Highlight Detection with Pairwise Deep Ranking for First-Person Video Summarization

Ting Yao, Tao Mei, and Yong Rui
Microsoft Research, Beijing, China
{tiyao, tmei, yongrui}@microsoft.com

Abstract

The emergence of wearable devices such as portable cameras and smart glasses makes it possible to record life logging first-person videos. Browsing such long unstructured videos is time-consuming and tedious. This paper studies the discovery of moments of user's major or special interest (i.e., highlights*) in a video, for generating the summarization of first-person videos. Specifically, we propose a novel pairwise deep ranking model that employs deep learning techniques to learn the relationship between highlight and non-highlight video segments. A two-stream network structure by representing video segments from complementary information on appearance of video frames and temporal dynamics across frames is developed for video highlight detection. Given a long personal video, equipped with the highlight detection model, a highlight score is assigned to each segment. The obtained highlight segments are applied for summarization in two ways: video timelapse and video skimming. The former plays the highlight (non-highlight) segments at low (high) speed rates, while the latter assembles the sequence of segments with the highest scores. On 100 hours of first-person videos for 15 unique sports categories, our highlight detection achieves the improvement over the state-of-the-art RankSVM method by 10.5% in terms of accuracy. Moreover, our approaches produce video summary with better quality by a user study from 35 human subjects.*

1. Introduction

Wearable devices have become pervasive. People are taking first-person videos using these devices everyday and everywhere. For example, wearable camcorders such as GoPro cameras and Google Glass are now able to capture high quality first-person videos for recording our daily experience. These first-person videos are usually extremely unstructured and long-running. Browsing and editing such videos is really a tedious job. Therefore, video summarization, which produces a short summary of a full-length video and ideally encapsulates its most informative parts, is becoming increasingly important for alleviating the problem of first-person video browsing, editing and indexing.

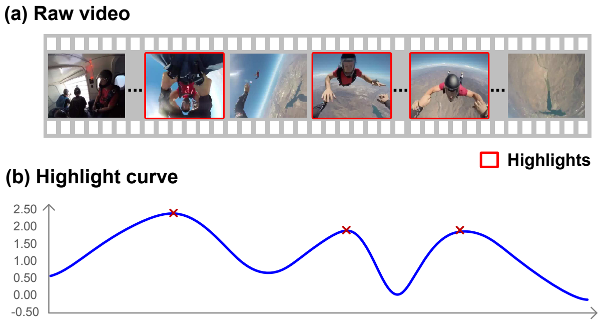

Figure 1. Retrieving highlights from unconstrained first-person videos: (a) raw video, where each segment is represented by a frame sampled from the segment; (b) highlight prediction score curve. The segments with high scores are selected as highlights in red bounding boxes.

The research on video summarization has mainly proceeded along two dimensions, i.e., keyframe or shot-based [15, 18], and structure-driven [17, 22] approaches. The keyframe or shot-based method always selects a collection of keyframes or shots by optimizing diversity or representativeness of a summary, while structure-driven approach exploits a set of well-defined structures in certain domains (e.g., audience cheering, goal or score events in sports videos) for summarization. In general, existing approaches offer sophisticated ways to sample a condensed synopsis from the original video, reducing the time required for users to view all the contents.

However, defining video summarization as a sampling problem in conventional approaches is very limited as users' interests in a video are fully overlooked. As a result, the special moments may be omitted due to the visual diversity criteria of excluding redundant parts in a summary. The limitation is particularly severe when directly applying those methods to first-person videos. First-person videos captured with wearable devices record experiences from a first-person perspective in unconstrained environments, making them long, redundant and unstructured. Moreover, the continuous nature of such videos even yields no evident shot boundaries for summary; nevertheless, there should be the moments (segments) of major or special interest (i.e. *highlights*) in raw videos. Therefore, our goal is to provide a new paradigm for first-person video summarization by exploring the highlights in a video.

As the first step towards this goal, Figure 1 demonstrates the process of retrieving highlights in raw first-person videos. A raw video is divided into several segments. The highlight measure for each segment is equivalent to learning a function to predict the highlight score given the representations of the segment. The higher the score, the more highlighted the segment. Thus, the segments with high scores can be selected as video highlights. Furthermore, in order to incorporate both spatial and temporal information for better depicting a video segment, complementary streams on visual appearance from static frames and temporal dynamics across multiple frames are jointly exploited. As such, we devise a two-stream deep convolution neural networks (DCNN) architecture by fusing DCNN on each stream for video highlight detection. In particular, considering that highlight score expresses only a relative degree of interest within each video, we propose to train DCNN on each stream independently with a pairwise deep ranking model, which characterizes the relative relationships by a set of pairs. Each pair contains a highlight and a non-highlight segment from the same video. The DCNN on each stream aims to optimize the function making the detection score of highlight segment higher than that of non-highlight segment.

Then, by assigning a highlight score to each segment, the highlight-driven summary of a video can be generated in two ways: video timelapse and video skimming. The former keeps all the segments in the video while adjusting their speed rates of playing based on highlight scores (highlight segments with lower playing rate, and vice versa). The latter assembles the sequence of only highlight segments while trimming out the other non-highlight ones. We evaluate both video highlight detection and highlight-driven video summarization on a newly created dataset including about 100 hours of first-person videos captured by GoPro cameras for 15 sport categories, which is so far the largest scale first-person video dataset.

The remaining sections are organized as follows. Section 2 describes the related work. Section 3 presents the architecture of video highlight detection, while Section 4 formulates the problem of video summarization over the predicted video highlight. In Section 5, we provide empirical evaluations on both video highlight detection and video summarization, followed by the discussions and conclusions in Section 6.

2. Related Work

The research area of first-person video summarization is gaining an increasing amount of attention recently [6, 10, 13, 14, 21]. The objective of video summarization is to explore the most important parts of long first-person video sequences. In [13], a short subsequence of the video was selected by using the importance of the objects as the decision criteria. Similar in spirit, video subshots which depict the essential events were concatenated to generate the summary [14]. Later in [6] and [21], Gygli *et al.* and Potapov *et al.* formulated the problem as scoring each video segment in terms of visual importance and interestingness, respectively. Then the summary was produced by selecting the segments with highest scores. Recently, Joshi *et al.* created a stabilized time-lapse video by rendering, stitching and blending appropriately selected source frames for each output frame [10]. In contrast, our approach explores the moments of user interest in the videos, which we show is vital to distill the essence in the original videos.

In addition to first-person video summarization, there is a large literature on summarization for general videos. Keyframe or shot-based methods use a subset of representative keyframes or segments from the original video to generate a summary. In [15], keyframes and video shots were sampled based on their scores of attention, which are measured by combining both visual and aural attention. Similar in spirit, Ngo *et al.* presented a video as a complete undirected graph, which is partitioned into video clusters to form a temporal graph and further detect video scenes [18]. Video summarization can be generated from the temporal graph in terms of both the structure and attention information. Later in [16], subshots were first detected and classified into five categories according to the dominant camera motion. Then a number of representative keyframes, as well as structure and motion information were extracted from each subshot to generate the video summary.

Different from keyframe or shot-based methods, structure-driven approaches exploit video structure for summarization. The well-defined structure often exists in broadcast sports videos. A long sports game can be divided into parts and only a few of these parts contain certain informative segments. For instance, these segments include the score moment in soccer games or the hit moment in baseball games. Based on the well-defined structure, specifically designed audio-visual features, such as crowds, cheering, goal, etc., are used in the structure-driven methods [17, 22].

Most of the above methods focus on selecting frames, shots or segments independently, ignoring the relationship between them. Our work is different that we claim to learn the relationship of video segments in a pairwise manner, which characterizes the relative preferences of all the segments within a video and will benefit video summarization.

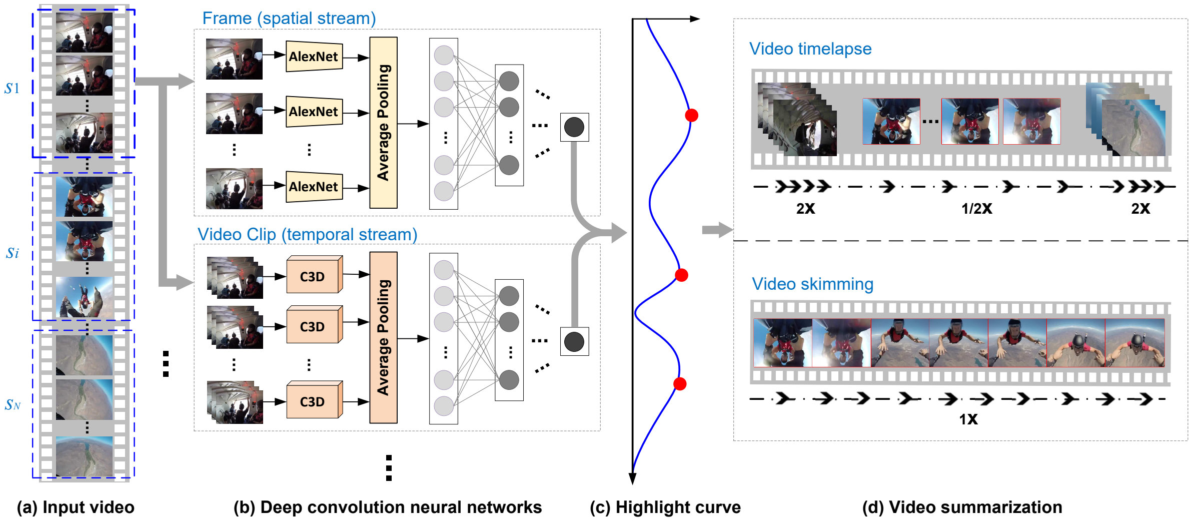

Figure 2. Highlight-driven video summarization framework (better viewed in color). (a) The input video is split into a set of segments. (b) Each video segment is decomposed into spatial and temporal streams. The spatial stream is in the form of multiple frame appearance, while the temporal stream is represented by temporal dynamics in a video clip. A deep convolution neural networks architecture for highlight prediction is devised for spatial and temporal stream, respectively. The output of the two components are combined by late fusion as the final highlight score for each video segment. (c) By assigning a highlight score to each video segment, a highlight curve can be obtained for each video. The segments with highest highlight scores are regarded as "highlights" in the video. (d) Two highlight-driven video summarization methods, i.e., video timelapse and video skimming, can be easily formulated.

3. Video Highlight Detection

In this section, we first present our highlight detection for each video segment by combining two deep convolution neural networks architectures on spatial and temporal stream, followed by the pairwise deep ranking model for the training of each DCNN structure.

3.1. Two-Stream DCNN for Highlight Detection

Video can be naturally decomposed into spatial and temporal components, which are related to ventral and dorsal streams for human perception respectively [4, 23]. The ventral stream plays the major role in the identification of objects, while the dorsal stream mediates the sensorimotor transformations for visually guided actions at such objects. Therefore, we devise a novel two-stream DCNN architecture (TS-DCNN) by late fusing spatial and temporal DC-NN for video highlight detection, as shown in Figure 2 (a)-(c). The spatial component depicts scenes and objects in the video by frame appearance while the temporal part conveys the temporal dynamics in a video clip (multiple frames).

Given an input video, a set of video segments can be delimited by uniform partition in temporal, shot boundary detection, or change point detection. For each video segment, spatial DCNN operates on multiple frames extracted from the segment. The static frame appearance is useful as some highlights are strongly associated with particular scenes and objects. The first stage of the architecture is to generate a fixed-length visual representation for each video segment. For this purpose, AlexNet [12], which is the recent advance image classification architecture, is exploited for extracting the softmax scores for multiple frames. Then, an average pooling [1] is performed over all the frames to get a single 1,000 dimensional vector for each video segment. The AlexNet is pre-trained on 1.2 million images of ImageNet challenge dataset [2]. The resulting 1,000 dimensional representation of video segment forms the input to a following neural networks for predicting the highlight score of this segment. The architecture of this neural networks is $F1000-F512-F256-F128-F64-F1$, which contains six fully-connected layers (denoted by F with the number of neurons). The output of the last layer is taken as the highlight score.

Unlike the spatial DCNN, the inputs to temporal DCNN architecture are comprised of multiple video clips and each video clip contains multiple frames. Such inputs explicitly describe the temporal dynamics between frames. To generate the representations of each video clip, 3D CNN is utilized. Different from traditional 2D CNN, 3D CNN architecture takes video clip as the inputs and consists of alternating 3D convolutional and 3D pooling layers, which are further topped by a few fully connected layers as described in [8]. Specifically, C3D [25], which is pre-trained on Sports-1M video dataset [11], is exploited and we regard the outputs of the fc6 fully-connected layer as representations for each video clip. Similar to spatial DCNN architecture, temporal DCNN fuses the outputs of C3D on each video clip, followed by importing into a neural networks for video highlight detection.

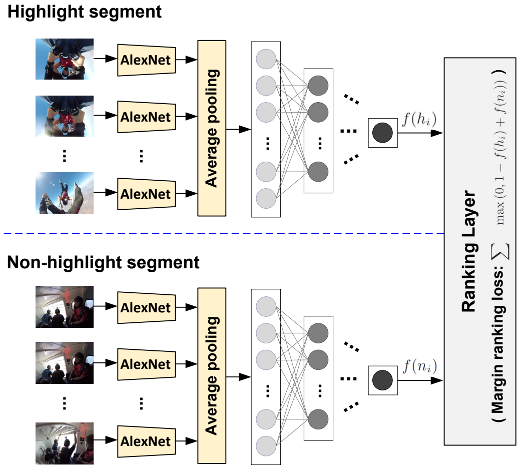

Figure 3. The training of spatial DCNN architecture with pairwise deep ranking model. The inputs are a set of highlight and non-highlight video segment pairs, which are fed independently into two identical spatial DCNN with shared architecture and parameters. A ranking layer is on the top to evaluate the margin ranking loss of the pair. Note that the training of temporal DCNN follows the same philosophy.

By late fusing the two predicted highlight scores of spatial and temporal DCNN, we can obtain the final highlight score for each video segment and form a highlight curve for the whole video. The video segments with high scores are selected as video highlights accordingly. It is worth noticing that although the two streams used here are visual appearance and temporal dynamics, our approach is applicable to include any other stream, e.g., audio stream.

3.2. Pairwise Deep Ranking Model

As with most deep learning problems, the learning of our spatial and temporal DCNN architectures are critical for video highlight detection. Existing deep learning models for visual recognition often focus on learning category-level representation [9, 12]. The learnt representations mainly correspond to visual semantics. Instead, in our case, a highlight score for every video segment reflects its degree of interest within a video and represents a relative measure. It is a straight forward way to formulate it as a supervised ranking problem. More importantly, the intrinsic property between visual recognition and ranking tasks is different, as visual recognition is modeled as a binary classification problem while ranking task is considered as a regression problem. As such, a good network for visual recognition may not be optimal for distinguishing video highlights.

Deriving from the idea of exploring relative relationship through ranking [27, 28], we develop a pairwise deep ranking model to learn our spatial and temporal DCNN architectures for predicting video highlights. Figure 3 shows the training of spatial DCNN with pairwise deep ranking model. Given a pair of highlight and non-highlight video segments, we wish to optimize our spatial DCNN architecture, which could output a higher score of highlight segment than that of non-highlight one. Formally, suppose we have a set of pairs $\mathcal{P}$, where each pair (h_i, n_i) consists of a highlight video segment h_i and a non-highlight segment n_i from an identical video. The two segments are fed separately into two identical spatial DCNN with shared architecture and parameters. A pair characterizes the relative highlight degree for the two video segments. The output $f(\cdot)$ of the spatial DCNN computes the highlight score of the input video segment. Our goal is to learn the DCNN architecture that assigns higher output score to the highlight segment, which can be expressed as

$$f(h_i) \succ f(n_i), \quad \forall\, (h_i, n_i) \in \mathcal{P}. \tag{1}$$

As the output scores exhibit a relative ranking order for the video segments, a ranking layer on the top is employed to evaluate the margin ranking loss of each pair, which is a convex approximation to the 0-1 ranking error loss and has been used in several information retrieval methods [20, 29]. Specifically, it can be given by

$$\mathit{min}: \sum_{(h_i, n_i) \in \mathcal{P}} \max\left(0, 1 - f(h_i) + f(n_i)\right). \tag{2}$$

The ranking layer does not have any parameters. During learning, it evaluates the model's violation of the ranking order, and back-propagates the gradients to the lower layers so that the lower layers can adjust their parameters to minimize the ranking loss. To avoid overfitting, dropout [7] with a probability of 0.5 is applied to all the fully-connected layers after AlexNet in our architecture.

The process of temporal DCNN training is the same as spatial DCNN. After training, the learnt spatial and temporal DCNN architectures are late fused for video highlight detection as shown in Figure 2 (b).

4. Highlight-driven Video Summarization

After we get the highlight score for each video segment, how to use them for video summarization? Two video summarization approaches, i.e., video timelapse and video skimming (Figure 2 (d)), will be easily formulated.

4.1. Video Timelapse

A simple and robust technique for video summarization is timelapse, i.e., increasing the speed of the non-highlight video segments by selecting every r^{th} frame and showing the highlight segments in slow motion. Particularly, as all the segments are included finally and thus there is no strict demand on video segmentation, we simply divide the video into segments evenly in this case rather than analyzing video content. Let L_v, L_h and L_n be the length of original video, highlight segments and non-highlight segments, respectively. Typically we have $L_h \ll L_n, L_v$. Without loss of generality, we consider the case when the rate of decelerating highlight segments and speeding up non-highlight parts is the same and denote r as the rate. Given a maximum length of summary L, the problem is then equivalent to find a proper rate r which satisfies the formula as

$$rL_h + \frac{1}{r}L_n \leq L. \quad (3)$$

Since $L_h + L_n = L_v$, we can derive that: $r = \left\lfloor \frac{L}{2L_h} + \sqrt{Y} \right\rfloor$, where $Y = \frac{L^2 - 4L_vL_h + 4L_h^2}{4L_h^2}$.

In this way, we generate a video summary by compressing the non-highlight video segments while expanding the highlight parts. In general, video timelapse has two major characteristics. First, all the video content are contained in the summary. As a result, there is no risk of omitting any important segments, making the summary more coherent and continuous in telling camera wearer's story. Furthermore, the video segments of interest are underlined and presented in detail.

4.2. Video Skimming

Video skimming addresses the summarization problem by providing a short summary of original video which ideally includes all the important video segments. A common practice to video skimming is to first perform a temporal segmentation, followed by singling out a few segments to form an optimal summary in terms of certain criteria, e.g., interestingness and importance. Following [21], we exploit a kernel temporal segmentation approach which is originated from the idea of multiple change point detection. Readers can refer to [21] for more technical details.

After the segmentation, highlight detection is applied to each video segment, producing the highlight score. Given the set of video segments $\mathcal{S} = \{s_1, \ldots, s_c\}$ and each segment is associated with a highlight score $f(s_i)$, we aim to single out a subset with a length below the maximum length L and the sum of the highlight scores is maximized. Specifically, the problem is defined as

$$\max_{\mathbf{b}} : \sum_{i=1}^{c} b_i f(s_i) \quad s.t. \quad \sum_{i=1}^{c} b_i \left|s_i\right| \leq L \ , \quad (4)$$

where $b_i \in \{0, 1\}$ and $b_i = 1$ indicates that the i^{th} segment is selected. $|s_i|$ is the length of the i^{th} segment. The maximization is a standard $0/1$-knapsack problem and can be solved with dynamic programming for a global optimal solution [5].

5. Experiments

We conducted our experiments on a newly created first-person video dataset crawled from YouTube and evaluated our approaches on both video highlight detection and highlight-driven video summarization.

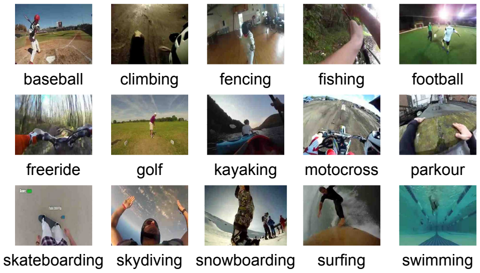

Figure 4. One representative frame selected from each category in our dataset. The category is given in the lower row.

5.1. Dataset

While the research on first-person video analysis has recently received intensive attention, the public datasets to date are still small (e.g., up to 20 hours) and very specific (e.g., in a kitchen). To substantially evaluate our approach, we collect a new large dataset from YouTube for first-person video highlight detection. The dataset consists of 100 hours videos mainly captured by GoPro cameras for 15 sports related categories. In particular, we query the YouTube database with "category name + GoPro" to retrieve relevant videos. Given the retrieved videos, those with visible edited traces, as with scene splicing or rendering, are removed manually. Hence, our dataset is constructed with only raw videos. Figure 4 shows a representative frame selected from each category in our dataset. For each category, there are about 40 videos, each with a duration between 2 to 15 minutes.

To evaluate our highlight results, we first split the video into a set of five-second segments evenly sampled across each raw video and ask multiple evaluators to label the highlight level of each segment. We invited 12 evaluators from different education backgrounds, including linguistics, physics, business, computer science, and design. All evaluators are outdoor sports enthusiasts and some of them are from local outdoor sports club. Each video segment was annotated on a three point ordinal scale: 3–highlight; 2–normal; 1–boring. To make the annotation as objective as possible, there are three labelers assigned to each video. Only video segments with their aggregate scores at or over 8 points were selected as "highlight." Note that obtaining these annotations was very time consuming. The labelers are requested to watch the whole video before assigning labels to each segment as the highlight is a relative judgement within a video. The dataset is partitioned into training and test sets evenly on all 15 categories for our experiments.

5.2. Highlight Detection

The first experiment was conducted on our first-person video dataset to examine how our spatial and temporal DCNNs work on highlight detection.

Compared Approaches. We compare the following approaches for performance evaluation:

(1) Rule-based model [16] (*Rule*). The video is first segmented into a series of shots based on color information. Each shot is then decomposed into one or more subshots by a motion threshold-based approach. The highlight score for each subshot is proportion to its length, giving that the longer subshot usually contains more informative content.

(2) Importance-based model [21]. A linear SVM classifier per category is trained to score importance (highlight) of each video segment. For each category, we use all the video segments of this category as positive examples and the video segments from the other categories as negatives. We adopt both improved dense trajectories motion features proposed in [26] and the average of DCNN frame features for representing each video segment. The detailed settings will be presented in parameter settings. The two runs based on improved dense trajectory and DCNN are named as *Imp+IDT* and *Imp+DCNN*, respectively.

(3) Latent ranking model [24]. A latent linear ranking SVM model per category is trained to score highlight of each video segment. For each category, all the highlight and non-highlight video segment pairs within each video are exploited for training. Similarly, improved dense trajectories and the average of DCNN frame features are extracted as the representations of each segment. We refer to the two runs as *LR+IDT* and *LR+DCNN*, respectively.

(4) Deep Convolution Neural Networks model. We designed three runs for our proposed approaches: *S-DCNN*, *T-DCNN* and *TS-DCNN*. The two runs *S-DCNN* and *T-DCNN* predict the highlight score of video segment by separately using spatial DCNN and temporal DCNN, respectively. The result of *TS-DCNN* is the weighted summation of *S-DCNN* and *T-DCNN* by late fusion.

Parameter Settings. In the experiments, we uniformly pick up three frames every second and hence have 15 frames for each five seconds' video segment. Following [25], each video clip is composed of the first 16 continuous frames in every second and then have 5 video clips for each segment. For *S-DCNN* and *T-DCNN* training, only the segment pairs, the difference of whose aggregate scores is over 3 points, are selected and in total we have 105K pairs in training set.

To ensure the performance of these methods comparable, the representation of video segment is the average of the outputs of AlexNet on selected frames in both *Imp+DCNN* and *LR+DCNN*, which is the same as our *S-DCNN*. For the extraction of trajectory descriptor, we use the default parameters which results in 426 dimensions. The local descriptor is then reduced to half of the dimensions with PCA separately on each component of the descriptor. Finally, each video segment is encoded in a Fisher Vector [19] based on a GMM of 256 Gaussians. Furthermore, following the setting in [24], we use the liblinear package [3] to train both *LR+IDT* and *LR+DCNN* with the same stopping criteria: maximum 10K iterations and $\varepsilon = 0.0001$.

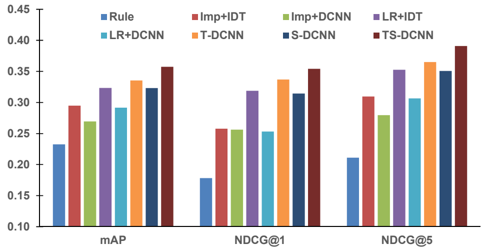

Figure 5. Performance comparison of different approaches for highlight detection.

Evaluation Metrics. We calculate the average precision of highlight detection for each video in test set and mean Average Precision (mAP) averaging the performance of all test videos is reported. In addition, since it is naturally to treat highlight detection as a problem of ranking segments in one video, we further adopt Normalized Discounted Cumulative Gain ($NDCG$) which takes into account the measure of multi-level highlight scores as the performance metric. Given a segment ranked list for a video, the $NDCG$ score at the depth of d in the ranked list is defined by: $NDCG@d = Z_d \sum_{j=1}^{d} \frac{2^{r^j}-1}{\log(1+j)}$, where $r^j = \{5 : as \geq 8; 4 : as = 7; 3 : as = 6; 2 : as = 5; 1 : as \leq 4\}$ represents the rating of a segment in the ground truth and as denotes the aggregate score of each segment. Z_d is a normalization constant and is chosen so that $NDCG@d = 1$ for perfect ranking. The final metric is the average of $NDCG@d$ for all videos in the test set.

Performance Comparison. Figure 5 shows the performances of eight runs averaged over all the test videos in our dataset. Overall, the results across different evaluation metrics consistently indicate that our *TS-DCNN* leads to a performance boost against others. In particular, the mAP of *TS-DCNN* can achieve 0.3574, which makes the improvement over *LR+IDT* by 10.5%. More importantly, the run time of *TS-DCNN* is less than *LR+IDT* by several dozen times and more details are given in the following run time section. Since *Rule* run is only based on the general subshot detection and without any highlight knowledge as a prior, it is not surprise that all the other methods exhibit significantly better performance than the *Rule* run.

There is a significant performance gap between *S-DCNN* (*T-DCNN*) and *LR+DCNN* (*LR+IDT*). Though both runs train the model in a pairwise manner, they are fundamentally different in the way that *S-DCNN* (*T-DCNN*) is by using a DCNN architecture, and *LR+DCNN* (*LR+IDT*) is based on ranking SVM techniques. The results basically indicate the advantage of exploiting DCNN architecture on highlight detection task. Furthermore, *LR+DCNN* (*LR+IDT*) exhibits better performance than *Imp+DCNN* (*Imp+IDT*) which formulates highlight detection as a binary classification problem by using linear SVM model. In addition, as observed in our results, using motion (temporal) features can constantly offer better performance than multiple stat-

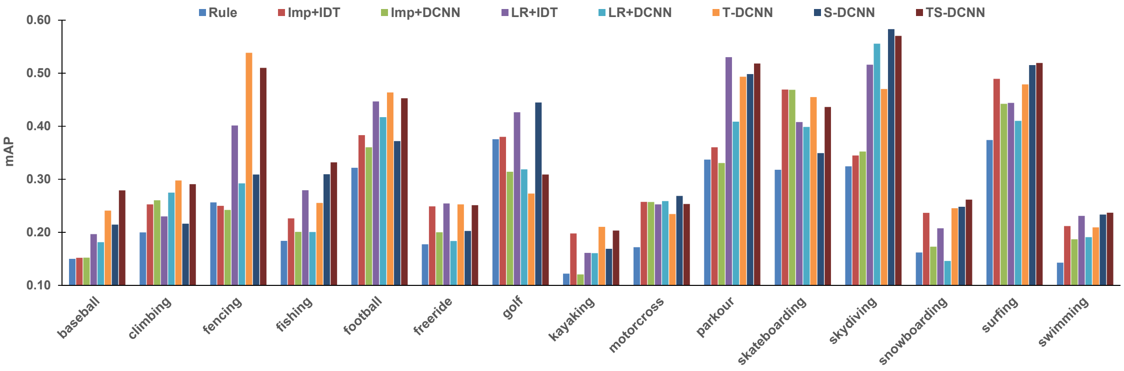

Figure 6. Per-category mAPs of different approaches for all the 15 categories.

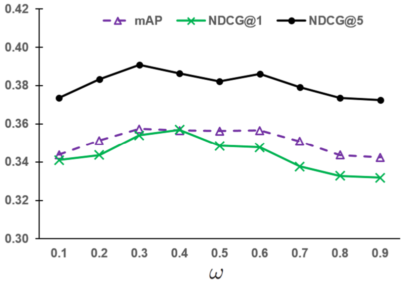

Figure 7. Performance of *TS-DCNN* by late fusing *S-DCNN* and *T-DCNN* with different ω.

ic frame appearances on all three models. This somewhat reveals that video highlights often appear in the segments with some special motions and hence they could be better represented by temporal features.

Figure 6 details the mAP performance across different categories. Different from importance-based model and latent ranking model which are category-specific, our model is general for all categories. Nevertheless, among the 15 categories, our *S-DCNN*, *T-DCNN* and *TS-DCNN* achieve the best performances for 11 categories, which empirically verify the merit of our model from the aspect of category independent. However, considering that all the 15 categories are sports related, it is still not clear that whether the proposed techniques are generalized to handle video contents from all domains. Moreover, the complementarity between *S-DCNN* and *T-DCNN* is generally expected. For instance, the videos of the category "fencing" is diverse in frame appearance, resulting in poor performance by *S-DCNN*. Instead, temporal dynamics is found to be more helpful for this category. In the case of category "golf" where motion is relatively few, visual features of frames show better performance.

Figure 8 further shows ten segments uniformly sampled from a video for "surfing," "skydiving," and "climbing." Each segment is represented by one sampled frame. As illustrated in the figure, the ten segments are ranked according to their predicted highlight scores by our *TS-DCNN* and we can easily see that the ranking order reflects the relative degree of interest within a video.

Table 1. Run time of different approaches on a five minutes' video. The experiments are conducted on a regular PC (Intel dual-core 3.33GHz CPU and 32 GB RAM).

App.	*Rule* [16]	*LR+IDT* [24]	*LR+DCNN*	*S-DCNN*	*TS-DCNN*
Time	25s	5h	65s	72s	360s

Fusion Parameter. A common problem with late fusion is the need to set the parameter to tradeoff *S-DCNN* and *T-DCNN*, i.e., ω in $\omega \times$ *S-DCNN* $+ (1-\omega) \times$ *T-DCNN*. In the previous experiments, ω was optimally set in terms of mAP performance. Furthermore, we conducted experiments to test the performance, when the values of ω are set from 0.1 to 0.9. Figure 7 shows the mAP, *NDCG@1* and *NDCG@5* performances with respect to different values of ω. We can see that the performance curves are relatively smooth and achieve the best result around $\omega = 0.3$. In general, this again confirms that *T-DCNN* leads to better performance gain and thus is given more weights in fusion.

Run Time. Table 1 listed the detailed run time of each approach on predicting a five minutes' video. Note that the run time of *LR+IDT* and *Imp+IDT*, *LR+DCNN* and *Imp+DCNN*, *T-DCNN* and *TS-DCNN* is the same respectively, only one of each is presented in the Table. We see that our method has the best tradeoff between performance and efficiency. Our *TS-DCNN* finishes in 360 seconds, which is slightly longer than the video duration.

5.3. Video Summarization

The second experiment was performed to evaluate our highlight-driven video summarization.

Compared Approaches. We compare the following approaches for performance evaluation:

(1) Uniform sampling (*UNI*). A simple approach by uniformly selecting K subshots throughout the video.

(2) Importance-driven summary [21] (*IMP*). Kernel temporal segmentation is first applied to video segmentation, then an importance score is assigned to each segment by *Imp+DCNN*. Finally, segments are included in the summary by the order of their importance.

(3) Interestingness-driven summary [6] (*INT*). The method starts by finding positions appropriate for a cut and

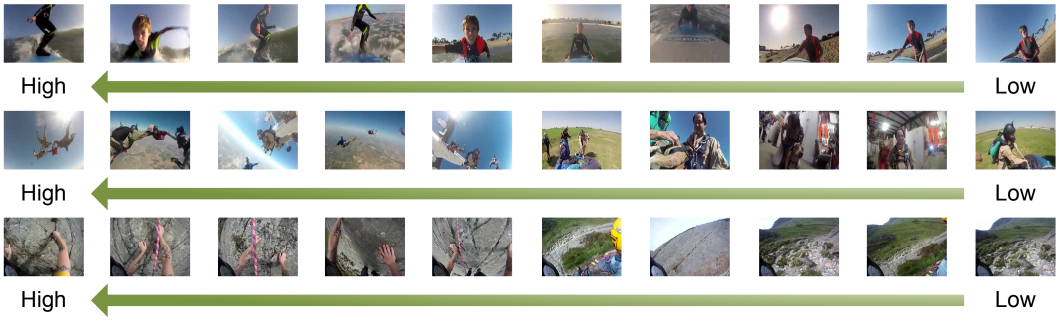

Figure 8. Examples of segments ranking from low (right) to high (left) according to our predicted highlight score for "surfing," "skydiving," and "climbing" categories. We uniformly sampled ten segments in a video and one frame is selected to represent each segment.

Table 2. Percentage of users who prefer the summary generated by *HD-VT* over each of other four approaches.

	UNI	*IMP* [21]	*INT* [6]	*HD-VS*
Coverage	91.4%	80.1%	74.3%	68.6%
Presentation	85.7%	60.2%	64.8%	34.3%

Table 3. Percentage of users who prefer the summary generated by *HD-VS* over each of other four approaches.

	UNI	*IMP* [21]	*INT* [6]	*HD-VT*
Coverage	87.2%	77.1%	71.4%	31.4%
Presentation	88.6%	74.3%	82.9%	65.7%

gets a set of segments. The interestingness score of each segment is the sum over the interestingness of its frames, which are jointly estimated by low-level (e.g., quality and saliency) and high-level (e.g., motion and person detection) features. Based on the interestingness score, an optimal subset of segments is selected to concatenate a summary.

(4) Highlight-driven summary. We designed two runs for our proposed highlight-driven video summarization approaches described in Section 4, i.e. *HD-VT* and *HD-VS*, which exploit video timelapse and video skimming techniques respectively.

Performance Comparison. We conduct subjective evaluation to compare the generated summaries. The evaluation process is as follows. First, all the evaluators are required to watch the original video. Then we show them once two summaries for that video. One is by *HD-VT* or *HD-VS* and the other is from the rest four runs. Note that we do not reveal which is ours and order the two summaries randomly. After viewing both, the evaluators are asked two questions: 1) *Coverage:* Which summary better covers the progress of the video? 2) *Presentation:* Which summary better distills and presents the essence of the video?

We randomly selected three videos in the test set from each of the 15 categories and the evaluation set is consisted of 45 original videos associated with five summaries for each. As only the comparisons between our methods and other three baselines are taken into account, we have 45×7 pairs of summaries to be tested in total. We invited 35 evaluators from different education backgrounds and they range from 20-52 years old. On each pair of compared approaches, the percentage of 35 evaluators' choices is averaged on all 45 videos and finally reported.

Table 2 and 3 show the statistics of our proposed *HD-VT* and *HD-VS*, respectively. Overall, a strong majority of the evaluators prefer the summaries produced by *HD-VT* and *HD-VS* over other three methods in terms of both *Coverage* and *Presentation* criteria. The results support our point that video highlights can distill the moments of user interest and thus better summarize the first-person videos. Compared to *HD-VS*, *HD-VT* achieves more preferences in *Coverage* as it keeps all the video content. *HD-VS*, in contrast, is benefited from the way of skimming the non-highlight segments and hence gets more votes in *Presentation*. Furthermore, *UNI* which concatenates the uniformly selected subshots is not informative for the long and unstructured first-person videos in general. Though both *IMP* and *INT* involve utilization of scoring each video segment for summarization, they formulate the scoring as a classification problem and tend to include more near-duplicate segments. *HD-VT* and *HD-VS*, in contrast, treat it as a pairwise ranking problem and have a better ability in differentiating each segment, and thus allow better summarization in terms of both *Coverage* and *Presentation*.

6. Discussion and Conclusion

We have presented a new paradigm of exploring the moments of user interest, i.e., highlights, for first-person video summarization. Particularly, we propose a category-independent deep video highlight detection model, which incorporates both spatial and temporal streams based on deep convolution neural networks. On a large first-person video dataset, performance improvements are observed when comparing to other highlight detection techniques such as linear SVM classifier and latent linear ranking SVM model, which are both category-specific. Furthermore, together with two summarization methods, we have developed our highlight-driven video summarization system. Our user study with 35 human subjects shows that a majority of the users prefer our summaries over both importance-based and interestingness-based methods.

References

[1] L. Cao, Y. Mu, A. Natsev, S. F. Chang, G. Hua, and J. Smith. Scene aligned pooling for complex video recognition. In *ECCV*, 2012.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[4] M. A. Goodale and A. D. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992.

[5] M. T. Goodrich and R. Tamassia. *Algorithm Design: Foundation, Analysis and Internet Examples*. John Wiley and Sons, 2006.

[6] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[7] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[8] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. on PAMI*, 35(1):221–231, 2013.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014.

[10] N. Joshi, W. Kienzle, M. Toelle, M. Uyttendaele, and M. F. Cohen. Real-time hyperlapse creation via optimal frame selection. In *SIGGRAPH*, 2015.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[13] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.

[14] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013.

[15] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang. A generic framework of user attention model and its application in video summarization. *IEEE Trans. on MM*, 7(5):907–919, 2005.

[16] T. Mei, L.-X. Tang, J. Tang, and X.-S. Hua. Near-lossless semantic video summarization and its applications to video analysis. *ACM TOMCCAP*, 2013.

[17] S. Nepal, U. Srinivasan, and G. Reynolds. Automatic detection of goal segments in basketball videos. In *ACM MM*, 2001.

[18] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang. Video summarization and scene detection by graph modeling. *IEEE Trans. on CSVT*, 15(2):296–3.5, 2005.

[19] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013.

[20] Y. Pan, T. Yao, T. Mei, H. Li, C.-W. Ngo, and Y. Rui. Click-through-based cross-view learning for image search. In *SIGIR*, 2014.

[21] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014.

[22] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for tv baseball programs. In *ACM MM*, 2000.

[23] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[24] M. Sun, A. Farhadi, and S. Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014.

[25] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[26] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[27] J. Wang, Y. Song, and etc. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.

[28] T. Yao, T. Mei, and C.-W. Ngo. Learning query and image similarities with ranking canonical correlation analysis. In *ICCV*, 2015.

[29] T. Yao, T. Mei, C.-W. Ngo, and S. Li. Annotation for free: Video tagging by mining user search behavior. In *ACM MM*, 2013.