Spatiotemporal Modeling and Label Distribution Learning for Video Summarization

Wei-Ta Chu
National Cheng Kung University
Tainan, Taiwan
wtchu@gs.ncku.edu.tw

Yu-Hsin Liu
National Chung Cheng University
Chiayi, Taiwan
share790113@gmail.com

***Abstract*—For a video which content does not follow specific production rules, or without professional editing, at least two problems should be solved to generate a good video summary. First, the summarization system should jointly model visual content in the spatial domain and visual dynamics in the temporal domain. Second, the system should consider the inconsistency between users, i.e., different users may annotate the same video segment with different importance scores. In this paper, we present a video summarization system that models spatiotemporal information of video segments, and predicts the distribution of importance scores for each segment. Based on the estimated importance scores, video summaries are generated by picking the ones with higher scores. We especially demonstrate the effectiveness of label distribution learning based on two video benchmarks.**

***Index Terms*—video summarization, recurrent neural network, label distribution learning**

I. Introduction

With the explosive growth of smartphones and cameras, people can easily capture a large number of videos and share them on the internet. According to YouTube's statistics in 2018, the length of uploaded videos per minute is 500 hours. A wide variety of videos enriches our life but also raises significant challenges on efficient access. Therefore, an automatic video summarization system or highlight detection system is urgently demanded to help us quickly access the most informative parts.

Many studies of video summarization or video highlight detection have been proposed, and some open datasets (e.g., SumMe [1], TVSum [2], and CoSum [3]) have been released in public. According to how videos were produced and edited, videos can be divided into two categories: (1) the ones with production rules or professional editing and (2) the ones without them. In the first category, things happen according to rules or are edited professionally (e.g., sports competition and news). In broadcasted baseball videos, for example, well-defined events are included, such as Home Run, Strike Out, and Steal. By detecting and including specific events, we are able to generate good video summaries. In the second category, videos are usually captured casually by mobile cameras like smartphone or GoPro. This type of videos may not include predefined events, and summaries cannot be generated based on detected events. Summarizing the first type of videos has been widely studied and achieved promising performance. In this paper, we focus on the second type of videos.

978-1-7281-1817-8/19/$31.00 ©2019 IEEE

Video summarization is somewhat subjective. Someone may think that a part is important, but others may not. This situation is even severe when content of videos doesn't follow production rules or without professional editing. The inconsistency between users have been quantitatively and qualitatively demonstrated in [4].

Video summarization is a problem that we need to select subsets of video frames that contain the most informative parts of the original video. Particularly, given a sequence of frames $\boldsymbol{x} = (x_1, x_2, ..., x_N)$, we need to estimate the degree of importance $\hat{\boldsymbol{y}} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_N)$ corresponding to each frame. The estimated scores $\hat{\boldsymbol{y}}$ should be as close to the ground truth $\boldsymbol{y} = (y_1, y_2, ..., y_N)$ as possible. Therefore, the video summarization can be conceptually formulated as finding a function f that maps $\boldsymbol{x}$ into $\boldsymbol{y}$. In finding the function f, two key issues should be considered: 1) How to effectively find a model that well describes the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$? 2) For a given video, summaries created by different users are usually inconsistent because of subjectivity. How to generate a summary that is generally good? To handle the first issue, we would develop a neural network modeling spatiotemporal information of a video. For the second issue, we would propose a new loss function that jointly considers labels given by different users. A label distribution is used to describe the degree of importance for each video segment, rather than a single value averaged from multiple users' labels. The main contribution of this work is that we consider label distribution learning in the proposed network.

The rest of this paper is organized as follows. Section II presents details of the network modeling spatiotemporal information of videos. Section III describes the loss function based on label distribution. Evaluation results are shown in Section IV, followed by concluding remarks in Section V

II. Spatiotemporal Modeling

A video contains a sequence of frames showing dynamics of objects or scenes. We therefore need to describe visual content of each frame in the spatial domain, as well as visual dynamics in the temporal domain. In this work, we modify the framework proposed in [4] to model spatiotemporal information of a video.

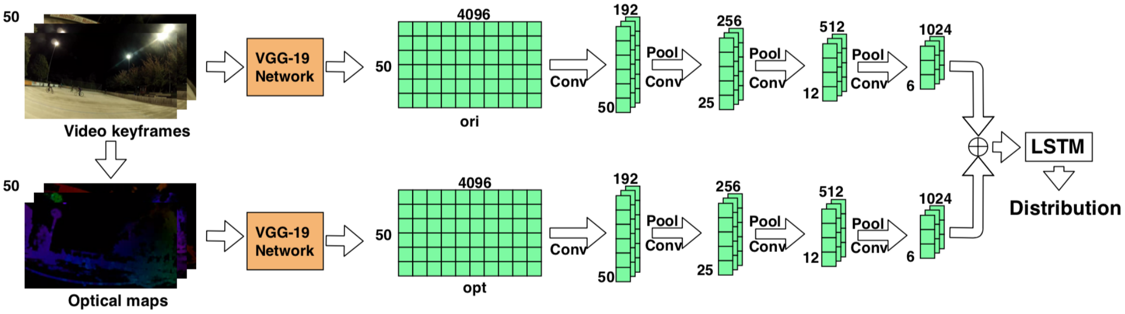

Fig. 1. Illustration of the framework that extracts spatiotemporal information by convolutional neural networks and predicts the label distribution.

The main difference between the proposed model and [4] is twofold. First, we jointly consider visual information extracted from video frames and optical flow maps, while the work [4] relies only on video frames. Second, we propose to model different users' ratings as a label distribution, and the objective of the proposed network is to predict the distribution rather than a single importance score. In [4], they refine the ground truth by evaluating different users' rating quality. Figure 1 illustrates our idea. We describe the spatiotemporal modeling in the following, and describe label distribution learning in Sec. III.

To pick important video segments to form a video summary, we would like to evaluate the importance score of each video segment by the framework shown in Figure 1. In this work, we estimate the importance score of every 1-second segment. To evaluate the ith second segment, we consider context information from the $(i-2)$th second to the $(i+2)$th second, i.e., 5 seconds in total. With this setting, we divide a given video into overlapped segments by a sliding window of 5 seconds with stride 1 second. The length of each video segment is 5 seconds, overlapped with the next segment by 4 seconds. For each video segment, we uniformly pick one out of three frames as the keyframes, in order to largely reduce volume of processing data. To further consider short-time motion dynamics, we adopt Gunner Farneback's algorithm [5] implemented in the Dlib toolkit[1] to find the dense optical flow map associated with each video frame. We also uniformly pick one out of three maps as the keymaps.

To describe visual content and visual dynamics, we respectively feed each keyframe and each keymap into the VGG-19 network [6] that was pre-trained based on the ImageNet dataset [7]. The 4096-dimensional output vector of the second last layer of the VGG-19 network is taken as the descriptor of each keyframe or each keymap. The feature extraction process is basically completed by 2D convolution defined in the VGG-19 network. In the following, let $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_K)$ denote the descriptors of K keyframes, and $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_K)$ denote the descriptors of K key optical flow maps. As each video segment is 5 seconds, and frame rate of the evaluated videos is 30, the value K is 50 in our work.

Visual content changes along the temporal dimension. Taking keyframes as the example, we first stack the K transposed descriptors $\boldsymbol{u}_1^T, ..., \boldsymbol{u}_K^T$ to form a $K \times 4096$ feature map $\boldsymbol{M}_u$. To describe short-term visual evolution, a sequence of convolution and pooling processes are applied to the feature map $\boldsymbol{M}_u$. Details of the processes are described in Table I. Notice that the convolution over rows of $\boldsymbol{M}_u$ captures dynamic characteristics over multiple keyframes. Taking the first convolutional layer in Table I as an example, the convolution kernel is 11×4096, which means 11 keyframes/keymaps are jointly considered, and each keyframe/keymap is represented as a 4096-dimensional vector. Notice that this approach is using convolutional architecture to model temporal evolution. It has been shown effective from the perspectives of network complexity and training cost [8].

After the sequence of convolution and pooling, the feature map $\boldsymbol{M}_u$ is transformed into a 6×1024 feature map $\boldsymbol{\mu}$. Similarly, we also can apply the same process to descriptors $\boldsymbol{v}_1^T, ..., \boldsymbol{v}_K^T$ of key optical flow maps, and then obtain a 6×1024 feature map $\boldsymbol{\nu}$. To jointly consider visual content and visual dynamics, the aforementioned two feature maps are combined as a 6×2048 map, where each row is integrated information in the representation of a 2048-dimensional vector, and the input video segment is roughly represented as a series of six 2048-dimensional vectors. To estimate the importance score of the given video segment, we consider the "long-term" evolution along the 6 time instants, and feed the 6 vectors to a bidirectional long short term memory network (LSTM). Given a video segment from the $(i-2)$th second to the $(i+2)$th second, this bidirectional LSTM finally outputs a score estimating the importance of the ith second.

In our work, features are extracted from keyframes and keymaps. To jointly consider information from two types of data, in fact two different fusion schemes can be adopted:

- Late fusion: As mentioned in Figure 1, we can separately form two $K \times 4096$ matrices, and apply a sequence of convolution and pooling processes to them. Results of two streams, i.e., $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, are then concatenated and are used together to estimate the importance score.
- Early fusion: For each keyframe/keymap, we can first concatenate two types of descriptors together to form

[1] http://dlib.net

TABLE I
DETAILED CONFIGURATION OF THE PROPOSED NETWORK.

Name	Filter Size	#Filters	Size of output
Conv_1	11×4096	192	50×192
Max_1	2×1	–	25×192
Conv_2	5×192	256	25×256
Conv_3	5×256	256	25×256
Max_2	2×1	–	12×256
Conv_4	3×256	512	12×512
Conv_5	3×512	512	12×512
Max_3	2×1	–	6×512
Conv_6	1×512	1024	6×1024
Bidirectional LSTM	–	–	1 or 5

a 8192-dimensional vector $\boldsymbol{w} = (\boldsymbol{u}, \boldsymbol{v})$, and then stack descriptors of a video segment to form a $K \times 8192$ matrix. A sequence of convolution and pooling processes are then applied to this matrix, and finally the transformed vector is fed to an LSTM to estimate the importance score. The approach adopted in [4] is similar to the early fusion scheme.

Two fusion methods will be compared in the evaluation section.

Given a video $\boldsymbol{X} = (x_1, x_2, ..., x_N)$ and the importance score of each frame $\boldsymbol{Y} = (y_1, y_2, ..., y_N)$, we can train the proposed network by minimizing the loss defined as the mean square error between the ground truth $\boldsymbol{Y} = (y_1, y_2, ..., y_N)$ and the predicted scores $\hat{\boldsymbol{Y}} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_N)$, i.e.,

$$L = \|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\|_2^2. \tag{1}$$

In the evaluation section, we call the network trained based on the loss mentioned above as the *baseline* method.

III. LABEL DISTRIBUTION

The ground truth score y_i of each frame is usually averaged from multiple users' scores. However, because of the inconsistency between different users [4], the average score generally doesn't represent the consensus between users. Averaging multiple scores, on the other hand, may decrease the rich variations between users. Motivated by [9], we would like to represent the scores given by multiple users as a "label distribution". We can train the proposed network based on the label distribution rather than a single average score.

Fan et al. investigated facial attractiveness estimation in [9]. Similar to video summarization, estimating the degree of attractiveness of a face is a subjective task, and an attractive score averaged from multiple subjects' opinions is not a universal indicator, especially for controversial faces. Considering further that training data for facial attractiveness estimation are scarce, they collected scores given by subjects and described a face by the score distribution rather than a single score. They thus formulated attractiveness estimation as a label distribution learning problem [10].

Motivated by [9] and [10], we also want to adopt the idea of label distribution prediction for video summarization. For a video, the ith user may label a few segments with specific importance scores as $\boldsymbol{T}^{(i)} = \{(b_1^{(i)}, e_1^{(i)}, s_1^{(i)}), (b_2^{(i)}, e_2^{(i)}, s_2^{(i)}), ..., (b_J^{(i)}, e_J^{(i)}, s_J^{(i)})\}$, where $b_1^{(i)}$ is the beginning time, $e_1^{(i)}$ is the end time, and $s_1^{(i)}$ is the importance score of the first segment, respectively. Notice that we normalize importance scores based on the maximum value given by each subject, causing the values of $s_j^{(i)}$ ranging from 0 to 1. Totally J segments are annotated in this case, and each segment is represented as a 3-tuple. Another user may annotate different numbers of video segments, with different beginning time, end time, and important scores. Given a set of tuples $\{\boldsymbol{T}^{(1)}, ..., \boldsymbol{T}^{(I)}\}$, by a quantization function $\boldsymbol{Q}$, we first quantize the given scores into one of the five ranges, say $R_1 = [0, 0.2)$, $R_2 = [0.2, 0.4)$, $R_3 = [0.4, 0.6)$, $R_4 = [0.6, 0.8)$, and $R_5 = [0.8, 1]$. For the kth second of a video, the value of the rth range R_r is calculated by

$$\boldsymbol{B}_{k,r} = \sum_{i=1}^{I} \sum_{j=1}^{J} \delta(b_j^{(i)}, e_j^{(i)}, k), \tag{2}$$

$$\delta(b_j^{(i)}, e_j^{(i)}, k) = \begin{cases} 1 & \text{if } b_j^{(i)} \leq k \leq e_j^{(i)} \text{ and } \boldsymbol{Q}(s_j^{(i)}) \in R_r, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The label distribution $\boldsymbol{B}_k$ for the kth second can be constructed by checking the number of scores falling into ranges of R_1 to R_5.

Figure 2 shows examples of label distributions corresponding to different video segments. As can be seen, the distribution of an important segment is left-skewed because most users label it with higher scores, and the distribution of a less important segment is right-skewed because most users label it with lower scores.

Given a video of K seconds $\boldsymbol{X} = (x_1, x_2, ..., x_K)$ and the distribution of importance scores of each second $\boldsymbol{B} = (\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_K)$, we train the proposed network based on the loss defined as the cosine proximity between the truth label distribution $\{\boldsymbol{B}_i\}$ and the predicted distribution $\{\hat{\boldsymbol{B}}_i\}$, i.e.,

$$L = -\sum_{i=1}^{N} \frac{\boldsymbol{B}_i \cdot \hat{\boldsymbol{B}}_i}{\|\boldsymbol{B}_i\| \|\hat{\boldsymbol{B}}_i\|}. \tag{4}$$

In this scenario, the bidirectional LSTM mentioned in Figure 1 and Table I outputs a 5-dimensional vector estimating the importance distribution of the ith second.

Given a test video, the proposed framework first extracts spatiotemporal features mentioned in Sec. II and predicts label distributions for every 1-second video segment. Assume that, for the kth 1-second video segment, the predicted distribution is $\hat{\boldsymbol{B}}_k = (\hat{b}_1, \hat{b}_2, ..., \hat{b}_5)$, the overall estimated importance score is calculated as the weighted average of $\hat{b}_1$ to $\hat{b}_5$, i.e., $s_k = 0.1 \times \hat{b}_1 + 0.3 \times \hat{b}_2 + ... + 0.9 \times \hat{b}_5$, where 0.1, 0.3, ..., 0.9 are the midpoints of ranges R_1 to R_5. Therefore, a video of K seconds will be associated with K importance scores $s_1, s_2, ..., s_K$. To generate the summary, video segments with higher importance scores are picked. In practice, to fairly compare with other works, the segments with the top 15% highest scores are picked. This is the 0/1 knapsack problem,

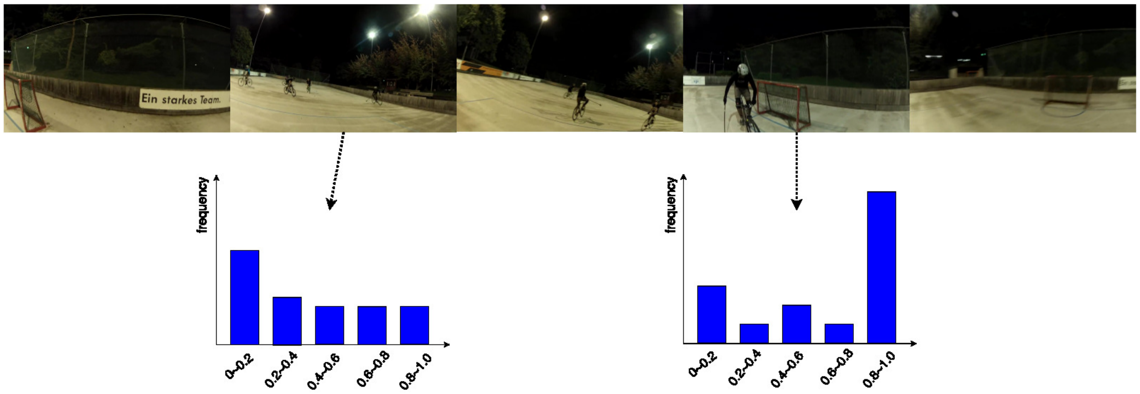

Fig. 2. Examples of label distributions of two video segments. The left shows the distribution corresponding to an overview video segment, and the right shows the distribution corresponding to the video segment when the player makes a shot in a hockey game.

and we solve it by dynamic programming [1]. Finally, the picked video segments are concatenated in chronological order to be the final summary.

IV. Evaluation Results

We evaluate the proposed framework based on two benchmark datasets: the SumMe dataset [1] and the TVSum dataset [2]. The SumMe dataset is consisted of 25 videos, including a wide range of videos covering holidays, events and sports videos. The TVSum dataset contains 50 videos in various genres, such as news, documentaries, and user-generated content from YouTube. Each video in these two datasets are annotated by multiple users, and frame-level importance scores are provided. As mentioned in Sec. III, we construct a truth label distribution for each 1-second video segment based on the provided frame-level importance scores.

Following the setting in [1] and [4], we adopt the 5-fold cross validation scheme to evaluate the proposed method. Furthermore, given the generated summary S_A and the ground truth summary S_B, the F-score defined as follows is taken as the performance metric:

$$F(S_A, S_B) = \frac{2 \times P \times R}{P + R} \times 100\%, \tag{5}$$

where $P = \frac{\|S_A \cap S_B\|}{\|S_A\|}$ and $R = \frac{\|S_A \cap S_B\|}{\|S_B\|}$. The term $\|S_A\|$ denotes the length of the summary S_A, and the term $\cap$ denotes the temporal overlap.

Table II shows performance comparison between our methods and the state of the arts. The rows of "Ours-*w/o LD*" show performance of the proposed framework with spatiotemporal modeling but without label distribution estimation. The network is trained based on the loss defined in eqn. (1). The rows of "Ours-*w. LD*" show performance when spatiotemporal modeling and label distribution estimation are jointly considered. Comparing these two sets of results, we clearly see the performance gain brought by label distribution learning, i.e., 3%–5% improvements in terms of F-scores can be obtained.

In the rows of "Ours-*w. LD*", the late fusion scheme clearly outperforms the early fusion scheme. Different features may present in different temporal scales. If different types of features are concatenated before processing, unique information may lose. On the other hand, the late fusion scheme seems to maintain more information at late stages and yields better performance, especially when label distribution learning is adopted. Both the early and late fusion schemes are better than the state of the art [4]. In [4], the best performance can be obtained if the ground truth is refined by prioritizing different users' scores. To concentrate the comparison on methodologies, we compare with the version in [4] without ground truth refinement. The idea of ground truth refinement can also be integrated into our work in the future.

TABLE II
Performance comparison between different methods, in terms of F-scores.

Methods	SumMe	TVSum
Cycle-SUM [13]	41.9	57.6
DR-DSN$_{\text{sup}}$ [14]	42.1	58.1
SASUM$_{\text{sup}}$ [11]	45.3	58.2
HSA-RNN [12]	44.1	59.8
TS-STN [4]	46.1	60.0
Ours-*w/o LD* (early fusion)	43.6	57.3
Ours-*w/o LD* (late fusion)	42.2	57.9
Ours-*w. LD* (early fusion)	46.6	60.1
Ours-*w. LD* (late fusion)	**47.6**	**61.0**

To evaluate generality of the proposed framework, we try two different settings: train on the TVSum dataset and test on the SumMe dataset (denoted as T2S), and train on the SumMe dataset and test on the TVSum dataset (denoted as S2T). Table III shows performance of two different settings. Just like our expectation, when different sets of data are used for training and testing, performance degrades. However, comparing Table III with Table II, the degree of degradation is slight, which shows that the proposed method is robust.

Summarization performances for different videos are different. Figure 3 shows the distribution of F-scores over 25 videos in the SumMe dataset. In each run of training and testing, we use 20 videos for training, and the remaining 5 videos are tested. To form Figure 3, we accumulate the statistics of F-scores in five runs. As can be seen, F-scores for some videos

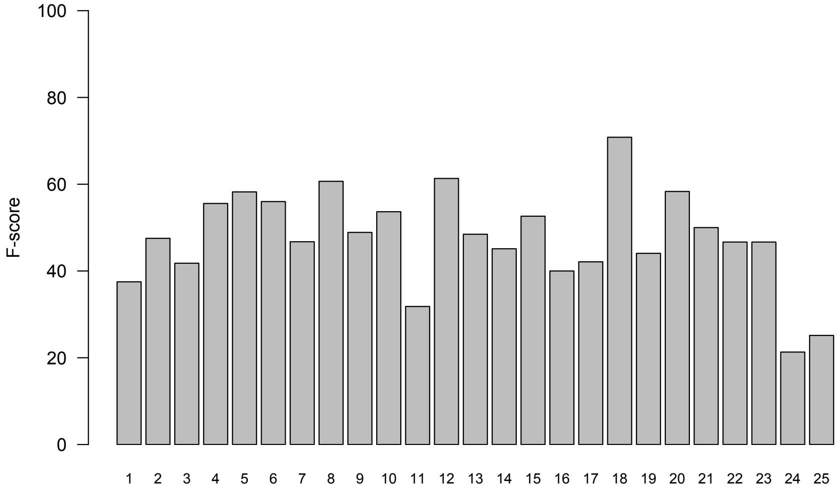

Fig. 3. The distribution of F-scores over 25 videos in the SumMe dataset, obtained based on the late fusion scheme. As we expect, different summarization performances are obtained for different videos.

TABLE III
PERFORMANCE OF DIFFERENT TRAINING/TESTING SETTINGS, IN TERMS OF F-SCORES.

Methods	T2S	S2T
Ours-*w/o LD* (early fusion)	39.2	51.6
Ours-*w/o LD* (late fusion)	40.8	53.6
Ours-*w. LD* (early fusion)	43.1	55.2
Ours-*w. LD* (late fusion)	**44.0**	**56.1**

are as high as 0.7, and some are around 0.2. For example, the F-score of the 24th video "Saving Dolphins" is only 21.3%. All frames of this video show people on the beach saving dolphins, as shown in the top row of Figure 4. There is no special event or movement, and more than 95% of importance scores are lower than 0.3. In other words, most subjects think that this video is nothing important. Similarly, the F-score of the 25th video "Uncut Evening Flight" is only 25.2%. This video was captured by a camera mounted on the wing of a drone, and the drone itself always occupies the central part of all frames, as shown in the bottom row of Figure 4. In these two videos, subjects' preferences are inconsistent, and the plain visual content makes summarization more challenging.

Figure 5 shows an example visualizing the summarization result of the "Notre Dame" video in the SumMe dataset (the 8th video shown in Figure 3, the F-score is 60.67%). The blue blocks in the top bar and the bottom bar indicate the positions and lengths of important segments picked manually (ground truth) and selected automatically by our system, respectively. There are 5 important segments in the ground truth and in the automatic summary, respectively. We show one keyframe for each important segment. Below the bottom bar, we show the curve of estimated importance scores. In this example, we see that 4 of the 5 important segments (shown in red borders) are picked by our system. This example shows that our estimated scores well reflect importance of different segments, and can be well utilized in video summarization.

V. CONCLUSION

We have presented a framework that extracts spatiotemporal information of video frames and optical flow maps by a neural network. Moreover, to tackle with the inconsistency of user preference, the proposed network predicts a distribution of importance scores rather than a single score to evaluate importance of video segments. Based on these estimated importance scores, video summaries are formed by picking the most informative segments. The evaluation results show that the proposed method outperforms the state of the arts, and generality of the method is also shown. In the future, more features in addition to visual appearance and motion should be extracted to improve summarization performance. Much larger-scale evaluation is also needed.

ACKNOWLEDGMENT

This work was partially supported by the Ministry of Science and Technology under the grant 108-2221-E-006-227-MY3, 107-2221-E-006-239-MY2, 107-2923-E-194-003-MY3, 107-2627-H-155-001, and 107-2218-E-002-055.

REFERENCES

[1] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool, "Creating summaries from user videos," in *Proceedings of European Conference on Computer Vision*, 2014, pp. 505–520.

[2] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes, "Tvsum: Summarizing web videos using titles," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5179–5187.

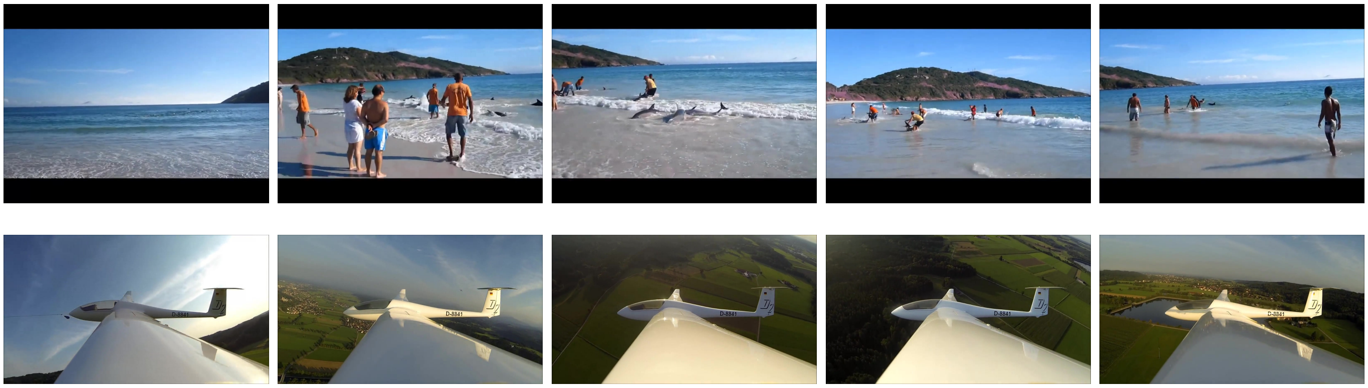

Fig. 4. Keyframes sampled from the videos “Saving Dolphins” (top row) and “Uncut Evening Flight” (bottom row) in the SumMe dataset.

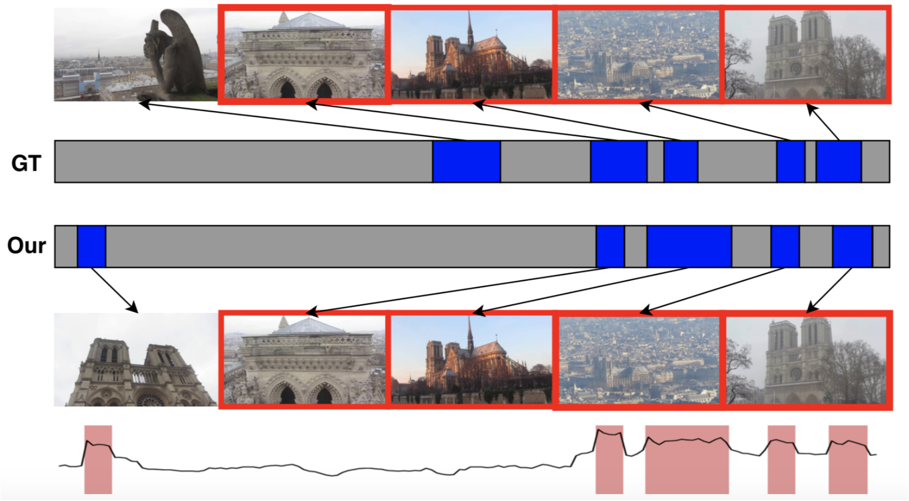

Fig. 5. An example visualizing summarization results of the “Notre Dame” video in the SumMe dataset. Four of five important segments are automatically picked into the summary.

[3] Wen-Sheng Chu, Yale Song, and Alejandro Jaimes, “Video co-summarization: Video summarization by visual co-occurrence,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3584–3592.

[4] Siyu Huang, Xi Li, Zhongfei Zhang, Fei Wu, and Junwei Han, “User-ranking video summarization with multi-stage spatio–temporal representation,” *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2654–2664, 2019.

[5] Gunnar Farneback, “Two-frame motion estimation based on polynomial expansion,” in *Proceedings of Scandinavian Conference on Image Analysis*, 2003, pp. 363–370.

[6] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proceedings of International Conference on Learning Representations*, 2015.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[8] Jyoti Aneja, Aditya Deshpande, and Alexander G. Schwing, “Convolutional image captioning,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[9] Yang-Yu Fan, Shu Liu, Bo Li, Zhe Guo, Ashok Samal, Jun Wan, and Stan Z Li, “Label distribution-based facial attractiveness computation by deep residual learning,” *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2196–2208, 2018.

[10] Xin Geng, “Label distribution learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734–1748, 2016.

[11] Huawei Wei, Bingbing Ni, Yichao Yan, Huanyu Yu, and Xiaokang Yang, “Video summarization via semantic attended networks,” in *Proceedings of AAAI Conference on Artificial Intelligence*, 2018.

[12] Bin Zhao, Xuelong Li, and Xiaoqiang Lu, “HSA-RNN: Hierarchical structure-adaptive rnn for video summarization,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7405–7414.

[13] Li Yuan, Francis EH Tay, Ping Li, Li Zhou, and Jiashi Feng, “Cycle-sum: Cycle-consistent adversarial lstm networks for unsupervised video summarization,” in *Proceedings of AAAI Conference on Artificial Intelligence*, 2019.

[14] Kaiyang Zhou, Yu Qiao, and Tao Xiang, “Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward,” in *Proceedings of AAAI Conference on Artificial Intelligence*, 2018.