CONTENT ADAPTIVE VIDEO SUMMARIZATION USING SPATIO-TEMPORAL FEATURES

Hyunwoo Nam and Chang D. Yoo

Korea Advanced Institute of Science and Technology
School of Electrical Engineering
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

ABSTRACT

This paper proposes a video summarization method based on novel spatio-temporal features that combine motion magnitude, object class prediction, and saturation. Motion magnitude measures how much motion there is in a video. Object class prediction provides information about an object in a video. Saturation measures the colorfulness of a video. Convolutional neural networks (CNNs) are incorporated for object class prediction. The sum of the normalized features per shot are ranked in descending order, and the summary is determined by the highest ranking shots. This ranking can be conditioned on the object class, and the high-ranking shots for different object classes are also proposed as a summary of the input video. The performance of the summarization method is evaluated on the SumMe datasets, and the results reveal that the proposed method achieves better performance than the summary of worst human and most other state-of-the-art video summarization methods.

***Index Terms*—** Video Summarization, Video Analysis, Motion Magnitude, Saturation, Convolutional Neural Networks

1. INTRODUCTION

Video summarization is a process that uses an algorithm to summarize a video to a short clip as shown in Fig. 1. It is receiving increasing attention due to the growing number of videos uploaded on the internet. The popularity of social network services, such as Facebook, Instagram, and Twitter, has increased the demand for such methods. To reduce the time required to go through a vast number of videos, video summarization is essential. Various summarization methods have been proposed based on visual attention, story, and auxiliary information of the video.

Visual attention based methods [1, 2, 3, 4] use visual features, such as motion, face, and saliency. Thus, the performance of these methods relies on the accuracy of the algorithm in extracting visual features and the existence of visual features in the video. For example, the performance of the method proposed by Ma et al. [2] which is based on facial features depends on a face detection algorithm and on whether or not a face is in the video. Story based methods [5, 6] assume that there is a storyline associated with the input video, and they try to capture the storyline. Thus, these methods are only applicable to those videos that satisfy the assumption. Auxiliary information based methods [7, 8, 9, 10] use annotation, title, and category available from websites for summarization. Again, these methods are only applicable to those videos with available text information. Thus, if large text data does not cover the input video, these methods can show low performance. In addition, the method proposed by Song et al. [8] finds images similar to those in video scenes using text information. However, web videos often have incorrect or vague titles. Thus, it can be difficult to extract text information perfectly. Additionally, a deep neural network based method [11] has been proposed for video summarization. However, this method requires a large database of videos and its summary. Manual summaries of many subjects are difficult to collect.

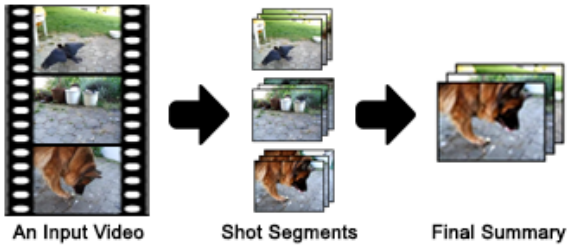

Fig. 1. Overview of video summarization.

Our method comprises three stages. First, the input video is divided into many shots. Second, motion magnitude, object class prediction, and saturation are obtained for each shot. Finally, the sum of the normalized features per shot are ranked in descending order, and the summary is determined by the two ranking approaches. A diagram of our algorithm is shown in Fig. 2. The contribution of our method is that it is able to adaptively summarize a video based on its contents.

This work was partly supported by the ICT R&D program of MSIP/IITP [2016-0-00563, Research on Adaptive Machine Learning Technology Development for Intelligent Autonomous Digital Companion] and partly supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2017R1A2B2006165).

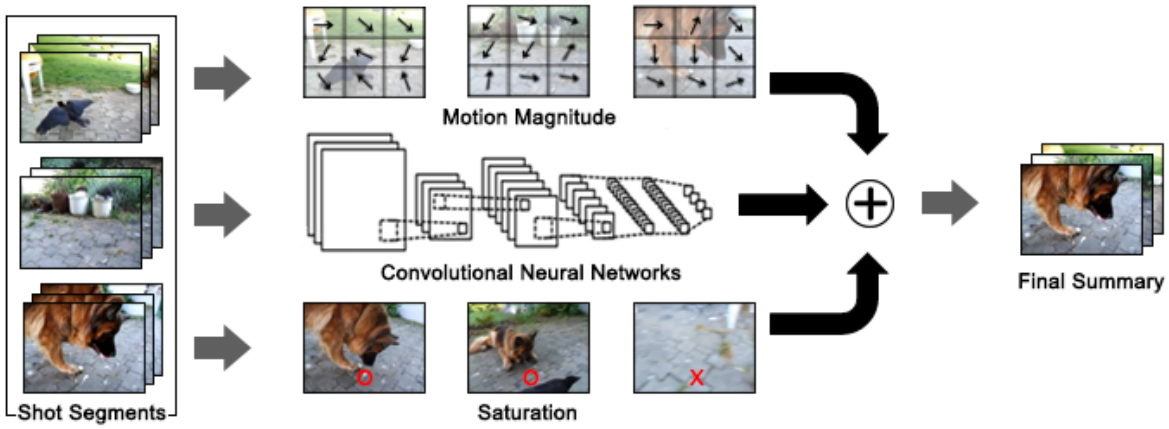

Fig. 2. Overview of proposed algorithm.

2. CONTENT ADAPTIVE VIDEO SUMMARIZATION USING SPATIO-TEMPORAL FEATURES

2.1. Shot Segmentation

The first stage for video summarization is dividing an input video into lots of shots. Many approaches were suggested by many researchers. Basically, lots of methods divide a video into shots have fixed length. However, these do not guarantee natural scene change on the video because these do not consider the motion on the scenes. Thus, we utilize temporal shot segmentation based on the motion. It was proved that this approach was better than other methods because it was possible to make smooth shot boundary [12].

2.2. Temporal Feature: Motion Magnitude

Our algorithm uses motion magnitude per frame as a temporal feature. For this purpose, we utilize Features from Accelerated Segment Test (FAST) algorithm to find feature points set, and estimate motion vectors by calculating forward and backward motions using the feature points set. The forward motions mean the difference between motion vectors of present and next frame. The backward motions mean the difference between motion vectors of present and past frame. Conclusively, we calculate average between forward and backward motions, and use it for motion magnitude of the present frame. Then, if frames of the i^{th} shot are $X_i = \{x_0, x_1, ..., x_n\}$, the motion magnitude $M(\cdot)$ for the i^{th} shot s_i is as follows:

$$M(s_i) = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{F_s}{n_p \delta (x_i^W \times x_i^H)}\sum_{j=0}^{n_P}(P_j^N - P_j^O)\right), \quad (1)$$

where n is the number of frames on the shot, δ is the step size for frame skipping, x_i^W and x_i^H are width and height of the i^{th} frame respectively, and P_j^N is the j^{th} feature points set in a new frame. The new frame can be either next or past frame. P_j^O is the j^{th} feature points set in a present frame, F_s is frames per second, n_P is the number of extracted features. We set step size δ as 5 in our implementation.

2.3. Spatial Feature: Saturation

Generally, frames have an achromatic color on the video are considered as useless frames. Thus, we utilize saturation of frames on the video as spatial features to ensure that frames of our summary have vivid color. We calculate saturation per frame on the video by modifying an equation suggested by Hasler et al. [13]. If frames of the i^{th} shot are $X_i = \{x_0, x_1, ..., x_n\}$, we define saturation $C(\cdot)$ for the i^{th} shot s_i as follows:

$$C(s_i) = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{\sqrt{\sigma_{x_i^{rg}}^2 + \sigma_{x_i^{yb}}^2} + 0.3\sqrt{\mu_{x_i^{rg}}^2 + \mu_{x_i^{yb}}^2}}{x_i^W \times x_i^H}\right), \quad (2)$$

where n is the number of frames on the shot, $\sigma_{x_i^{rg}}$ and $\sigma_{x_i^{yb}}$ are the standard deviations of pixels on the RG and YB color space respectively. $\mu_{x_i^{rg}}$ and $\mu_{x_i^{yb}}$ are the means of pixels on the RG and YB color space respectively.

2.4. Spatial Feature: Object Class Prediction

We use object class prediction from Convolutional neural networks (CNNs) as spatial features. The object class prediction is composed of prediction labels and accuracies. We utilize the accuracy corresponds to the top-1 label from CNNs only for shot importance scoring. For the prediction, we extract key frames on the video. A key frame is defined as the center frame between the start frame and the end frame of a shot boundary. Key frames are used as the input of CNNs to make our approach fast and effective. To recognize scenes on the video, CNNs are referred to as Inception-v3 reaches top-5 error rate 3.46% [14].

We test our input videos using the pre-trained Inception-v3 model. The input of this model is key frames extracted

Algorithm 1 The ranking algorithm for maximizing score

Input: User-defined length L_U
Output: The final set S^*

```
Find the number of prediction labels N.
while L <= L_U do
    Find candidate shots S_C have maximum score.
    Sort candidate shots S_C found in descending order.
    for j = 1 : N do
        Fill the final set S* with a candidate shot S_{C_j}.
        Update the final set length L, and check L > L_U.
    end for
end while
```

from shots, and the output of this model is prediction labels include recognized various objects and accuracies which denote recognition rates. If the key frames of the i^{th} shot are $K_i = \{k_0, k_1, ..., k_n\}$, the accuracy $P(\cdot)$ for the i^{th} shot s_i has the key frame are inferred through the feed-forward network as follows:

$$P(s_i) = \sigma(\sum_i \omega_i k_i + b), \tag{3}$$

where ω_i is the i^{th} weight of the CNNs, b is the bias terms, and $\sigma(\cdot)$ is the activation function for non-linearity.

2.5. Shot Importance Scoring

We combine temporal features into spatial features for generating final summaries by normalizing. Thus, we define a score function for the i^{th} shot s_i which denotes weighted sum among the motion magnitude, saturation, and object class prediction as follows:

$$S(s_i) = P(s_i) + \lambda_1 M(s_i) + \lambda_2 C(s_i), \tag{4}$$

where $P(\cdot)$ is the accuracy from CNNs, $M(\cdot)$ is the motion magnitude, $C(\cdot)$ is the saturation, λ_1 and λ_2 are weight parameters controls the influence among the accuracy, motion magnitude, and saturation. We found optimal weights through parameter tuning in the SumMe dataset [12] as follows. λ_1 is always 2.0, and if the input video has multi-scene which will be explained in the next paragraph, λ_2 is 2.0, and if not, λ_2 is 1.0 in our implementation.

2.6. Generating Content Adaptive Video Summary

For generating summaries, we classify videos into two types because we assume optimal summary depends on the number of scenes on the video. If the input video has lots of scenes, we should summarize it to be able to include every scene. On the contrary, if the input video has one or two scenes, summarizing it based on score function in order is better than containing every scene. Thus, we define two conditions to decide whether the input video has multi-scene or not. If $\sigma_P < T_P$ and $\mu_M > T_M$ are both satisfied, we regard the input as the video has multi-scene. In these two conditions, σ_P is the standard deviation for the frequency rate of each prediction label, μ_M is the mean for all motion magnitude, T_P and T_M are averages of σ_P and μ_M respectively in the SumMe dataset [12]. Thus, T_P and T_M are 0.17 and 0.38 respectively in our implementation.

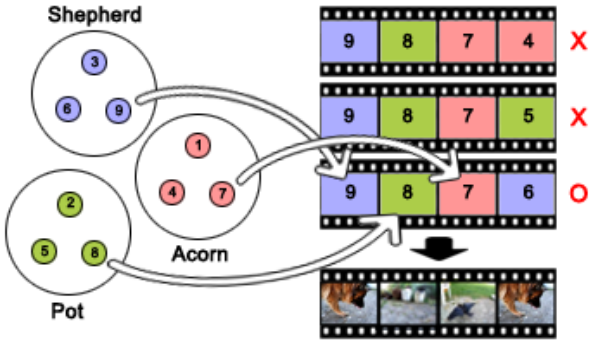

Fig. 3. This denotes an example of the ranking algorithm for maximizing the shot importance score in case of the video has multi-scene. We assume there are four blanks in the final set, and we should fill the blank with shots. Circles have the same color on the left side mean shots have same labels. Moreover, the numbers in the circles mean the score.

We define two methods to extract important shots by deciding whether the input video has multi-scene or not. If it has not multi-scene, we simply put shots in the final set based on the score function in order of score. If it has multi-scene, we find an optimal set which both has every prediction label and maximizes the sum of score per each prediction label on the final set as shown in Alg. 1. First, we extract unused shots have maximum score per each prediction label. Then, we sort the shots in descending order of the score. In addition, we put the sorted shots in the final set one by one as shown in Fig. 3. Shots stored in the final set are marked as used shots. We repeat these stages until final summaries length exceeds user-defined length. After finishing these stages, the final set rearranges shots in temporal order.

Conclusively, we generate a final video summary has user-defined length L_U by regarding two procedures mentioned earlier as an optimization problem as follows:

$$S^* = \arg\max_S \sum_{i=1}^{n} u_i S(s_i) \quad s.t. \quad \sum_{i=1}^{n} u_i L_i < L_U, \tag{5}$$

where $u_i \in \{0, 1\}$, n is the number of shots, $S(s_i)$ is the shot importance score of the i^{th} shot, and L_i is the length of the i^{th} shot. As a result, the video summary is generated by concatenating shots with $u_i = 1$ in temporal order. Practically, user-defined length is decided as lower value compared to the input video length.

3. EXPERIMENTS

To evaluate the performance of our algorithm, we use SumMe dataset [12] include 25 videos have a variety of scenes. Additionally, it includes manual summaries of videos edited by human. Thus, we compared our approach with the result of manual summaries. In addition, we evaluated our method compared to other baseline methods already suggested [1, 4, 12].

3.1. Evaluation Metric

We evaluate our algorithms using the pairwise F-measure F_i of human selection i as follows. It was used for quantitive evaluation of many video summarization methods [1, 8, 15]:

$$F_i = \frac{1}{N} \sum_{i=1}^{n} \frac{2 \times p_i \times r_i}{p_i + r_i}. \quad (6)$$

where N is the number of human participants, p_i is the precision, and r_i is the recall of human selection i using a ground truth.

3.2. Baseline Methods

We compare our method with five baseline methods from manual editing method by the human to several computational methods. We limit our video summaries length to 15% of the input video length to evaluate ours objectively.

Manual editing: It measured the mean F-measure of one human to all the others. The value of worst human means the mean F-measure of a video summary which is least similar to other summaries. We use the mean F-measure for the worst human as baseline [12].

Uniform: This method selects shots on the video based on uniform distribution.

Clustering: It divides the video into unique temporal events. Then, it clusters scenes which have similar color distributions.

Attention: This approach extract key frames from the input video using temporal and spatial saliency map. We use video summaries made by key frames [1].

Interestingness: This approach divides a video into the superframes which mean short clip. Then, interestingness is computed using various features such as attention, colorful, and landmark [12].

3.3. Results

The mean F-measure of our method was 0.214 which outperformed 0.179 for the summary edited by the worst human described in Gygli et al. [12]. Moreover, we achieved better performance compared to other computational methods except for interestingness method. However, it is a computationally expensive method because it uses eight features by considering both low-level features such as the contrast and saliency, and high-level features such as landmark and face for computation of interestingness. However, our method considers only the motion magnitude, object class prediction, and saturation. In other words, our method is more efficient than interestingness method because we use much less features. Table 1 presents the quantitive evaluation of our algorithm using the pairwise F-measure. The best and second best F-measure in our method are presented in bold. In the result, we achieved better or competitive performance on 13 videos of all datasets have 25 videos compared to other state-of-the-art video summarization methods.

Table 1. The quantitive evaluation of our algorithm

Video Name	Uni.	Clu.	Att.[1]	Int.[12]	Ours
Air Force One	0.161	0.143	0.215	0.318	**0.321**
Base Jumping	0.168	0.109	0.194	0.121	**0.203**
Bearpack Climbing	0.152	0.158	0.227	0.118	**0.173**
Bike Polo	0.058	0.130	0.076	0.356	**0.205**
Bus in Rock Tunnel	0.124	0.102	0.112	0.135	**0.158**
Car over Camera	0.099	0.296	0.201	0.372	**0.374**
Car Railcrossing	0.146	0.146	0.064	0.362	0.105
Cockpit Landing	0.129	0.156	0.116	0.172	**0.223**
Cooking	0.171	0.139	0.118	0.321	**0.253**
Eiffel Tower	0.166	0.179	0.136	0.295	**0.249**
Excavators River C.	0.131	0.163	0.041	0.189	0.121
Fire Domino	0.233	0.349	0.252	0.130	0.081
Jumps	0.052	0.298	0.243	0.427	**0.513**
Kids Playing in Leaves	0.209	0.165	0.084	0.089	0.097
Notre Dame	0.124	0.141	0.138	0.235	0.133
Paintball	0.109	0.198	0.281	0.320	**0.350**
Paluma Jump	0.132	0.072	0.028	0.181	0.058
Playing Ball	0.179	0.176	0.140	0.174	0.137
Playing on Water Slide	0.186	0.141	0.124	0.200	0.148
Saving Dolphines	0.165	0.214	0.154	0.145	0.115
Scuba	0.162	0.135	0.200	0.184	0.171
St Maarten Landing	0.092	0.096	0.419	0.313	**0.505**
Statue of Liberty	0.143	0.125	0.083	0.192	0.142
Uncut Evening Flight	0.122	0.098	0.299	0.271	**0.303**
Valparaiso Downhill	0.154	0.154	0.231	0.242	0.220
Mean	0.143	0.163	0.167	0.234	**0.214**

4. CONCLUSIONS

This paper proposed a video summarization method based on novel spatio-temporal features that combine motion magnitude, object class prediction, and saturation. For the sum of the normalized features per shot, shots were ranked in descending order, and the summary was determined by the highest ranking shots. In particular, the ranking can be conditioned on the object class, and high-ranking shots for different object classes was proposed as a summary of the input video has multi-scene. In other words, our algorithm can be used on both single scene and multi-scene videos. Our experiments show that our method achieved competitive performance in comparison to five baseline methods. In future work, we will consider using deep features derived from an activation map by the deep neural network to generate a summary.

5. REFERENCES

[1] Ejaz, Naveed, Irfan Mehmood, and Sung Wook Baik, "Efficient visual attention based framework for extracting key frames from videos," in *Signal Processing: Image Communication*, 2013, pp. 34–44.

[2] Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li, "A user attention model for video summarization," in *Proceedings of the tenth ACM international conference on Multimedia*, 2002, pp. 533–542.

[3] Marat Sophie, Mickael Guironnet, and Denis Pellerin., "Video summarization using a visual attention model," in *Signal Processing Conference, 2007 15th European*, 2007, pp. 1784–1788.

[4] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman., "Discovering important people and objects for egocentric video summarization," in *Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1346–1353.

[5] Zheng Lu and Kristen Grauman., "Story-driven summarization for egocentric video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2714–2721.

[6] Bo-Wei Chen, Jia-Ching Wang, and Jhing-Fa Wang, "A novel video summarization based on mining the story-structure and semantic relations among concept entities," in *IEEE Transactions on Multimedia*, 2009, pp. 295–392.

[7] Meng Wang, Richang Hong, Guangda Li, Zheng-Jun Zha, Shuicheng Yan, and Tat-Seng Chua, "Event driven web video summarization by tag localization and key-shot identification," in *IEEE Transactions on Multimedia*, 2012, pp. 975–985.

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

[9] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan, "Large-scale video summarization using web-image priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2698–2705.

[10] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid, "Category-specific video summarization," in *European conference on computer vision*, 2014, pp. 540–555.

[11] Ting Yao, Tao Mei, and Yong Rui, "Highlight detection with pairwise deep ranking for first-person video summarization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 982–990.

[12] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool, "Creating summaries from user videos," in *European conference on computer vision*, 2014, pp. 505–520.

[13] David Hasler and Sabine E. Suesstrunk, "Measuring colorfulness in natural images," in *Electronic Imaging*, 2003, pp. 87–95.

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[15] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman, "Video summarization with long short-term memory," in *European Conference on Computer Vision*, 2016, pp. 766–782.