Supplementary Material for "First-Person Video Summarization From Third Person's Point of Views"

Hsuan-I Ho[1], Wei-Chen Chiu[2], and Yu-Chiang Frank Wang[1]

[1] Department of Electrical Engineering, National Taiwan University, Taiwan
{b01901029, ycwang}@ntu.edu.tw
[2] Department of Computer Science, National Chiao Tung University, Taiwan
walon@cs.nctu.edu.tw

1 Dataset

First-Person Video Data We provide the details of our proposed first-person video dataset, FPVSum. This dataset is collected from YouTube by following the procedure of [7]. That is, we select 10 video categories from [1, 5, 7] plus 4 new ones (as listed in Table A). When collecting this video dataset, we focus on continuous first-person videos only (i.e., no transition within or between points of views); moreover, videos with unrelated contents will be excluded. Therefore, a total number of 98 first-person videos are obtained. Table A lists the videos of 14 categories. As discussed later, we will explain how the annotation is provided for selected videos for training, test, and evaluation purposes.

Table A: Descriptions and properties of our proposed FPVSum dataset. Note that (a) denotes the total length, (b) lists the numbers of highligh/non-highlight segments, and (c) shows the number of annotated/total number of frames.

Category	**(a)**	**(b)**	**(c)**	Cronb. α	f-measure
Biking	38m 22s	51 / 290	20595 / 67669	0.879	0.414
Bikepolo	32m 31s	40 / 323	23729 / 54270	0.733	0.252
Boxing	45m 39s	72 / 347	25312 / 77237	0.754	0.294
HorseRiding	54m 39s	48 / 307	21491 / 98369	0.954	0.609
Jumping	22m 25s	43 / 208	15230 / 39279	0.875	0.422
LongBoarding	28m 32s	58 / 300	21636 / 49335	0.771	0.297
Motor	24m 24s	41 / 232	16545 / 39337	0.907	0.466
Parkour	21m 49s	41 / 232	16561 / 35411	0.838	0.337
Plane	29m 50s	61 / 271	20069 / 53787	0.753	0.279
RockClimbing	49m 17s	80 / 377	27565 / 88709	0.495	0.244
Scuba	44m 28s	98 / 412	30773 / 80089	0.618	0.225
Skate	23m 46s	15 / 153	10263 / 40733	0.890	0.457
Ski	38m 3s	66 / 269	20250 / 63522	0.870	0.431
Surfing	22m 16s	48 / 158	12581 / 40098	0.903	0.524
Total	476m 1s	762 / 3879	282600 / 827845	0.783	0.362

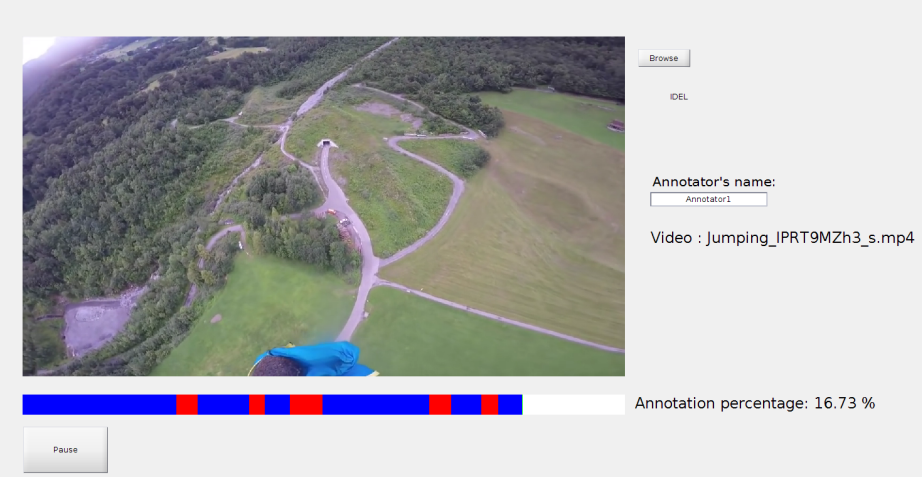

Fig. A: Our human interface for highlight annotations. For a given input video, the blue and red color bars denote non-highlight and highlight segments selected by a user, respectively.

Annotations We follow [1] to perform video annotation. That is, given each video, annotators are asked to produce a summary that contains most of its important content and highlight segments using our designed human interface shown in Fig. A. The interface shows each video excluding its audio track, ensuring annotators select highlight based on visual content only. Annotators are able to use the interface for moving forward and backward and modify their annotations at any time. The details of our annotation process are shown as follows:

- The annotators require to select highlight/non-highlight segments in each video. They need to finish watching each video once, then they start the labeling process.
- The annotators are asked to select the video parts which they consider interesting or important (i.e., mark the parts to red color using the interface in Fig. A). We note that an interesting part being marked may vary in any length.
- The annotators are encouraged to produce the summary which accounts for 10% to 20% of the full video length.
- Each frame would get an importance score which indicates how many annotators mark on this frame. We finally select frames ranked in the top 15% of all video frames as the highlight ones.

The consistency of human annotation for our FPVSum dataset can be evaluated by two metrics, Cronbach α and pairwise f-measure, which are both utilized

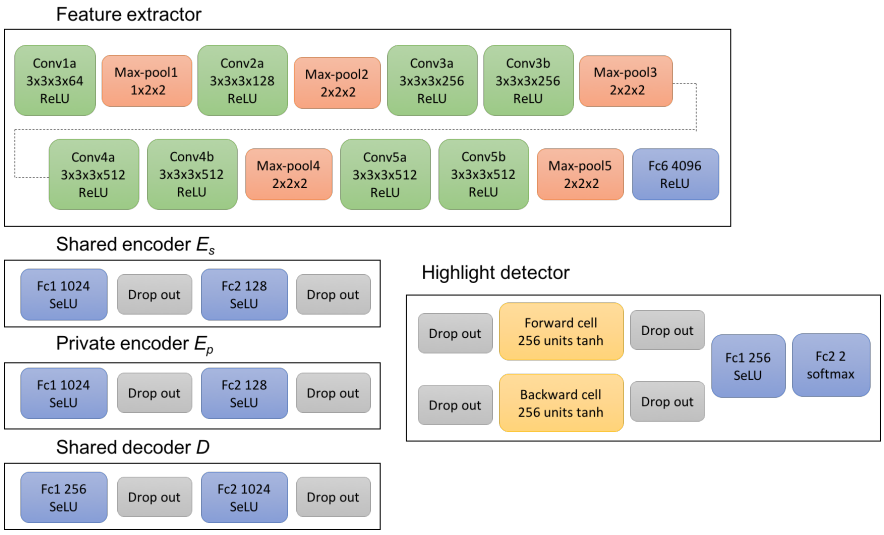

Fig. B: Our network topology for the components of feature extractor, feature embedding, and highlight detector.

to evaluate that of SumMe in [1]. Table A shows both human consistency metrics in each category.

As noted in our manuscript, videos with long durations typically result in inconsistent annotation scores from the users, since he/she tends to lose concentration when viewing/assigning highlight or non-highlight labels. Thus, the collected video sequences have durations of about 1 to 6 minutes for each particular category. Finally, about 65% of the videos are regarded as unlabeled data for learning. As for the remaining ones with ground truth scores (which are annotated by 10 different users), about 80% will be randomly chosen for training and the rest for testing.

2 Implementation Details

Inputs In our first-person video summarization framework, we take video segments with fixed-length as the basic elements for capturing spatiotemporal information in the video. In particular, a video is split into a series of 2-second segments, and each segment is composed of 16 video frames (i.e., videos are downsampled to 8 fps). We further categorize all video segments into highlight and non-highlight subsets according to their importance scores (segments of the top 15% importance scores are highlight ones, while the rest are non-highlight ones). The total number of highlight/non-highlight segments in FPVSum is shown in Table. A. Together with the videos from other datasets (i.e, SumMe and TvSum

as listed in Table 2 of the main submission), we generate extensive training sets within and across first- and third-person highlight/non-highlight subsets.

Network Structures We first train a feature extractor for capturing spatiotemporal information in each video segment. We adopt the architecture of 3-Dimensional Convolutional Networks (i.e., C3D [6]) as our feature extractor, in which its weights are initialized by the C3D learned from Sport1M [3] video classification dataset while further fine-tuned in our training procedure of video summarization. The feature (4096-d) yielded from the fc-6 layer of extractor serves as the input of the cross-domain feature embedding network.

Our cross-domain feature embedding network consists of two private encoders, a shared encoder, and a shared decoder. Each encoder is a two-layer fully connected network (1024, 128 SeLU units), and the shared decoder has a two-layer fully connected structure (256, 1024 SeLU units). The sequential highlight detection network consists of a biLSTM with 256 hidden units in the both forward and backward cells followed by a 256-units fully connected layer and a softmax output layer. We present detailed network topology in Fig. B.

Parameter Settings To train the proposed model, we perform a two-stage optimization process as we mentioned in the main paper. That is, we first train the feature embedding network with segment-based highlight detector, where each segment is treated independently, then perform joint training of sequential highlight detector by using sets of consecutive video segments as input.

To be detailed, in the first stage we train our feature embedding network based on the 400K training sets generated from both first- and third-person highlight/non-highlight subsets. The margin parameter $\mathcal{M}$ of triplet loss is set as 1.2 and the size of shared and private features is 128. We train our network using Adam optimizer with a batch size of 8, first- and second-momentum of 0.9 and 0.99, and dropout probability of 0.8. We use the hyperparameters $\alpha = 0.5, \beta = 10^3, \gamma = 1.0$ to balance overall losses. The learning rate of the feature embedding network is set to 10^{-4} while the segment-based highlight detector is set to 10^{-5}. The network is trained in total 50K steps. We note that, since the unlabeled data needs pseudo labels as described in Section 3.2 of our main paper, they are used after 10K steps of training.

In the second stage, consecutive video segments are used for jointly learning the parameters of the sequential highlight detector. We optimize our network by Adam optimizer with a batch size of 4, first- and second-momentum of 0.9 and 0.99, dropout probability of 0.8. The learning rate of the sequential highlight detector is set as 10^{-5} whereas the feature embedding network is finetuned with a learning rate of 10^{-6}. The overall network is trained in total 3K steps.

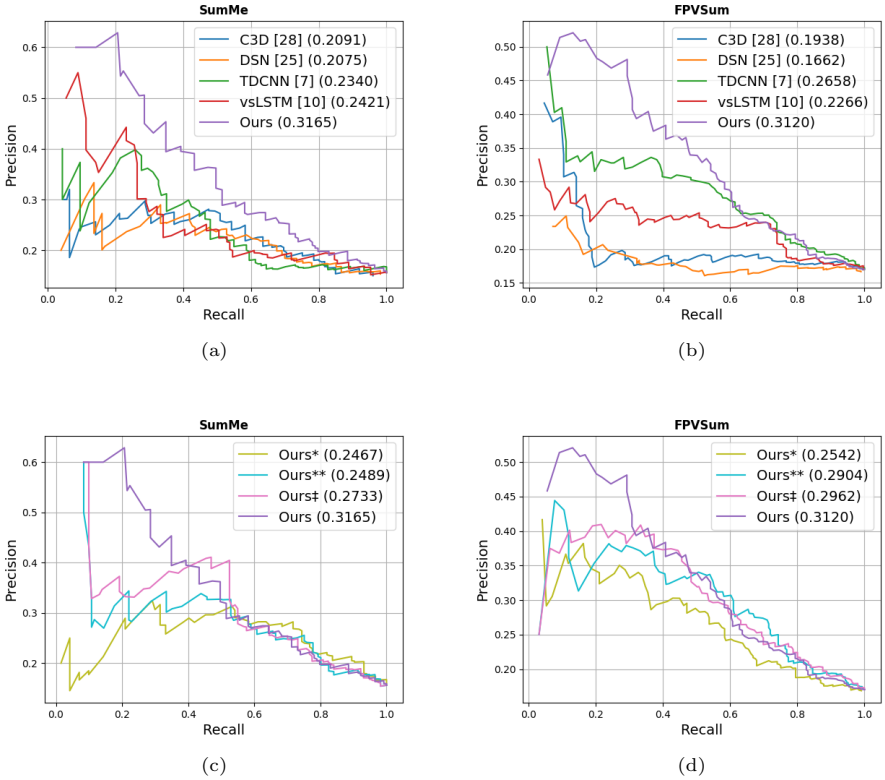

Fig. C: Precision-recall curves and AUC scores for SumMe and FPVSum. Note that (a) and (b) compare the P-R curves of several recent approaches, while (c) and (d) evaluate those of our variants (i.e., those listed in Table 4). The number followed by each method/model indicates the AUC score (e.g., 0.3165 for Ours on SumMe).

3 Additional Results

3.1 Precision-Recall Curves

In the main article, we follow the settings and evaluation metrics as those in [1, 2, 4, 5, 8], and compare f-measures of different methods. We additionally consider precision-recall curves and the corresponding area-under-curve (AUC) values for further evaluation.

Figures C(a) and (b) present the P-R curves and AUC scores of different methods on both SumMe and FPVSum datasets. It can be seen that our proposed model consistently performed against recent deep learning methods. On the other hand, Figures C(c) and (d) compare P-R curves and AUC scores of different variants of our model (i.e., those presented in Table 4). Note that **Ours‡** indicates the non-sequential version of our model which use only fully connected

layers instead of RNN as the final highlight classifier. With such ablation studies, we again verify the contributions of the introduced components, which support the full version of our model for cross-domain video summarization.

3.2 Visualization

In this section, we show additional visualization results of testing videos. As in the main paper, the user-annotated scores (ground truth) are shown in blue, while the predicted summaries from our works, vsLSTM, and TDCNN are shown in green, red and yellow, respectively. The red horizontal line split the scores into highlight (i.e., top 15%) and non-highlight ones.

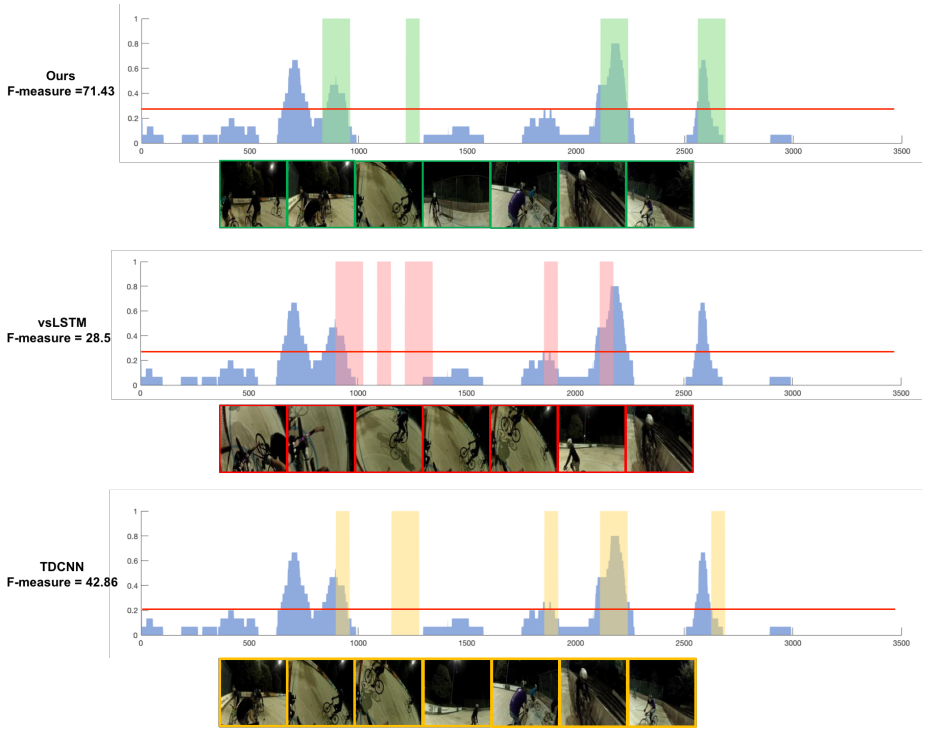

Fig. D: Performance comparisons on the video "Bike Polo" from SumMe. Note that the predicted highlight segments are denoted in green, red, and yellow for the methods of ours,vsLSTM, and TDCNN, respectively. We see that our summarization result is able to capture three precise highlight moments (e.g., shoot, goal, etc.) whereas others contained only parts of them.

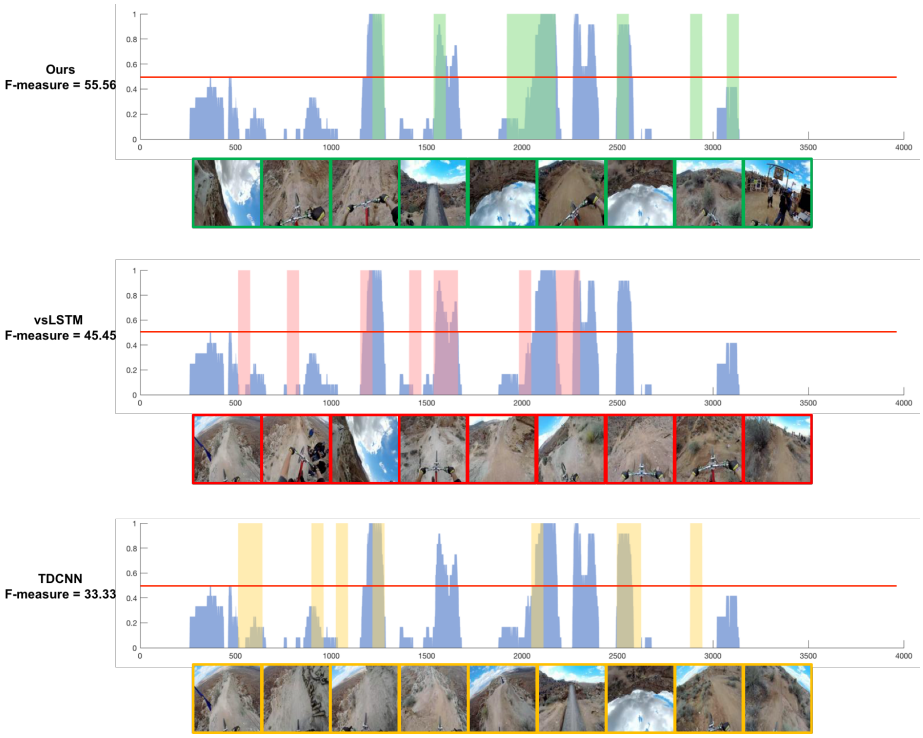

Fig. E: Performance comparisons on the first-person mountain biking video from FPV-Sum. We note that our summarization result includes unique moments in first-person biking videos such as “360° backflip” and “landing”.

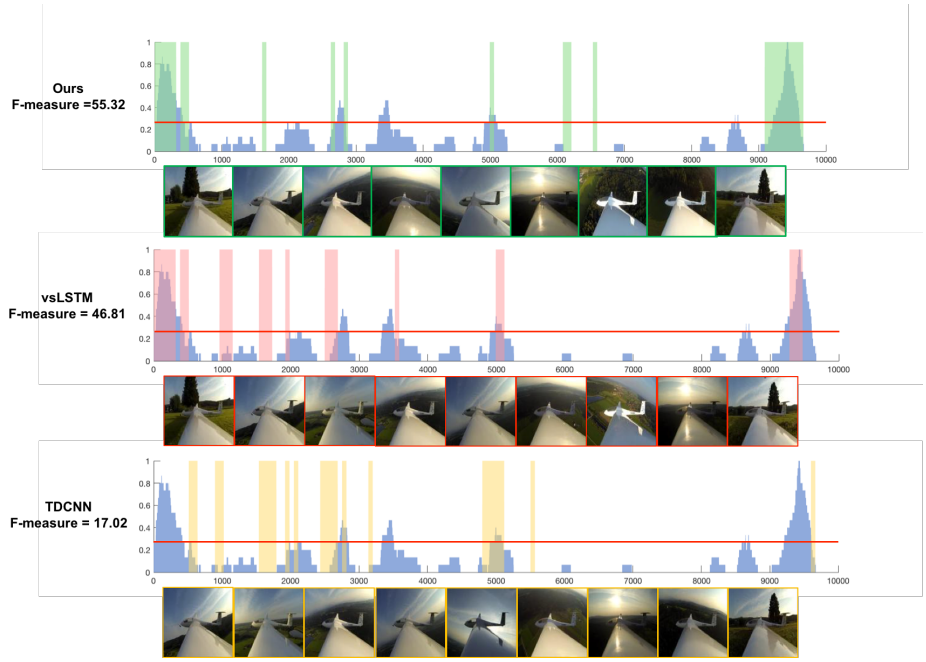

Fig. F: Example summarization results of video "Uncut Evening Flight" from SumMe. Note that our method captures moments like take-off, landing, or particular sunset scenes in the summarization output.

References

1. Gygli, M., Grabner, H., Riemenschneider, H., Van Gool, L.: Creating summaries from user videos. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
2. Ji, Z., Xiong, K., Pang, Y., Li, X.: Video summarization with attention-based encoder-decoder networks. arXiv:1708.09545 (2017)
3. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
4. Mahasseni, B., Lam, M., Todorovic, S.: Unsupervised video summarization with adversarial lstm networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
5. Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: Tvsum: Summarizing web videos using titles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
6. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
7. Yao, T., Mei, T., Rui, Y.: Highlight detection with pairwise deep ranking for first-person video summarization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
8. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)

Fig. G: Thumbnails of videos in FPVSum dataset, which consists of 98 first-person videos in 14 categories captured by wearable devices from YouTube.