summaries from user videos. In: Proceedings of the European conference on computer vision
11. Gygli M, Grabner H, Van Gool L (2015) Video summarization by learning submodular mixtures of objectives. In: Proceedings of the 2015 IEEE conference on computer vision and pattern recognition
12. Hanjalic A, Xu LQ (2005) Affective video content representation and modeling. IEEE Trans Multimed 7(1):143–154
13. Holmberg N, Holmqvist K, Sandberg H (2015) Children's attention to online adverts is related to low-level saliency factors and individual level of gaze control. J Eye Mov Res 8(2):1–10
14. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick RB, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. CoRR arXiv:1408.5093
15. Kantorov V, Laptev I (2014) Efficient feature extraction, encoding, and classification for action recognition. In: Proceedings of the 2014 IEEE conference on computer vision and pattern recognition, pp 2593–2600
16. Karessli N, Akata Z, Schiele B, Bulling A (2017) Gaze embeddings for zero-shot image classification. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition
17. Kleiner M, Brainard D, Pelli D, Ingling A, Murray R, Broussard C (2007) What's new in psychtoolbox-3. Perception 36(14):1–16
18. Li Y, Merialdo B (2010) Multi-video summarization based on video-mmr. In: Proceedings of the 11th international workshop on image analysis for multimedia interactive services, pp 1–4
19. Li Y, Fathi A, Rehg JM (2013) Learning to predict gaze in egocentric video. In: Proceedings of the 2013 IEEE international conference on computer vision, pp 3216–3223. https://doi.org/10.1109/ICCV.2013.399
20. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94
21. Ma YF, Lu L, Zhang HJ, Li M (2002) A user attention model for video summarization. In: Proceedings of the Tenth ACM international conference on multimedia, pp 533–542. https://doi.org/10.1145/641007.641116
22. Mahasseni B, Lam M, Todorovic S (2017) Unsupervised video summarization with adversarial lstm networks. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition
23. Mishra AK, Aloimonos Y, Cheong LF, Kassim A (2012) Active visual segmentation. IEEE Trans Pattern Anal Mach Intell 34(4):639–653. https://doi.org/10.1109/TPAMI.2011.171
24. Nelson AL, Purdon C, Quigley L, Carriere J, Smilek D (2015) Distinguishing the roles of trait and state anxiety on the nature of anxiety-related attentional biases to threat using a free viewing eye movement paradigm. Cogn Emotion 29(3):504–526. https://doi.org/10.1080/02699931.2014.922460. pMID: 24884972
25. Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175. https://doi.org/10.1023/A:1011139631724
26. Papoutsakimz A, Sangkloy P, Laskey J, Daskalova N, Huang J, Hays J (2016) Webgazer: scalable webcam eye tracking using user interactions. In: Proceedings of the 25th international joint conference on artificial intelligence, pp 3839–3845
27. Pereira M, Camargo M, Aprahamian I, Forlenza O (2014) Eye movement analysis and cognitive processing: detecting indicators of conversion to alzheimer's disease. Neuropsychiatr Dis Treat 10:1273–1285
28. Potapov D, Douze M, Harchaoui Z, Schmid C (2014) Category-specific video summarization. In: Proceedings of the European conference on computer vision
29. Rovamo J, Virsu V (1979) Estimation and application of the human cortical magnification factor. Exper Brain Res Experimentelle Hirnforschung Expérimentation cérébrale 37:495–510
30. Salehin MM, Paul M (2017) A novel framework for video summarization based on smooth pursuit information from eye tracker data. In: 2017 IEEE International Conference on Multimedia Expo Workshops, pp 692–697. https://doi.org/10.1109/ICMEW.2017.8026294
31. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Proceedings of the 27th international conference on neural information processing systems, pp 568–576
32. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. CoRR arXiv:1409.1556
33. Song Y, Vallmitjana J, Stent A, Jaimes A (2015) Tvsum: summarizing web videos using titles. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5179–5187
34. Truong BT, Venkatesh S (2007) Video abstraction: a systematic review and classification. ACM Trans Multimed Comput Commun Appl 3(1):1–37
35. Vul E, Alvarez G, Tenenbaum JB, Black MJ (2009) Explaining human multiple object tracking as resource-constrained approximate inference in a dynamic probabilistic model. In: Bengio Y,

Schuurmans D, Lafferty JD, Williams CKI, Culotta A (eds) Advances in neural information processing systems, vol 22, pp 1955–1963
36. Wang Z, Bovik CA, Lu L (2003) Foveated wavelet image quality index. In: Proceedings of SPIE - the international society for optical engineering, p 4472
37. Wang L, Xiong Y, Wang Z, Qiao Y (2015) Towards good practices for very deep two-stream convnets. CoRR arXiv:1507.02159
38. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Gool LV (2016) Temporal segment networks: towards good practices for deep action recognition. CoRR arXiv:1608.00859
39. Wick DV, Martinez T, Restaino SR, Stone BR (2002) Foveated imaging demonstration. Opt Express 10(1):60–65. https://doi.org/10.1364/OE.10.000060
40. Wu J, Zhong Sh, Jiang J, Yang Y (2017) A novel clustering method for static video summarization. Multimed Tools Appl 76(7):9625–9641
41. Xie YH, Setia L, Burkhardt H (2007) Object-based color image retrieval using concentric circular invariant features. Int J Comput Sci Eng Syst 1:159–166
42. Xu J, Mukherjee L, Li Y, Warner J, Rehg JM, Singh V (2015) Gaze-enabled egocentric video summarization via constrained submodular maximization. In: Proceedings of the 2015 IEEE conference on computer vision and pattern recognition, pp 2235–2244
43. Yao T, Mei T, Rui Y (2016) Highlight detection with pairwise deep ranking for first-person video summarization. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition, pp 982–990
44. Yun K, Peng Y, Samaras D, Zelinsky GJ, Berg TL (2013) Studying relationships between human gaze, description, and computer vision. In: Proceedings of the 2013 IEEE conference on computer vision and pattern recognition, pp 739–746
45. Zach C, Pock T, Bischof H (2007) A duality based approach for realtime tv-l1 optical flow. In: Proceedings of the 29th DAGM conference on pattern recognition, pp 214–223
46. Zhang B, Wang L, Wang Z, Qiao Y, Wang H (2016) Real-time action recognition with enhanced motion vector cnns. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition, pp 2718–2726
47. Zhang K, Chao L, Wei, Sha F, Grauman K (2016) Summary transfer: exemplar-based subset selection for video summarization. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition
48. Zhang K, Chao L, Wei, Sha F, Grauman K (2016) Video summarization with long short-term memory. In: Proceedings of the European conference on computer vision
49. Zhang M, Ma KT, Lim JH, Zhao Q, Feng J (2017) Deep future gaze: gaze anticipation on egocentric videos using adversarial networks. In: Proceedings of the 2017 IEEE conference on computer vision and pattern recognition
50. Zhao S, Liu Y, Han Y, Hong R, Hu Q, Tian Q (2017) Pooling the convolutional layers in deep convnets for video action recognition. IEEE Trans Circ Syst Video Technol PP(99):1–11
51. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition



Jiaxin Wu received her B.Sc. in College of Computer Science and Software Engineering from Shenzhen University in 2015. She is currently a post-graduate student in the College of Computer Science and Software Engineering, Shenzhen University. Her current research interests include video content analysis and deep learning.

Sheng-hua Zhong received her B.Sc. in Optical Information Science and Technology from Nanjing University of Posts and Telecommunication in 2005 and M.S. in Signal and Information Processing from Shenzhen University in 2007. She got her Ph.D. from Department Computing, The Hong Kong Polytechnic University in 2013. She worked as a Postdoctoral Research Associate in Department of Psychological & Brain Sciences at The Johns Hopkins University from 2013 to 2014. Currently, she is an Assistant Professor in College of Computer Science & Software Engineering at Shenzhen University in Shenzhen. Her research interests include multimedia content analysis, brain science, and machine learning.

Zheng Ma received her B.Sc. in Cognitive Psychology from Peking University, China, in 2011. She got her M.A. and Ph.D. in the Department of Psychological and Brain Sciences from the Johns Hopkins University in 2013 and 2016. She is currently a postdoctoral research fellow at the Smith-Kettlewell Eye Research Institute in San Francisco. Her research interests include human visual cognition, oculomotor movements, and using computational models to understand human mind.

Stephen J. Heinen received his dual B.A. in Psychology and Mathematics from the Wright State University in 1981. He got his M.A. and Ph.D. in Experimental Psychology from the Northeastern University in 1987 and 1988. He is currently a senior scientist at the Smith-Kettlewell Eye Research Institute in San Francisco. His laboratory mainly studies how human visual perception and cognition guide the oculomotor system to provide information about basic neural mechanisms that drive this system.

Jianmin Jiang received the Ph.D. degree from the University of Nottingham, Nottingham, U.K., in 1994. He joined Loughborough University, Loughborough, U.K, as a Lecturer in computer science. From 1997 to 2001, he was a Full Professor of Computing with the University of Glamorgan, Wales, U.K. In 2002, he joined the University of Bradford, Bradford, U.K, as a Chair Professor of Digital Media, and Director of Digital Media and Systems Research Institute. In 2014, he moved to Shenzhen University, to carry on holding the same professorship. He is also an Adjunct Professor with the University of Surrey, Guildford, U.K. His current research interests include image/video processing in compressed domain, computerized video content under- standing, stereo image coding, medical imaging, computer graphics, machine learning, and AI applications in digital media processing, retrieval, and analysis. In 2010, he was elected as a scholar of One-Thousand-Talent-Scheme funded by the Chinese Government.