Weakly Supervised Summarization of Web Videos (Supplementary Material)

Rameswar Panda[1] Abir Das[2] Ziyan Wu[3] Jan Ernst[3] Amit K. Roy-Chowdhury[1]
[1] UC Riverside [2] Boston University [3] Siemens Corporate Technology
{rpand002@,amitrc@ece.}ucr.edu dasabir@bu.edu {ziyan.wu,jan.ernst}@siemens.com

Table 1. TABLE OF CONTENTS

Detailed Information on the Experimented Datasets

Table 2. **Descriptive Statistics of CoSum Dataset.**

Video Topics	# Videos	Length	# Frames	# Shots
Base Jumping	5	10m54s	17,960	201
Bike Polo	5	14m08s	22,490	264
Eiffel Tower	7	25m47s	43,729	435
Excavators River Xing	3	10m41s	16,019	162
Kids Playing in Leaves	6	15m40s	27,972	263
MLB	6	12m11s	21,271	205
NFL	3	13m28s	23,179	190
Notre Dame Cathedral	5	11m26s	20,110	217
Statue of Liberty	5	10m44s	18,542	184
Surfing	6	22m40s	34,790	373
Total	**51**	**2h27m40s**	**246,062**	**2494**

(Please see Fig. 1 for category-wise image samples from CoSum dataset.)

Table 3. **Descriptive Statistics of TVSum Dataset.**

Video Topics	# Videos	Length	# Frames	# Shots
Changing Vehicle Tire	5	25m25s	39,841	911
Getting Vehicle Unstuck	5	19m28s	35,014	841
Grooming an Animal	5	18m07s	30,920	767
Making Sandwich	5	24m58s	37,095	770
ParKour	5	24m50s	41,634	993
PaRade	5	25m03s	44,042	715
Flash Mob Gathering	5	18m37s	30,747	618
Bee Keeping	5	17m30s	30,489	678
Attempting Bike Tricks	5	14m39s	25,747	523
Dog Show	5	20m59s	36,827	754
Total	**50**	**3h29m42s**	**352,356**	**7570**

(Please see Fig. 2 for category-wise image samples from TVSum dataset.)

Figure 1. **CoSum dataset** contains 51 videos downloaded from YouTube using 10 topic keywords from the SumMe dataset with a duration filter of 4 minutes and an additional constraint such that each video set contains at least 10 minutes of videos.

Figure 2. **TVSum dataset** contains 50 videos collected from YouTube using 10 topic keywords from the TRECVid Multimedia Event Detection task with the following criteria: (a) under the Creative Commons license; (b) duration is 2 to 10 minutes; and (c) title is descriptive of the visual topic in the video.

Additional Results on Video Skimming

(a) Qualitative Results on CoSum Dataset

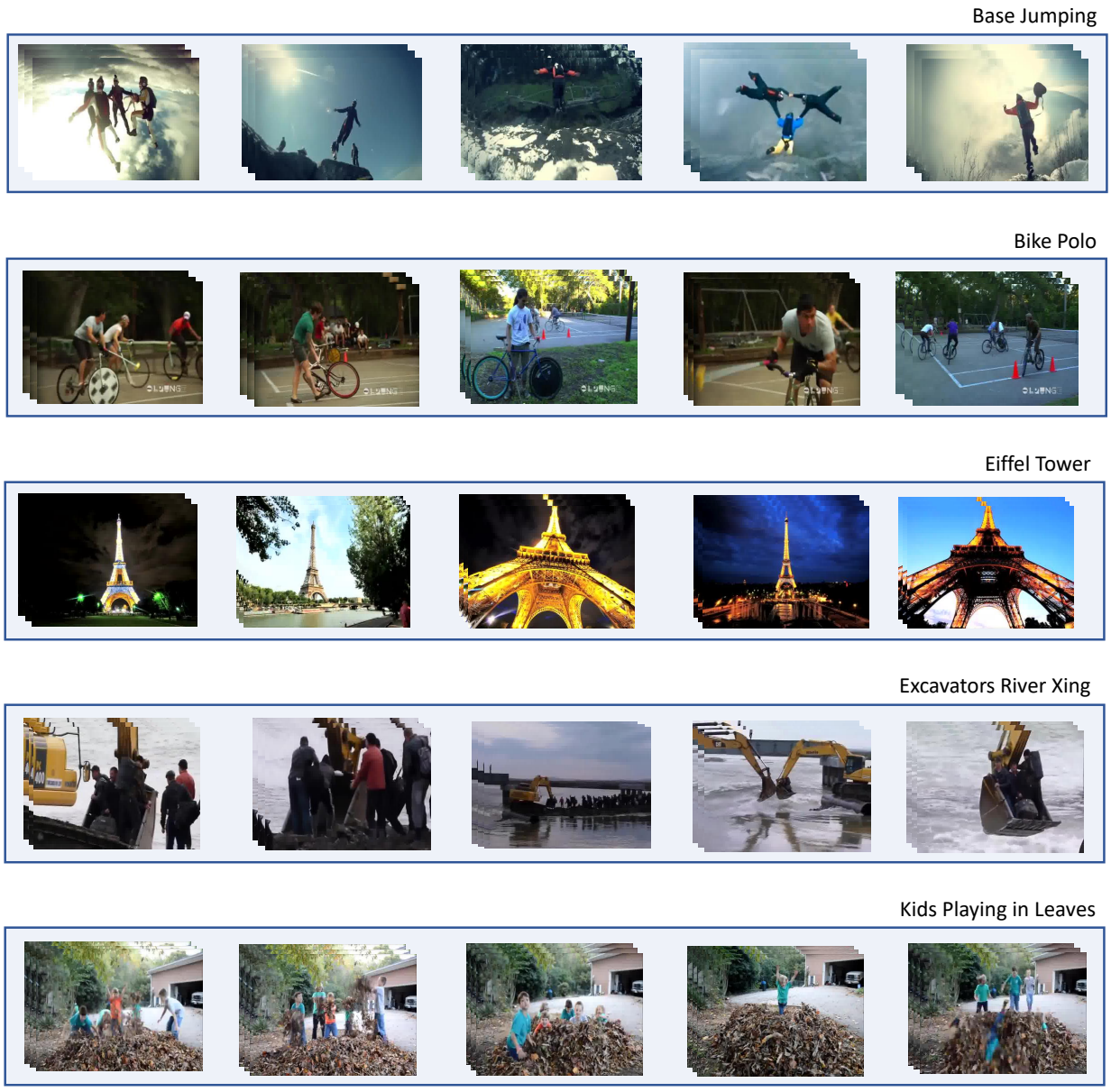

Figure 3. Illustration of summaries constructed with our method. We show the top-5 results represented by three central frames from each shot. As can be seen, our approach, `DeSumNet` selects a diverse set of informative shots that better visualizes different important aspects within a category. Our method produces summaries comparable to human created summaries. Best viewed in color.

(a) Qualitative Results on CoSum Dataset (Contd..)

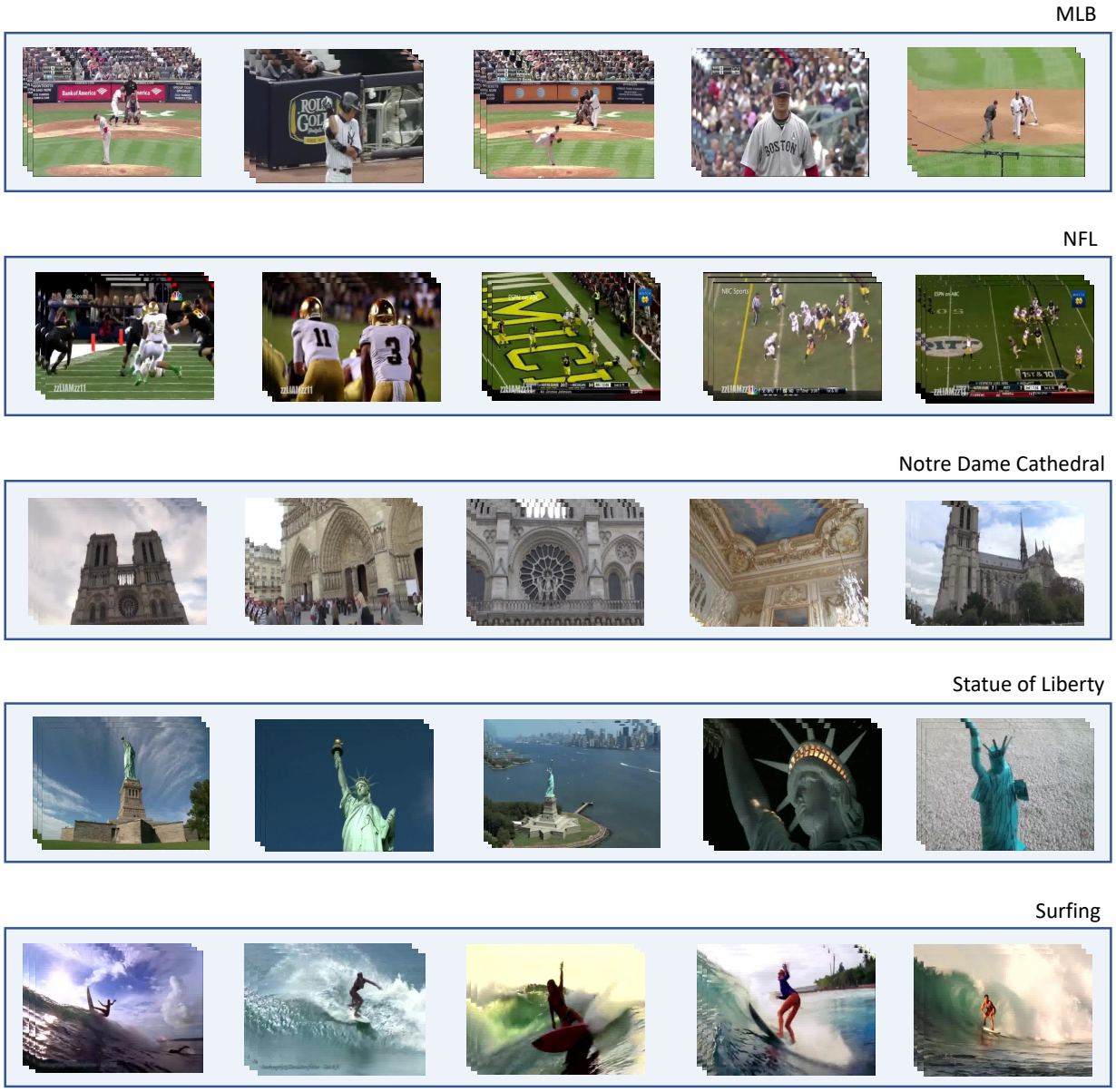

Figure 4. Illustration of summaries constructed with our method. We show the top-5 results represented by three central frames from each shot. As can be seen from the figures, our approach performs well in most of the cases. However, in some cases, e.g., while summarizing a video of the category Notre Dame Cathedral (3rd row), our approach focuses on selecting shots that shows only the picture of the church from different view angles. We believe this is due that fact that all the shots in this video look visually similar with little motion content and are temporally crowdely clustered. Best viewed in color.

(b) Qualitative Results on TVSum Dataset

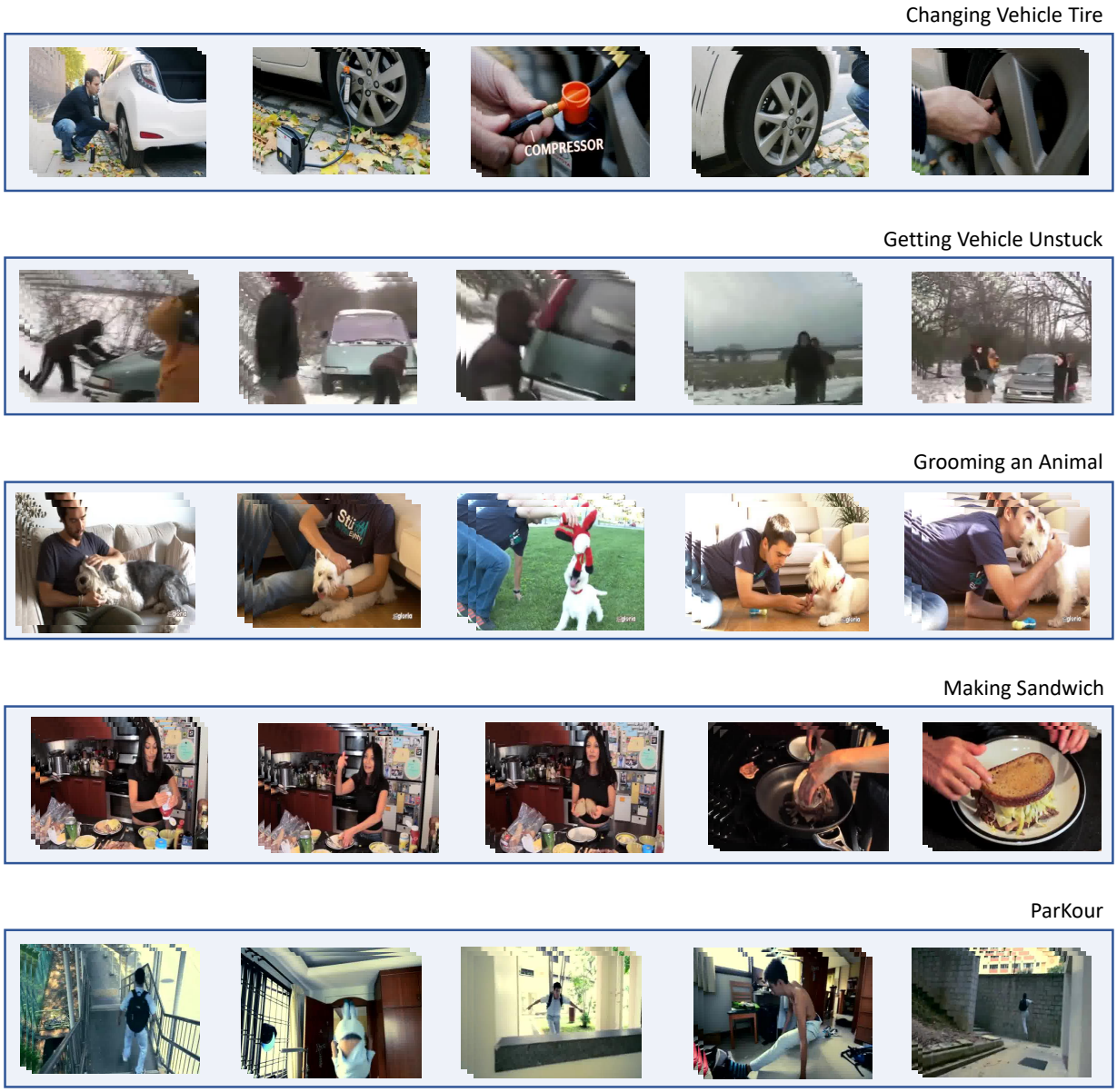

Figure 5. Illustration of summaries constructed with our method. We show the top-5 results represented by three central frames from each shot. Note that performance of our method (including all the compared methods), is somewhat low compared to the performance in CoSum dataset. We believe this is because summarization in this dataset is more challenging because of the unconstrained topic keywords. We also observe that the unsupervised methods (SMRS, LL, MBF, CVS) more often produces redundant summaries since they are blind to the video category, whereas our approach in general selects diverse contents in summarzing such videos. Best viewed in color.

(b) Qualitative Results on TVSum Dataset (Contd..)

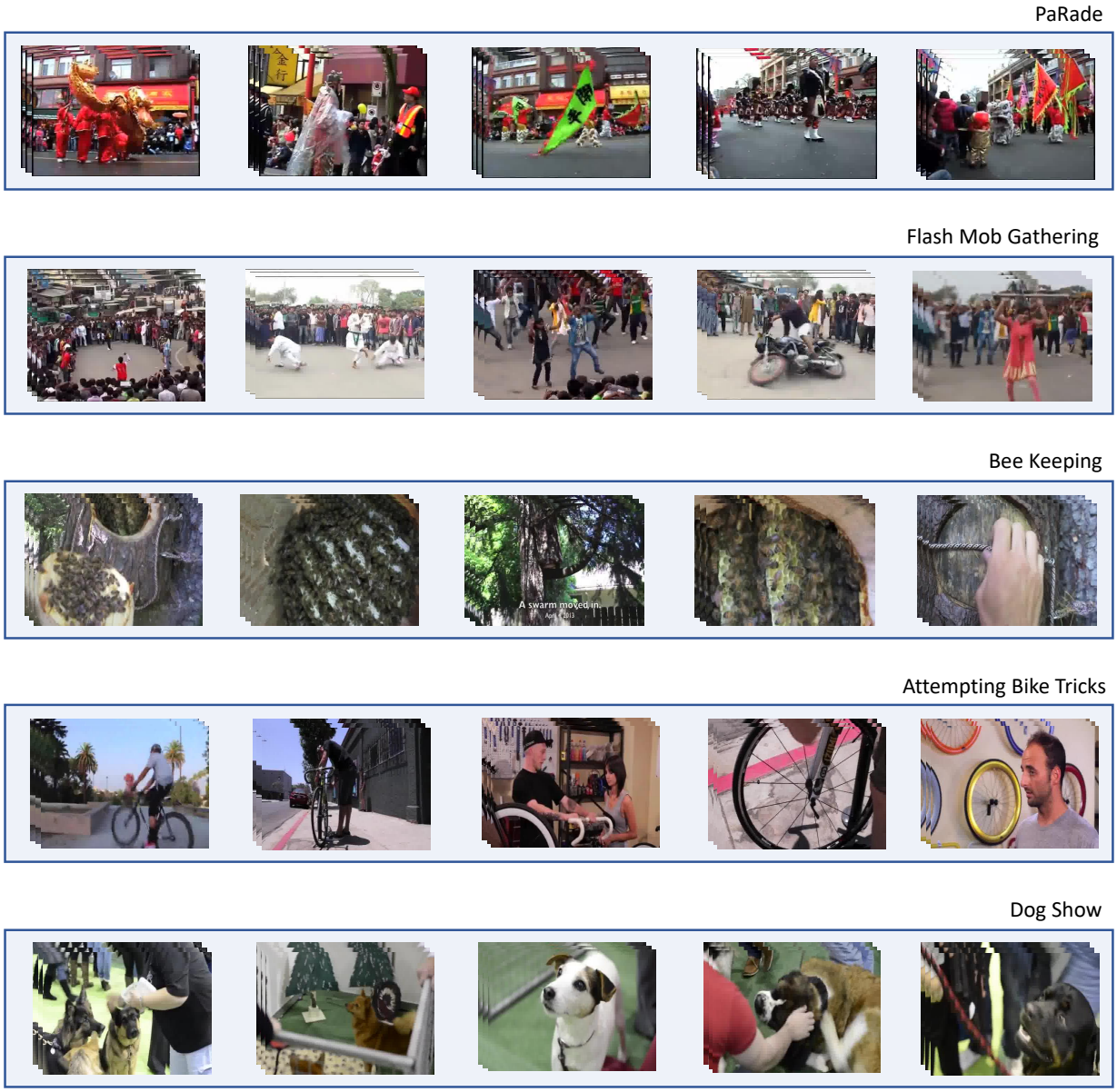

Figure 6. Illustration of summaries constructed with our method. We show the top-5 results represented by three central frames from each shot. As can be seen from the 4th row of the figure, our approach fails to select several important shots related to the bike tricks. This shows another failure case of our approach, where performance of our approach is slightly lower compared to the KVS baseline (Top-5 mAP: 0.396 vs 0.415). We believe this is because this video contains subtle semantics like riding the bike with different styles or performing different stunts which are difficult to capture without an additional semantic analysis. Best viewed in color.

(c) Summarizing Long User-generated Videos

Wearable devices have become pervasive. People are capturing high-quality videos using GoPro cameras and Google Glass everyday. Summarizing these first-person videos are more challenging compared to tradional web videos since these videos are usually extremely unstructured and long-running. To verify the performance of our approach in summarizing long videos, we collected 15 top-ranked videos from YouTube with category name Eiffel Tower and Notre Dame Cathedral as keyword (average duration$\sim$12 mins). Both categories are present in CoSum dataset and so it allows us to directly adopt our learned model without retraining. In absence of ground truth summaries for these videos, we follow [13] and perform an objective evaluation using shot reconstruction degree which measures how well a video can be reconstructed using the extracted summary. We compare with the recent collaborative summarization method [9] and observe that our approach still outperforms [9] by a margin of 6.1% in summarizing such videos. For even longer (say$\sim$1 hr) videos with multiple diverse activities, our approach can also flexibly leverage multi-label training (Sigmoid cross entropy loss instead of Softmax) with a class saliency aggregation scheme [11] and the divide-and-conquer strategy used in [16] for summarization. Another possible strategy would be to divide these long videos into several short segments by using an event detection pipeline [14] and then summarizing such events with our approach to create video summaries.

(d) Experiment using Newly Mined Web Videos

We implemented two baselines on top of the collaborative summarization method [9] using newly mined raw videos and refined videos as additional topic-related videos and observe that our approach still outperforms both variants of [9] by a margin of 4.6% and 3.4% respectively on CoSum dataset. We use same refined videos generated in our model adaptation scheme for a fair comparison with the baselines.

(e) Detailed Description on Ground truth Summaries

As described in Section 4.1 (Evaluation) of the main paper, we use multiple human-created groundtruth summaries to compute the average precision in both of the experimented datasets. This approach has the advantage that, once the ground truth summaries are obtained, experiments can be carried out indefinitely, which is desirable especially for multimedia systems that involve multiple iterations and testing. Both of the datasets provide multiple user-annotated summaries per video for comparison with the system generated summary. Below, we provide a more detailed description on the ground truth summaries for both of the datasets (CoSum and TVSum datasets).

[CoSum Dataset] The original CoSum dataset contains three human created summaries. Since video summarization is a subjective task and different users have different perception about what is important in a video, it is desirable to compare the system generated summaries with more number of human created summaries for a fair comparison. Moreover, we need importance annotations for each shot on the CoSum dataset. These annotations are essential to compare the machine generated results with the human performance. Specifically, for a particular video, given a human generated summary consisting of several shots as a ranked list, we want to evaluate the average precision of a machine generated summary $S_1, \cdots, S_n$, where S_1 is the highest scoring shot. We compute both top-5 and top-15 average precision values by considering top 5 and top 15 high scoring shots from the ranked list of $S_1, \cdots, S_n$ as a summary respectively. Towards this, authors in [9] have created two more ground truth summaries using a similar crowd sourcing experiment, as in [1] and also provide importance annotation for all the groundtruth summaries. We obtained the all the five ground truth summaries along with the importance annotations from [9] to compare with the system generated summaries in our experiments.

[TVSum Dataset] For TVSum dataset, we compare each summary with twenty ground truth summaries that are created via crowdsourcing. Since, ground truth annotations in this dataset contain frame wise importance scores, so, we first compute the shot-level importance scores by taking average of the frame importance scores within each shot and then select top 50% shots as the ground truth summary for each video. Finally, the ranked list of shots were constructed by arranging the shots in a descending order based on the average ratings.

(f) Implementation Details on the Compared Methods

As described in Section 4.1 (Compared Methods) of the main paper, we compare our approach with several methods that fall into three main categories: (1) unsupervised approaches including SMRS [4] (CVPR'12), Quasi [4] (CVPR'14), MBF [1] (CVPR'15), and CVS [9] (CVPR'17); (2) supervised methods including KVS [10] (ECCV'14), seqDPP [5] (NIPS'14), and SubMod [6] (CVPR'15); (3) human performance comparison including Worst Human, Mean Human, and Best Human. Here, we first present the common experimental settings that are applicable to all the methods and then describe the implementation details of each compared methods.

[Common Settings] We follow the following experimental settings in all our compared methods.

• Video Segmentation: For all the videos, we first segment them into multiple shots using an existing algorithm [1]. More specifically, we divide each video into multiple non-uniform shots by measuring the amount of changes between two consecutive frames in the RGB and HSV color spaces. A shot boundary is determined at a certain frame when the portion of total change is greater than 75%. We added an additional constraint to the segmentation algorithm to ensure that the number of frames within each shot lies in the range of [32,96]. The segmented shots serve as the basic units for feature extraction and subsequent processing to extract a video skims in all of the compared methods (including the proposed method).

• Features: We extract C3D features, by taking sets of 16 frames, applying 3D convolutional filters, and extracting the responses at layer FC6 as suggested in [12]. This is followed by a temporal mean pooling scheme to maintain the local ordering structure within a shot. Then the pooling result serves as the final feature vector of a shot (4096 dimensional) to be used in all of the compared methods. Note that our proposed architecture also uses 3D CNNs for computing the spatio-temporal importance scores and hence the use of c3d features gives a fair comparison with the compared methods.

• Evaluation: As described in Section 4.1 (Evaluation) of the main paper, we assess the quality of an automatically generated video skim by comparing it to human judgment. In particular, we compute the mean pairwise average precision (AP) between a system generated summary multiple ground truth summaries. We follow [16, 9] and extend VSUMM evaluation package [3][1] for finding matching pair of video shots. Given two sets of summaries, one formed by an algorithm and the other by human annotators, we first find the maximum number of matched pairs of shots between them. Two shots are qualified to be a matched pair if their visual difference is below a certain threshold, while each shot of one summary can be matched to at most one shot of the other summary. We then can measure average precision based on these matched pairs.

[Unsupervised Methods]

• SMRS: Sparse Modeling Representative Selection (SMRS) finds the representative shots using the entire video as the dictionary and selecting key shots based on the zero patterns of the coding vector. Mathematically, SMRS solves the following optimization problem to estimate a selection matrix $\mathbf{Z} \in \mathbb{R}^{n \times n}$:

$$\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 + \lambda \|\mathbf{Z}\|_{2,1} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{d \times n}$ is the video feature matrix whose columns represent shots. $||\mathbf{Z}||_{2,1} = \sum_{i=1}^{n} ||\mathbf{Z}_{i,.}||_2$ denotes the $\ell_{2,1}$-norm and $||\mathbf{Z}_{i,.}||_2$ is the ℓ_2-norm of the i-th row of $\mathbf{Z}$. $\lambda > 0$ is a regularization parameter that controls the level of sparsity in the reconstruction. We solve (1) using Alternating Direction of Method of Multipliers (ADMM) and select the representative shots for generating video skims as the points whose corresponding $||\mathbf{Z}_{i,.}||_2 \neq 0$. Note that [2] also uses the same objective function as in [4] for summarizing consumer videos. The only difference lies in the algorithm used to solve the objective function (Proximal vs ADMM). Hence, we compared only with [4].

• Quasi: Quasi real-time video summarization (Quasi) method uses an online variant of sparse coding to generate a video skim over time by measuring the redundancy using a dictionary of shots updated online. We implemented it using SPAMS library [8] with dictionary of size 200 and the threshold $\epsilon_0 = 0.15$, as described in [18].

• MBF: We compare with an unsupervised baseline method that leverage visual co-occurrence across the category-related videos to generate a summary. MBF finds maximal bicliques from the complete bipartite graph using a block coordinate

[1] https://sites.google.com/site/vsummsite/download

descent algorithm. We generate a summary by selecting top-ranked shots based on the visual co-occurrence score and set the threshold to select maximal bicliques to 0.3, following [1].

• `CVS`: Collaborative Video Summarization (`CVS`) exploits visual context from a set of category-related videos to extract an informative summary of a given video. Specifically, a collaborative sparse representative selection strategy is developed to simultaneously captures both important particularities arising in the given video, as well as, generalities identified from the additional category-related videos. We obtained the code from the authors for `CVS` and set the parameters same as in [9].

[Supervised Methods]

• `KVS`: Kernel Video Summarization (`KVS`) uses multiple linear SVM classifiers, one per each category to score importance of small video segments. 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.

• `seqDPP`: `seqDPP` uses a sequential version of determinantal point process for selecting diverse video shots. The basic idea of this approach is to learn from human-created summaries on how to select informative and diverse video subsets so as to best meet valuation metrics derived from human perceived quality. Since this method requires one human-created summary for each video in training, we follow a greedy algorithm as described in [5, 16] to generate training groundtruths (i.e. oracle summaries) from multiple human-created summaries in both datasets.

• `SubMod`: Given pairs of videos and their human-created summaries as training examples, `SubMod` learns a combined objective using submodular functions. Then, when given a new video as input, it creates summaries that are both interesting and representative. We use three submodular functions with respect to interestingness, representativeness and uniformity as the desired properties of a video summary and optimize combined objective using a projected subgradient method [7][2]. Note that there is no requirement of creating oracle summaries like the previous method `seqDPP` as `SubMod` considers multiple human-created summaries of a video while optimizing the submodular functions.

[Human Performance Comparison]

Given a video with multiple ground truth summaries, we compute the average precision (AP) between the human-created summaries to compare the machine generated results with the human performance. Specifically, we first compute the pairwise average precision between the human created summaries for each video. `Worst human` score for a video is computed using the summary which is the least similar to the rest of the human created summaries. Similarly, `best human` score for a video represents the summary that contain most shots that were selected by many humans. We also report the `mean human` score of a video which provides a pseudo upper bound for the summarization task. Note that all the human performance scores are computed using the publicly available ground truth summaries on both datasets and hence further annotation or creation of ground truths are required in our evaluation strategy to compute those scores.

(g) Experimental Comparison with [17]

We have compared with total 7 baseline methods (4 unsupervised and 3 supervised) in Sec.4.1 of the main paper. Here, we additionally compare with [17] using the publicly available code and observe that performs best in TVSum dataset and comes 2nd in CoSum dataset among supervised methods. Top-5 mAP difference between ours and [17] in CoSum is still moderate ($\sim 1\%$), suggesting that although being weakly supervised, our method is still competitive with the fully supervised method [17] in summarizing videos. On TVSum, the performance gap between ours and [17] begins to appear ($\sim$8%). This is expected as on a challenging dataset with very diverse videos, a weakly supervised approach cannot compete with a fully supervised one, especially when the later one is using huge amount of human-annotated video-summary pairs.

[2]We use the publicly available code at http://submodularity.org/#materials

Additional Discussions

(a) Detailed Information on User Study: Generating Video Time-lapse

As described in Section 4.2 of the main paper, since there exists no publicly available groundtruths to evaluate the quality of video time-lapse, we performed subjective evaluation using ten human experts. Here, we describe the entire task setup that we used to evaluate the performance of different methods in generating video time-lapse.

Task setup:

(a) Before the study begins, we first show the topic keyword (e.g., Surfing) along with the original video to a human and then asked to emulate various concepts related to the topic on their mind. Our objective is to understand how a user perceive the quality of a time-lapse video according to the topic-specific concepts.

(b) Study experts were asked to rate the overall quality of each time-lapse video by assigning a rating from 1 to 5, where 1 corresponded to "*The generated time-lapse video is not at all informative in visualizing different concepts from the original video*" and 5 corresponded to "*The time-lapse video is informative, visually appealing and very well describes the concepts related to the topic*". For each video, the human rating was computed as the averaged ratings from all study experts.

(d) To perform a fair comparison, we provided all the time-lapse videos at a time, where each time-lapse was generated from one method, i.e., `CVS`, `KVS`, and `DeSumNet` (ours) (as described in Sec 4.2: Compared methods, of the main paper). The time-lapse videos were shown in random order without revealing the identity of each method and the audio track was not included to ensure that the subjects chose the rating based solely on visual stimuli.

(e) Moreover, we did not maintain the same order of the time-lapses across different videos of the dataset. This is to ensure that the users will not be biased from the previous order and ratings while providing ratings for the current video of interest.

(f) Evaluation for both datasets took roughly 40 minutes for an user that amounts to total 3.5 hours of user time to complete the entire study.

(b) Computation Cost

Our approach is reasonably fast as spatio-temporal importance computation only requires a single backward pass. E.g, in a single Tesla K80 GPU, it takes on average$\sim$1 min to summarize a video on CoSum dataset while [9] takes 3 mins on same computing platform. Our approach is also flexible to use multi-GPU inference or sequential modeling trick used in [5] to further reduce cost.

(c) Analyzing Failure Cases and their Possible Solutions

Analyzing failure cases is an important aspect in any computer vision and machine learning approach. Since our approach is based on video categorization, the performance of summarization may drop if the categorization step fails to correctly classify a video. In this case one may extend our work to video tagging in the first module. One can also use multi-label classification pipeline [11] instead of the current video categorization approach to generate video summaries. Moreover, without changing the strategy of our approach, the two main modules of video categorization and importance computation may be replaced by improved methods such as neural excitation back-propagation [15].

References

[1] W.-S. Chu, Y. Song, and A. Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *CVPR*, 2015. 9, 10, 11

[2] Y. Cong, J. Yuan, and J. Luo. Towards Scalable Summarization of Consumer Videos Via Sparse Dictionary Selection. *TMM*, 2012. 10

[3] S. E. F. de Avila, A. P. B. Lopes, A. da Luz Jr., and A. de Albuquerque Arajo. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *PRL*, 2011. 10

[4] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *CVPR*, 2012. 10

[5] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*, 2014. 10, 11, 12

[6] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *CVPR*, 2015. 10

[7] H. Lin and J. A. Bilmes. Learning mixtures of submodular shells with application to document summarization. *arXiv preprint arXiv:1210.4871*, 2012. 11

[8] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19–60, 2010. 10

[9] R. Panda and A. K. Roy-Chowdhury. Collaborative summarization of topic-related videos. In *CVPR*, 2017. 9, 10, 11, 12

[10] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014. 10

[11] W. Shimoda and K. Yanai. Distinct class-specific saliency maps for weakly supervised semantic segmentation. In *ECCV*, 2016. 9, 12

[12] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3d: generic features for video analysis. *CoRR, abs/1412.0767*, 2:7, 2014. 10

[13] B. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *TOMCCAP*, 2007. 9

[14] Y. Yan, Y. Yang, D. Meng, G. Liu, W. Tong, A. G. Hauptmann, and N. Sebe. Event oriented dictionary learning for complex event detection. *TIP*, 2015. 9

[15] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *ECCV*, 2016. 12

[16] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *CVPR*, 2016. 9, 10, 11

[17] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *ECCV*, 2016. 1, 11

[18] B. Zhao and E. P. Xing. Quasi real-time summarization for consumer videos. In *CVPR*, 2014. 10