Learning Trailer Moments in Full-Length Movies with Co-Contrastive Attention Supplementary Materials

Anonymous ECCV submission

Paper ID 2971

1 Introduction

We appreciate the reviewers' constructive feedback. To reduce potential confusion and missing information due to the page limitation in the main paper, we provide the supplementary materials for better understanding our works, which includes:

- the details of collecting Trailer Moment Detection Dataset (TMDD);
- the details of annotating the data and creating the Pesudo Labels;
- the architectures of our proposed CCANet and more explanation about loss $\mathcal{L}_C$ Eq.(6) and θ Eq.(7);
- qualitative results for shots ranking;
- codes for reproducing the results are on the anonymous Github page: https://github.com/BlindRev/video_moment_det.git.

Note: a equation, Eq. (X), we mention in this document are refereed to the one in the main paper with the index X.

2 Trailer Moment Detection Dataset

We collected 150 full-length American movies, including genres of Action, Drama and Sci-Fi. We manually search the trailer on Youtube[1] according to the movie titles. All the trailers are verified manually to avoid mismatch to the original movies, where a trailer video is represented by a unique Youtube ID.

Removing the garbage trailer shots. In our experiments, trailers and movies are segmented into multiple shots [4]. We train a deep ranking model such that the garbage trailer shots are predicted with a low score and can be easily removed. The network is a C3D [5] plus the ranking loss defined as Eq. 1 in the main paper. We sample the shots of cast running, visual transition, etc., which usually show in the begin and end of the movies or trailers, as negative samples, and shots in the main body of movies as positive samples, where the positive to negative ratio is 2:1. Additionally, the garbage shots can be further removed by manual verifying in the annotation stage, which is described next.

[1] https://www.youtube.com/

Fig. 1: Given the trailer shot (top row), annotators are asked to select the movie shots as the ground-truth from the top-5 matching results. Option 'Not found' indicates that none of the top-5 movie shots match the trailer, and 'Discard' refers to the garbage trailer shots, including visual transition, a cast run, coming soon announcement, or shots with black images, etc.

Data access for reproducibility. We will release the TMDD dataset in two steps. For reproducing the results we achieved in the main paper, we will follow as the practice in YouTube8M [1] to release the basic S3D features of each shots in both movies and trailers, shot-level annotations, youtube-ID for trailer videos and pre-trained models. Second, to facilitate the research direction, we will publish the original full-length movies once we get the permission from (*hidden for blind review*).

3 Annotating Movie Data

We match a trailer with the full-length movie on the shot level according to the frames' visual similarity. We uniformly sample 5 or 10 frames from a trailer or movie shot, respectively, forming 50 image pairs, where shots in the movie are less than one seconds are discarded. The Resnet-18 [3] network pre-trained on ImageNet [2] is adopted to extract the visual feature for each frame, where a feature is the 512 dimensional vector before the last pooling layer. The similarity score of a pair is measured by the Cosine Similarity between their features. The

shot similarity is the median value among those 50 pairs. The shot similarities are used as the Pseudo Labels in the comparison experiments, described in the main paper's Section 4.1.

In the annotation stage, given a trailer shot, we show its top-5 matched movie shots to annotators according to the shot similarities and annotators need to verify the matched movie shot, as shown in Figure 1.

4 Network Architectures and Loss $\mathcal{L}_C$

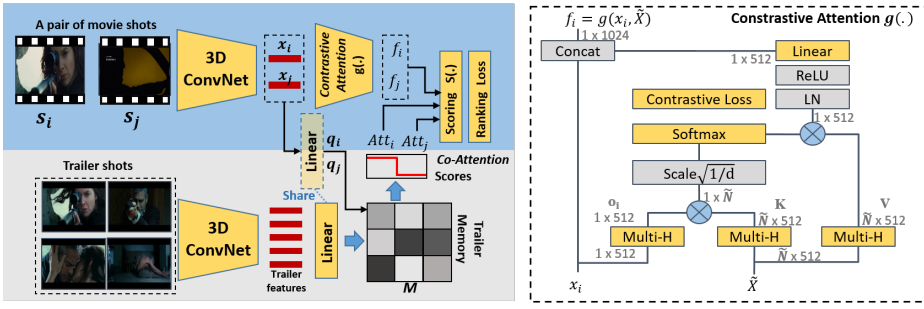

Fig. 2: The Network Architectures.

Network Structure. As shown in Figure 2, we model the scoring function $S(\cdot)$, and memory encoding with neural networks. We use a 3 fully connected layers for $S(\cdot)$ with hidden variables of 1024×512, 512×128 and 128×1. A fully connected layer with 512×512 hidden variables is used to map the 3D features into memory items for computing the Co-Attention scores between trailer and movie shots.

We follow the work [6] to use the "multi-head" attention to encode the 'o', 'K' and 'V' in the Contrastive Attention module $g(\cdot)$, where the head number is 8. The "linear" module in $g(\cdot)$ stands for a fully-connected layer with 512×512 hidden variables.

More explanation about loss $\mathcal{L}_C$ Eq. (6) and Eq. (7). Eq. (6) aims to learn the contrastive attention such that, to augment the feature from trailer shot s_i, other trailer shots *across* videos have larger attention weights than non-trailers surrounding s_i in the *same video*. As we use Co-Attention scores to soft label shots as trailer or non-trailer, we introduce a parameter θ defined by Eq. (7) to indicate the reliability of soft labels. We empirically set the hyper-parameters in Eq. (7), mapping the Co-attention into 0 or 1. We make Eq. (7) differentiable which can be incorporated into end-to-end model training. In the simple case that θ only takes value 0 or 1, Eq.(6) is a typical Contrastive loss:

$$\mathcal{L}_C = -\sum_i \log \frac{\sum_{j \in S^+} \exp(o_i^T k_j)}{\sum_{j \in S^+} \exp(o_i^T k_j) + \sum_{j \in S^-} \exp(o_i^T k_j)}$$

To compute the contrastive loss as Eq. (6), we need to construct the auxiliary shot set $\tilde{S}$, containing the positive or negative samples. We apply the shots' confidence weights to create the pseudo labels, where the weights computation is defined by Eq. (7). Given a mini-batch, we sample the positive shots $\tilde{s}_i^+ \in \tilde{S}$ which have a confidence weight larger than 0.8, and sample the negative shots $\tilde{s}_j^-$, where the confidence weight is less than 0.2. We force the negative samples only come from the same video based on the intuition that a trailer shot in a specific movie might not be more "trailerness" than shots in different movies. In our experiments, we observed that cross-video sampling the negative shots results in the performance drop.

5 Qualitative Results

We provide the qualitative results of movie shot ranking, where shots are shown as GIF images. Please check the slides, **"Qualitative_results.pptx"**.

6 Codes on Github

The codes for reproducing the results on Youtube Highlights Dataset are on the anonymous Github page for blind review, which achieves the largest marginal performance gain compared with the existing approaches:
https://github.com/BlindRev/video_moment_det.git

References

1. Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.
2. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
4. Alan F Smeaton, Paul Over, and Aiden R Doherty. Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, 114(4):411–418, 2010.
5. Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
6. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.