LiveSeg: Unsupervised Multimodal Temporal Segmentation of Long Livestream Videos

Jielin Qiu[1,2], Franck Dernoncourt[1], Trung Bui[1], Zhaowen Wang[1], Ding Zhao[2], Hailin Jin[1]
[1]Adobe Research, [2]Carnegie Mellon University
{jielinq,dingzhao}@andrew.cmu.edu, {dernonco,zhawang,bui,hljin}@adobe.com

Abstract

Livestream videos have become a significant part of online learning, where design, digital marketing, creative painting, and other skills are taught by experienced experts in the sessions, making them valuable materials. However, Livestream tutorial videos are usually hours long, recorded, and uploaded to the Internet directly after the live sessions, making it hard for other people to catch up quickly. An outline will be a beneficial solution, which requires the video to be temporally segmented according to topics. In this work, we introduced a large Livestream video dataset named MultiLive, and formulated the temporal segmentation of the long Livestream videos (TSLLV) task. We propose LiveSeg, an unsupervised ***Live****stream video temporal* ***Seg****mentation solution, which takes advantage of multimodal features from different domains. Our method achieved a* 16.8% *F1-score performance improvement compared with the state-of-the-art method.*

1. Introduction

Video temporal segmentation has become increasingly important since it is the basis for many real-world applications, i.e., video scene detection, shot boundary detection, etc. Video temporal segmentation can be considered an essential pre-processing step, and an accurate temporal segmentation result could benefit many other tasks. The video temporal segmentation methods lie in two directions: unimodal and multimodal approaches. Unimodal approaches only use the visual modality of the videos to learn scene change or transition in a supervised manner, while multimodal methods exploit available textual metadata and learn joint semantic representation in an unsupervised way.

A considerable amount of long Livestream videos are uploaded to the Internet every day, but it is challenging to understand the main content of the long video quickly. Traditionally, we can only have an inaccurate assumption by reading the video's title or using the control bar to manually access the video, which is time-consuming, inaccurate, and very easy to miss valuable information. An advantageous solution is to segment the long video into small segments based on the topics, making it easier for the users to navigate the content.

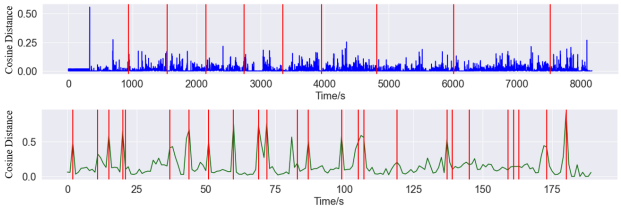


Figure 1. Comparison of temporal-pairwise cosine distance on visual features: (TOP) a Livestream video, (BOTTOM) a TVSum video (Blue & Green: distance; Red: segment boundaries).

Most existing video temporal segmentation work focused on short videos. Some work explored movie clips extracted from long videos but easily segmented temporally by scene change. Jadon et al. proposed a summarization method based on the SumMe dataset [26], which are 1-6 min short videos with clear visual change [31]. When it comes to the long Livestream videos, the previous methods do not work well due to the extremely long length and new characteristics of the Livestream videos. So the critical problem is finding a practical approach to temporally segment the Livestream videos into segments. The quality of segmentation results can significantly impact further tasks. So here we propose a new task, TSLLV, temporal segmentation of long Livestream videos, which has not been explored yet. Different from other long videos, i.e., movies, Livestream videos usually contain more noisy visual information due to the visually abrupt change, and more noisy language information due to random chatting, conversational languages, and intermittent sentences, which means the content is neither clear nor well-organized, making it extremely hard to detect the segment boundaries. Comparison of the visual noisiness of the Livestream video and other videos and examples of Livestream transcripts are introduced in Section 3.

To sum up, the main difficulties for temporally segmenting the Livestream videos are:
(1) The visual background remains similar for a considerable time, even though the topic has already changed, making the definition of boundaries ambiguous. For our MultiLive dataset collected from Behance[1], the hosts usually teach drawing or painting, where the main background is the board and remains similar for most parts of the video. Compared with movies, the movie's background changes dramatically when switching to another scene, so the Livestream videos can not be split directly based on visual scene change or transition differences. Fig. 1 shows an example comparison of temporal-pairwise cosine distance (distance between the ith frame and $i+1$th frame of the same video) on visual feature between a Livestream video and a TVSum video [64], which shows the Livestream video's segment boundaries are not aligned with the visual scene change, making it difficult to segment.
(2) The visual change is neither consistent nor clear. As shown in Fig. 1, there are abrupt changes in the visual site due to the host changing folders or zooming in/out, making the visual information extremely noisy.
(3) There is not enough labeled data for this kind of Livestream video, and it is challenging, time-consuming, and expensive to label them manually. Because it requires the human annotators to watch the entire video, understand the topics, and then temporally segment it, making it much more complicated than labeling images.

Our contributions are listed as follows:

- We introduced MultiLive, a new large dataset of Livestream videos, among which, 1,000 videos were manually segmented and annotated, providing human insights and references for evaluation.
- We formulate a new temporal segmentation of long Livestream videos (TSLLV) task according to the newly introduced MultiLive dataset.
- We proposed **LiveSeg**, an unsupervised **Live**stream temporal **Seg**mentation method by exploring multimodal visual and language information as a solution to TSLLV. We extract features from both modalities, explore the relationship and dependencies across domains, and generate accurate segmentation results. LiveSeg achieved a 16.8% F1-score performance improvement compared with the SOTA method.

2. Related Work

Video Temporal Segmentation Temporal segmentation aims at generating small segments based on the content or topics of the video, which is easy to achieve when the video is short or when the scene change is easy to detect, e.g., in movie clips. Previous works mainly focused on short videos or videos with clear scene changes, which is convenient to manually label a huge amount of videos as training sets for supervised learning [36, 61, 84, 48, 22, 62, 2].

Action, Shot, and Scene Segmentation Temporal action segmentation in videos has been widely explored [74, 83, 38, 23, 37, 59, 76]. However, those videos' characteristics are far different from Livestream videos, where the actions are well-defined, the main goal is to group similar actions based on visual change, and the length of videos is much shorter, so the methods can not be adopted directly. Shot boundary detection task is also very relevant and has been explored in many previous works [28, 66, 29, 3], where shot is defined by the visual change. However, in Livestream videos, segments are not solely defined by visual information, the topics contained in language also contribute to the definition of each segment. Video scene detection is the most relevant task. However, previous methods only used visual information to detect the scene change [52, 56, 57, 11, 81], so the methods can not be adopted directly for Livestream videos either.

Unsupervised Methods Recently, unsupervised methods have also been explored for video temporal segmentation. [34] proposed incorporating multiple feature sources with chunk and stride fusion to segment the video, but the datasets used are still short videos [26, 64]. [20] used Livestream videos as materials. However, they used internal software usage as the segmentation reference, which is not available for most videos, making their method highly restricted. Because for most videos, we can only get access to visual and audio/language metadata.

Summary Although previous models have shown reasonable results, they still suffer some drawbacks. Most work targeted short videos with clear scene changes instead of long videos, and only used visual information while ignoring other domains, like language. Due to the characteristics of the Livestream videos in our MultiLive dataset, methods that solely depend on visual features can not obtain accurate results, so a multimodal approach should be addressed to incorporate visual and language information.

3. MultiLive Dataset

We introduced a large Livestream video dataset from Behance[2], which contains Livestream videos for showcasing and discovering creative work. The dataset includes video ID, title, video metadata, extracted transcript metadata from audio signals (by Microsoft ASR [77]), offset (timestamp), duration of each sentence, etc. The whole dataset contains 11,285 Livestream videos with a total duration of 15,038.4 hours, the average duration per video is 1.3 hours. The entire transcript contains 8,001,901 sentences, and the aver-

[1] https://www.behance.net/live

[2] https://www.behance.net/live

age transcript length for each video is 709 sentences. (An example transcript is shown in the Appendix.) The detailed statistics of the dataset are shown in Table 1 and Table 2. From Tables 1,2, most videos are less than 3 hours and most videos' transcript contains less than 1,500 sentences. In addition, we showed the histogram of video length distribution and transcript length distribution in Fig 2.

Table 1. Distribution of Livestream video duration.

Video Duration	Number	Percentage
0-1 h	4,827	42.774%
1-2 h	2,945	26.097%
2-3 h	2,523	22.357%
3-4 h	705	6.247%
4-5 h	210	1.861%
5-6 h	70	0.620%
6-7 h	11	0.097%

Table 2. Distribution of transcript length.

Transcript Length	Number	Percentage
0-500	5,512	48.844%
500-1,000	2,299	20.372%
1,000-1,500	1,890	16.748%
1,500-2,000	989	8.746%
2,000-2,500	365	3.234%
2,500-3,000	118	1.046%
3,000-3,500	84	0.744%
3,500-4,000	35	0.310%
4,000-4,500	12	0.106%
4,500-5,000	3	0.027%

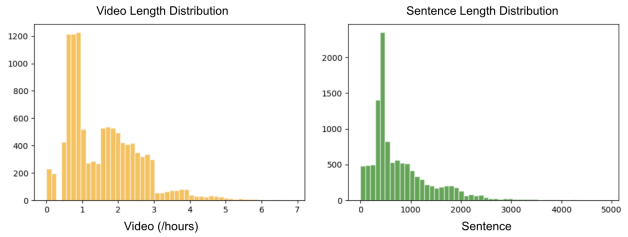


Figure 2. Histogram of MultiLive video length distribution and transcript length distribution (y-axis: number of videos).

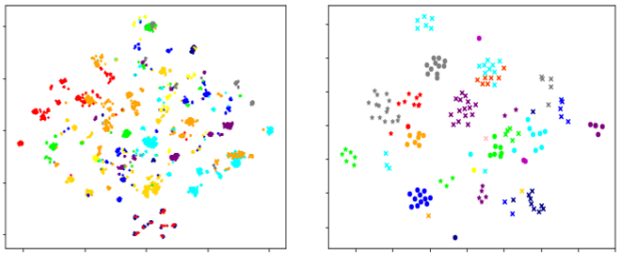
Besides, for the purpose of evaluation, we provide human annotations of 1,000 videos with segmentation boundaries annotated manually by human annotators for evaluation. The human annotators are asked to watch and understand the whole video and split each into several segments based on their understanding of the video content. The current 1,000 videos' annotation includes 10 annotators from Amazon Mechanical Turk [3] (legal agreement signed). The annotators were separated into groups and each group watched part of the videos and then discussed the results together about the segmentation results to ensure the quality of the annotation was agreed upon by all the annotators. They were instructed to pay more attention to topic change, w.r.t. the moment that the live-streamer starts discussing a different topic.

[3]https://www.mturk.com/

Table 3. Comparison of MultiLive with existing datasets.

Statistics	MultiLive	SumMe [26]	TVSum [64]	OVP [6]
Labeled videos	1,000	25	50	50
Ave. length (min)	78 mins	2.4 mins	4.2 mins	1.5 mins
Ave. scene num	8.8	5.1	52.2	8.8
Ave. SLR (min/scene)	8.86	0.47	0.08	0.17
Ave. SD	0.07	0.22	0.19	0.35

There are several widely used video datasets in temporal segmentation or video summarization tasks [26, 64, 6], Table 3 shows the comparison of our dataset with the others. The amount of labeled videos of the others is less than 50, while we provide human annotations for 1,000 videos. The average length of the videos from our dataset is much longer than others, while the number of segments is in the same order of magnitude or even smaller than the others. The effect is that, the average SLR (scene length ratio) of the Livestream dataset is much larger, where average SLR (scene length ratio) can be considered a metric to represent the average length of each scene in the video, calculated by (ave.length / ave. scene num). So the larger the ratio, the more content contained in each segment, leading to more difficulty finding the segment boundaries.



(a) Livestream video (b) TVSum video

Figure 3. (a) Visual features of a Livestream video; (b)Visual features of a TVSum video, where different colors represent different segments within one video.

To demonstrate a more precise understanding of the visual information of Livestream videos, we compared the visual features extracted from one example Livestream video and one example TVSum video [64]. We extracted video frames from the raw video sequence, used ResNet50 model [30] (pre-trained on ImageNet) to extract the visual features of each video frame, and adopted t-SNE [69] to visualize the visual features. Fig. 3(a) shows the Livestream video's visual feature distribution, different colors with the same marker "o" representing different segments, ten segments in total. We can find that feature points which belong to different segments mix together and thus are hard to separate. As for TVSum's video result in Fig. 3(b), different

color or different marker "o"/"x"/"∗" all represents different segments, 23 segments in total, which shows the points belong to different segments can be distinguished more easily than the Livestream video. This proves our statement that Livestream videos contain more noisy visual information, making it much harder to be temporally segmented by traditional methods.

Table 4. Comparison of different type of videos.

Statistics	ASR WER	USR
Film Corpus [70, 71]	0.01	0.126
Movie Dialog Corpus [1]	0.01	0.139
MultiLive	0.05	0.458

Table 4 also shows the comparison of our Livestream data with movie datasets [70, 71, 1], which were collected from IMDB and TMDB, to emphasize the differences between Livestream videos and movies. Table 4 shows the Livestream videos' ASR WER (word error rate) is higher than movies, and the USR (unrelated sentence rate) is much higher than movies, which contain more meaningless conversational languages. We further used hierarchical clustering to group the frames based on visual features and generated a dendrogram. As shown in Fig. 4, we could find that the video frames far away from each other in timestamp can still be clustered together into the same group if only visual features are used. It supports the claim that using only visual information is insufficient to generate accurate temporal segmentation results, as the visual domain lacks sufficient information. So other domain features should be explored to provide more information.

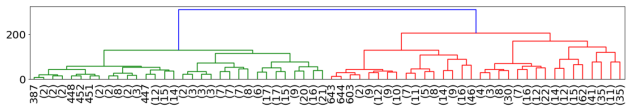


Figure 4. Dendrogram result of one Livestream video by hierarchical clustering of visual features, where the numbers below the bottom layer represent the number of images belongs to the corresponding sub-tree.

To show a representative comparison, we computed the frame-level average distance (ave. SD) between the segments of our MultiLive dataset and the SumMe, TVSum, and OVP datasets. The results are shown in Table 3. The distance is computed on the two adjacent frames on each video segment boundary (last frame of ith segment, and first frame of $(i+1)$th segment, and the average results could show the average visual difference comparison. As in Table 3, we can find that the ave. SD of the MultiLive dataset is much smaller than the ave. SD of other datasets, which could be a representative metric to demonstrate that Livestream video's visual change is much more noisy than existing datasets, making it more difficult to segment.

4. LiveSeg: Unsupervised Multimodal Temporal Segmentation of Livestream Videos

The TSLLV task (temporal segmentation of long Livestream video) aims to accurately and temporally segment the Livestream videos based on the topics. Due to the absence of segmented labels and the time-consuming of manually labeling a huge amount of such long videos, we adopt unsupervised methods to segment the Livestream videos temporally. The whole framework is shown in Fig. 5. Given a Livestream video $\mathcal{S}$, our target is to temporally segment video $\mathcal{S}$ into $[S_1, S_2, ..., S_k]$ based on topics, where k is the number of segments. The only available materials are video (visual input) and transcripts (language input). The number of segments of each to-be-segmented video is not preliminary given.

4.1. LiveSeg Framework

The LiveSeg model takes input from the visual domain and the language domain. For visual features, we sample video frames $[f_1, f_2, ..., f_n]$, where n is the timestamp, from the raw video $\mathcal{S}$ (one frame per second to reduce the computation complexity). Then we use ResNet-50 [30] pretrained on ImageNet [58] to extract visual features (fingerprints) $V_1 = [V_{11}, V_{12}, ..., V_{1n}]$, where the visual fingerprints represent the video content. For the language features, due to the fact that the transcript is not temporally perfectly aligned with video frames, we first assign the transcript sentences to the corresponding video frame. If there are overlaps between several sentences or several frames, we duplicate those in a corresponding manner, and formulate frame-transcript pairs for each sampled frame in the timeline. Since the frames are sampled by a one-second time window, the transcripts are also aligned with each time window. If one transcript sentence T_i does not end for the given window, meaning the language has overlapped with two adjunct time windows, then we will assign this sentence T_i to both time window t and time window $t+1$. We then extract sentence embeddings with BERT [16] to get sentence-level representations $L_1 = [L_{11}, L_{12}, ..., L_{1n}]$. The embedding model used in our formulation is "all-MiniLM-L6-v2" from Sentence-Transformers [54], which is trained on large sentence level datasets using a self-supervised contrastive learning objective from pre-trained model [75] and fine-tuned on a sentence pairs dataset. Due to the ambiguity of the transcript, i.e., the examples are shown in the Appendix, redundant and noisy words are removed before generating language embeddings (redundant and noisy words mean the words that appear more than three times in a row due to the live-streamer's speaking error).

The previous work [41, 9, 32] which took advantage of the alignment of vision and language features inspired us to assume that there should be a relationship and dependency

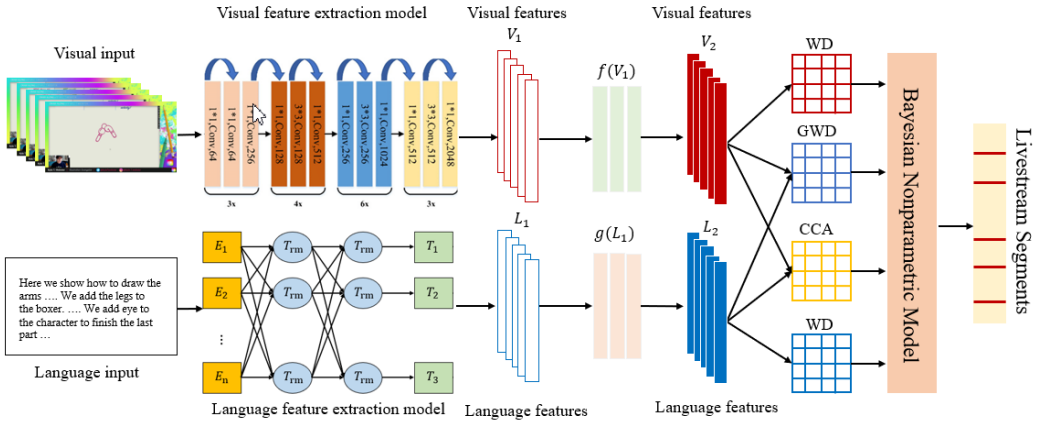


Figure 5. The framework of LiveSeg for unsupervised multimodal Livestream video temporal segmentation.

between visual and language features. [10, 40, 72, 12, 35] find that Optimal Transport shows tremendous power in sequence-to-sequence learning. In addition, [9, 80] find that Gromov Wasserstein Distance shows even better performance in measuring the distances in counterpart domains. Canonical Correlation Analysis, a well-known approach to exploring the correlation between different modalities, has been studied in many previous works for its ability to recognize the cross-domain relationship [4, 79, 78, 25]. [67, 44] showed that Bayesian Nonparametric Models performed well on temporal segmentation task, especially under unsupervised settings, which stands as a good candidate for our TSLLV task. Therefore, we adopt Deep Canonical Correlation Analysis [4] to encode the dependency for a hierarchical feature transformation. The networks transform raw visual features V_1 to high-level visual features V_2 with the transformation $f(V_1)$, and transform raw language features L_1 to high-level language features L_2 with the transformation $g(L_1)$. Then we compute the Wasserstein Distance (WD) on the high-level temporal visual features V_2 and language features L_2. We also calculate the Gromov Wasserstein Distance (GWD) and Canonical Correlation Analysis (CCA) on the two different modalities at the same timestamp, then use Bayesian Nonparametric models [33] to segment the Livestream videos temporally. The details of each part are introduced in the following paragraphs and sections. More details about WD, GWD, and CCA can also be found in the Appendix.

Wasserstein Distance Wasserstein Distance (WD) is introduced in Optimal Transport (OT), which is a natural type of divergence for registration problems as it accounts for the underlying geometry of the space, and has been used for multimodal data matching and alignment tasks [9, 80, 39, 15, 27, 50]. In Euclidean settings, OT introduces WD $\mathcal{W}(\mu, \nu)$, which measures the minimum effort required to "displace" points across measures μ and ν, where μ and ν are values observed in the empirical distribution. In our setting, we compute the temporal-pairwise Wasserstein Distance on both visual features and language features, considering each feature vector representing each frame or transcript embedding. The temporal-pairwise WD on both visual and language features encodes the temporal difference and consistency within the same domain.

Gromov Wasserstein Distance Classic OT requires defining a cost function across domains, which can be challenging to implement when the domains are in different dimensions [53]. Gromov Wasserstein Distance (GWD) [46] extends OT by comparing distances between samples rather than directly comparing the samples themselves. In our framework, the computed GWD across domains is to capture the relationship and dependencies between visual and language domains.

CCA and DCCA Canonical Correlation Analysis (CCA) is a method for exploring the relationships between two multivariate sets of variables, which can learn the linear transformation of two vectors in order to maximize the correlation between them, which is used in many multimodal problems [4, 51, 42, 7, 24, 43]. In our problem, we apply CCA to capture the cross-domain relationship of visual features V_{2l} and language features L_{2l}. To obtain V_2 and L_2, DCCA is applied in the framework for nonlinear feature transformation. The parameters are trained to optimize this quantity using gradient-based optimization by taking the correlation as the negative loss with backpropagation to update the nonlinear transformation model [4]. More details can be found in the Appendix.

4.1.1 Bayesian Nonparametric Model

We used Hierarchical Dirichlet Process Hidden semi-Markov Model (HDP-HSMM) to generate the video segments for modeling [33, 21], which can infer arbitrarily large state complexity from sequential and time-series data.

More discussion about HMM, HSMM, and their drawbacks are introduced in the Appendix.

The process of HDP-HSMM is illustrated in Fig. 6. In the model, z_i denotes the classes of the segments, β denotes an infinite-dimensional multinomial distribution, which is generated from the GEM distribution and parameterized by γ [47]. GEM denotes the co-authors Griffiths, Engen, and McCloskey, with the so-called stick-breaking process (SBP) [60]. The probability π_i denotes the transition probability, which is generated by the Dirichlet process and parameterized by β [68]:

$$\beta \mid \gamma \sim \mathrm{GEM}(\gamma),\ \ \pi_i \mid \alpha, \beta \sim \mathrm{DP}(\alpha, \beta), i = 1, 2, \cdots, \infty \tag{1}$$

where γ and α are the concentration parameters of the Dirichlet processes (DP). The probability distribution is constructed through a two-phase DP named Hierarchical Dirichlet process (HDP) [68, 44]. The class z_i of the ith segment is determined by the class of the $(i-1)$th segment and transition probability π_i.

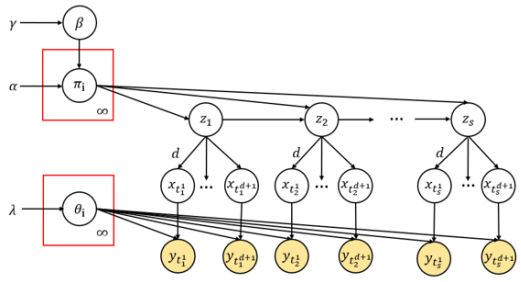


Figure 6. Graphical model of HDP-HSMM

In HSMM, state transition probability from state i to j can be defined as $\pi_{i,j} = p\left(x_{t+1} = j \mid x_t = i\right)$, then the transition matrix can be denoted as $\pi = \{\pi_{i,j}\}_{i,j=1}^{|\chi|}$, where $|\chi|$ denotes the number of hidden states. The distribution of observations y_t given specific hidden states is denoted by $p\left(y_t \mid x_t, \theta_i\right)$, where θ_i denotes the emission parameter of state i. Then the HSMM can be described as:

$$x_s \mid x_{s-1} \sim \pi_{x_{s-1}}, d_s \sim g\left(\omega_s\right), y_t \mid x_s, d_s \sim F\left(\theta_{x_s}, d_s\right) \tag{2}$$

where $F(\cdot)$ is an indexed family of distributions, the probability mass function of d_s is $p\left(d_t \mid x_t = i\right)$, $g\left(\omega_s\right)$ denotes a state-specific distribution over the duration d_s, and ω_s denotes the parameter priori of the duration distributions.

In HDP, let Θ be a measurable space with a probability measure H on the space, γ is a positive real number named the concentration parameter. $\mathrm{DP}(\gamma, H)$ is defined as the distribution of the random probability measure of G over Θ. For any finite measurable partition of Θ, the vector is distributed as a finite-dimensional Dirichlet distribution:

$$G_0 \sim \mathrm{DP}(\gamma, H), G_0 = \sum_{k=1}^{K} \beta_k \delta_{\theta_k}, \theta_k \sim H, \beta \sim \mathrm{GEM}(\gamma) \tag{3}$$

where θ_k is the distribution of H, $\beta \sim \mathrm{GEM}(\gamma)$ represents the stick-breaking construction process of the weight coefficient [45, 85], and δ_θ is the Dirac function. The model can be written as:

$$\theta_i \sim H(\lambda), i = 1, 2, \cdots, \infty,\ \ z_s \sim \bar{\pi}_{z_{s-1}}, s = 1, 2, \cdots, S \tag{4}$$

$$D_s \sim g\left(\omega_{z_s}\right), s = 1, 2, \cdots, S,\ \ \omega_i \sim \Omega \tag{5}$$

$$x_{t_s^1 : t_s^{D_s+1}} = z_s,\ \ y_{t_s^1 : t_s^{D_s+1}} \sim \mathrm{F}\left(\theta_{x_t}\right) \tag{6}$$

where π_i is the distribution parameter of hidden state sequence z_s, implying that HDP provides an infinite number of states for HSMM, D_s is the length distribution of the state sequence with distribution parameter ω, and y_{t_s} is the observation sequence with distribution parameter θ_i [49].

For parameter inference of the HDP-HSMM model, a weak-limit Gibbs sampling algorithm is applied [33]. The weak limit approximation transforms the infinite dimension hidden state into finite dimension form so that the hidden state chain can be updated according to the observation data [49]. It is assumed that the basic distribution $H(\cdot)$ and the observation series distribution $F(\cdot)$ are conjugated distributions, the hidden states distribution $g(\cdot)$ is a Poisson distribution, and the hidden states distribution and the observation series distribution are independent. We first sample the weight coefficient β and the state sequence distribution parameter π_i:

$$\beta \mid \gamma \sim \mathrm{Dir}(\gamma/S, \cdots, \gamma/S) \tag{7}$$

$$\pi_i \mid \alpha, \beta \sim \mathrm{Dir}\left(\alpha\beta_1, \cdots, \alpha\beta_s\right) j = 1, \cdots S \tag{8}$$

Then we sample the observation distribution parameters θ_i and state duration distribution parameter ω_i according to observation data. It is assumed that the observed data obey a multivariate Gaussian distribution, the model parameters $\theta_i = (u_i, \Sigma_i)$ obey the Normal–Inverse–Wishart distribution:

$$\mathrm{NIW}\left(u, \Sigma \mid v_0, \Delta_0, \mu_0, S_0\right) \triangleq \mathrm{N}\left(\mu \mid \mu_0, S_0\right) * \mathrm{IW}\left(\Sigma \mid v_0, \Delta_0\right) \tag{9}$$

where $\varphi = \{u_0, S_0, v_0, \Delta_0\}$ are prior parameters, μ_0 and S_0 are the prior mean and co-variance matrices, and ν_0 and Δ_0 are the degrees of freedom and scale of NIW distribution. The state duration distribution is a Poisson distribution, and parameter ω_i follows a Beta distribution: $\omega \sim \mathrm{Beta}\left(\eta_0, \sigma_0\right)$. Then we update parameters according to the observation data [33, 19].

4.1.2 Final Segmentation Boundaries

The raw output from the Bayesian nonparametric model contains both short and long segments, but the short segment may not contain comprehensive information, which will be useless as the final results. So we used a heuristic-based method to group the small segments into the large ones. The method is straightforward, where a parameter l_s defines the minimum length of the generated segments, if there is a segment shorter than l_s, then we compute the visual and textual similarity of this small segment with the two adjacent segments, and group the small segment into

the one with higher similarity. Since these small segments are mostly due to the live-streamer abruptly zooming in/out or randomly chatting about something unrelated to the main topic, which has little influence on the segmentation results (i.e., a small part inside a big chunk), we just used this simple method to make the results look more cleaner. The method introduced a parameter l_s, defined as the minimum length of the generated segments, which is used to hierarchically group the generated small segments into the bigger ones to eliminate the effect caused by small segments.

4.2. Baseline Methods

We select several strong and representative baseline methods for comparison, which include:

- **Hierarchical Cluster Analysis (HCA)** HCA aims at finding discrete groups with varying degrees of similarity represented by a similarity matrix [17, 14], which produces a dendrogram as the intermediate result. The distance for the Livestream video setting is defined as: $d = \alpha_b d_t + (1 - \alpha_b) d_f$, where d_t is the timestamp distance, d_f is the feature content distance, and α_b is a balance parameter. Feature points representing content get separated further apart when the time distance of corresponding features is large.
- **TransNet V2** Soucek et al. proposed TransNet V2 model for shot transition detection [65], which can also generate segmentation results and showed better performance than previous method [66].
- **Hecate** Song et al. proposed the Hecate model to generate thumbnails, animated GIFs, and video summaries from videos [63], where shot boundary detection is one of the steps. This step will be used to compare with the other baseline methods as well as our method.
- **Optimal Sequential Grouping (OSG)** Rotman et al. proposed video scene detection algorithms based on the optimal sequential grouping [56, 57], which included finding pairwise distances between feature vectors and splitting shots into non-intersecting groups by optimizing a distance-based cost function.
- **LGSS** Rao et al. proposed a local-to-global scene segmentation framework (LGSS) [52], which used multiple features extracted by ResNet50, Faster-RCNN [55], TSN [73], and NaverNet [13]. The temporal segmentation step is based on PySceneDetect [8].

5. Experiments and Results

5.1. Temporal Segmentation on MultiLive Dataset

For Livestream videos, the raw visual feature dimension is 2,048, and the raw language feature dimension is 384 extracted by pre-trained-BERT models from Huggingface[4]. For the hierarchical transformation performed by DCCA, the network architecture and parameters are shown in the Appendix. In our experiments, we set l_s to one minute. To make a fair comparison, we also applied the post-processing step in Sec 4.1.2 to all the baselines.

Due to the characteristic of Livestream videos, the video frames and transcripts are not perfectly aligned with the segments. Besides, different people segment the same video differently due to human preferences, which also needs to be considered. We performed the TSLLV task using baseline methods in Section 4.2 and our LiveSeg method on the MultiLive dataset. We evaluate the performance of different methods on the 1,000 annotated videos. Comparison results of baseline methods, our method, and human annotations for one Livestream video are shown in Fig. 7. We can see that scene transition detection methods will generate inaccurate segments as the visual change is noisy for Livestream videos. However, many essential boundaries will be missed if simply improving the clustering threshold. Compared with existing methods, our results are more accurate and can be comparable with human annotations.

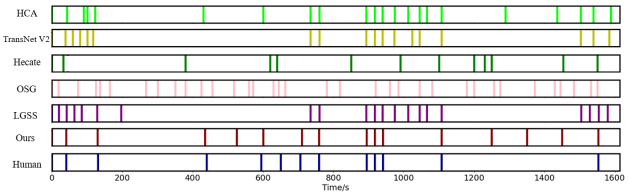


Figure 7. Comparison of boundary candidates by different methods, from top to bottom: (1) HCA [14], (2) TransNet V2 [66], (3) Hecate [63], (4) OSG [57], (5) LGSS [52], (6) ours (LiveSeg), and (7) Human Annotations.

Table 5. Comparison of segmentation results.

Methods	Backbone	Modality	Precision	Recall	F1-score
HCA [14]	HCA	Visual	0.482	0.487	0.484
TransNet V2 [66]	ResNet-18	Visual	0.536	0.525	0.530
Hecate [63]	Clustering	Visual	0.539	0.533	0.536
OSG [57]	DP	Visual	0.574	0.557	0.565
LGSS [52]	Bi-LSTM	Visual	0.587	0.581	0.584
LiveSeg-Visual	LiveSeg	Visual	0.591	0.666	0.626
LiveSeg-Language	LiveSeg	Language	0.589	0.568	0.578
LiveSeg-Multimodal	LiveSeg	Multimodal	**0.673**	**0.697**	**0.685**

For quantitative analysis, tolerance interval ω_t is introduced. The correctness of the segmentation is judged at each position of this interval: a false alarm is declared if the algorithm claims a boundary in the interval while no reference boundary exists in the interval, and a miss is declared if the algorithm does not claim a boundary in the interval while a reference boundary exists in the interval [18]. In our experiment, we set ω_t to one minute, and we adopt precision, recall, and F1-score metrics to compare the performance of our results with human annotations. As shown in Table 5, our segmentation results outperform other baseline results. Besides, considering modality, multimodal segmentation outperforms single modality results, showing that the

[4]https://huggingface.co/

relationship learned between the visual domain and the language domain can truly benefit temporal segmentation.

5.2. Ablation Study

Multiple heuristics could have an impact on the segmentation performance, such as tolerance interval ω_t and the parameters in the Bayesian nonparametric model. We carried out several ablation experiments on the influence of different parameters with multimodal features, where the results are shown in Table 6 and Fig. 8. More detailed results are shown in the Appendix due to the page limit.

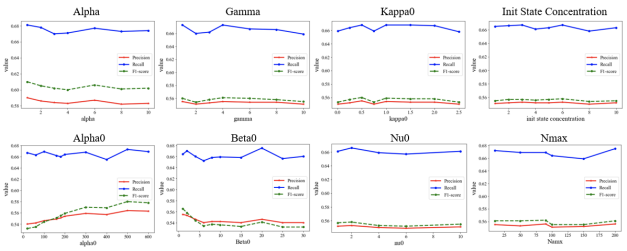


Figure 8. Segmentation performance with different parameters (Red: precision; Blue: recall; Green: F1-score).

Table 6. Comparison of performance with different interval ω_t.

ω_t	Precision	Recall	F1-score
0.5 min	0.608	0.672	0.627
1.0 min	0.673	0.697	0.685
1.5 min	0.605	0.666	0.621
2.0 min	0.600	0.659	0.615
2.5 min	0.598	0.653	0.610
3.0 min	0.595	0.647	0.605

We also provided an ablation study on different components, since only using WD is the same as LiveSeg-Visual and LiveSeg-Language, so we provide additional ablation study results on GWD and CCA. In Table 7, we can find that combining all of them (LiveSeg-Multimodal) can achieve better performance than using only one of the components.

Table 7. Ablation study of different components.

	Precision	Recall	F1-score
GWD	0.622	0.673	0.646
CCA	0.603	0.654	0.615
WD (LiveSeg-Visual)	0.591	0.666	0.605
WD (LiveSeg-Language)	0.589	0.568	0.606

5.3. Comparison on Other Datasets

In addition, we compare our method with state-of-the-art unsupervised video summarization method [5] on the famous video summarization benchmark datasets, SumMe [26] and TVSum [64]. We used the same key-fragment-based approach for evaluation [82], where the similarity between a machine-generated and a user-defined ground-truth summary is represented by expressing their overlap using the F-Score. For a given video and a machine-generated summary, this protocol matches the latter against all the available user summaries for this video and computes a set of F-Scores. More details of this experiment are shown in the Appendix. Table 8 shows the comparison F1-score of our method with SUM-GAN [5], our method can still show slightly better results on the SumMe dataset and competitive results on the TVSum dataset, which clearly demonstrated the effectiveness of our method.

Table 8. Comparison with SOTA unsupervised baseline on traditional video summarization datasets.

F1-score	LiveSeg	SUM-GAN [5]
SumMe	51.3	50.8
TVSum	60.9	60.6

6. Discussion of Limitation and Future Work

The current method targets long Livestream videos, which shows better performance than existing ones, given that the current setting of both visual input and language input are highly noisy. However, it may not work better than supervised methods on short videos where scene change is clear, under which supervised methods could perform better when large-scale labeled training samples are available. However, the collected training samples highly constrain the generability and robustness of those approaches.

Due to the fact that labeling large-scale long videos is time-consuming and expensive, so the current annotation result can be considered as the average result, which ensures the general quality, while may not preserve the nature that individual annotator may have different preferences, which can be useful as user-study materials. In future work, we will execute annotations for the same videos by different annotators separately, for evaluation and verification, which could provide human upper bound and future insights.

7. Conclusion

In this paper, we proposed LiveSeg, an unsupervised multimodal framework, focusing on temporal segmentation of long Livestream video (TSLLV) task, which has not been explored before. We collected a large Livestream video dataset named MultiLive, and provided human annotations of 1,000 Livestream videos for evaluation. By quantitative analysis and human evaluation of our experimental results, we demonstrated that our model is able to generate high-quality temporal segments, which established the basis for Livestream video understanding tasks and can be extended to many real-world applications.

Acknowledgement

We sincerely appreciate the inspiration and guidance from Jiacheng Zhu, Bo Li, and Daniel Fried.

References

[1] Movie dialog corpus: A metadata-rich collection of fictional conversations from raw movie scripts.

[2] Sathyanarayanan N. Aakur and Sudeep Sarkar. A perceptual prediction framework for self supervised event segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1197–1206, 2019.

[3] Sadiq H. Abdulhussain, Abd. Rahman bin Ramli, M. Iqbal Saripan, Basheera M. Mahmmod, Syed Abdul Rahman Al-Haddad, and Wissam A. Jassim. Methods and challenges in shot boundary detection: A review. *Entropy*, 20, 2018.

[4] Galen Andrew, R. Arora, Jeff A. Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, 2013.

[5] Evlampios E. Apostolidis, E. Adamantidou, Alexandros I. Metsai, Vasileios Mezaris, and I. Patras. Ac-sum-gan: Connecting actor-critic and generative adversarial networks for unsupervised video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, 31:3278–3292, 2021.

[6] Sandra Avila, Ana Paula Brandão Lopes, Antonio da Luz, and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognit. Lett.*, 32:56–68, 2011.

[7] Lan-Qing Bao, Jie-Lin Qiu, Hao Tang, Wei-Long Zheng, and Bao-Liang Lu. Investigating sex differences in classification of five emotions from eeg and eye movement signals. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6746–6749, 2019.

[8] Brandon Castellano. Intelligent scene cut detection and video splitting tool. `https://bcastell.com/projects/PySceneDetect/`, 2021.

[9] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. *ArXiv*, abs/2006.14744, 2020.

[10] Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. Improving sequence-to-sequence learning via optimal transport. *ArXiv*, abs/1901.06283, 2019.

[11] Shixing Chen, Xiaohan Nie, David D. Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot contrastive self-supervised learning for scene boundary detection. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9791–9800, 2021.

[12] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[13] Joon Son Chung. Naver at activitynet challenge 2019 - task b active speaker detection (ava). *ArXiv*, abs/1906.10555, 2019.

[14] V. Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and C. Mathieu. Hierarchical clustering: Objective functions and algorithms. In *SODA*, 2018.

[15] Pinar Demetci, Rebecca Santorella, Björn Sandstede, William Stafford Noble, and Ritambhara Singh. Gromov-wasserstein optimal transport to align single-cell multi-omics data. *bioRxiv*, 2020.

[16] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[17] B. Everitt and A. Skrondal. Comprar the cambridge dictionary of statistics — b. s. everitt — 9780521766999 — cambridge university press. 2010.

[18] Jonathan G. Fiscus, George R. Doddington, John S. Garofolo, and Alvin F. Martin. Nist's 1998 topic detection and tracking evaluation (tdt2). In *EUROSPEECH*, 1999.

[19] Emily B. Fox. Bayesian nonparametric learning of complex dynamical phenomena. 2009.

[20] C. Ailie Fraser, Joy Kim, Hijung Shin, Joel Brandt, and Mira Dontcheva. Temporal segmentation of creative live streams. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.

[21] Daniel Fried, Jean-Baptiste Alayrac, Phil Blunsom, Chris Dyer, Stephen Clark, and Aida Nematzadeh. Learning to segment actions from observation and narration. In *ACL*, 2020.

[22] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Personal-location-based temporal segmentation of egocentric video for lifelogging applications. *Journal of Visual Communication and Image Representation*, 52:1–12, 2018.

[23] Zhanning Gao, Le Wang, Qilin Zhang, Zhenxing Niu, Nanning Zheng, and Gang Hua. Video imprint segmentation for temporal action detection in untrimmed videos. In *AAAI*, 2019.

[24] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision*, 106:210–233, 2013.

[25] Chenfeng Guo and Dongrui Wu. Canonical correlation analysis (cca) based multi-view learning: An overview. *ArXiv*, abs/1907.01693, 2019.

[26] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[27] William Han, Jielin Qiu, Jiacheng Zhu, Mengdi Xu, Douglas Weber, Bo Li, and Ding Zhao. An empirical exploration of cross-domain alignment between language and electroencephalogram. *ArXiv*, abs/2208.06348, 2022.

[28] Ahmed Hassanien, Mohamed A. Elgharib, Ahmed A. S. Seleim, Mohamed Hefeeda, and Wojciech Matusik. Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. *ArXiv*, abs/1705.03281, 2017.

[29] Eman Hato and Matheel Emaduldeen Abdulmunem. Fast algorithm for video shot boundary detection using surf features. *2019 2nd Scientific Conference of Computer Sciences (SCCS)*, pages 81–86, 2019.

[30] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[31] Shruti Jadon and Mahmood Jasim. Unsupervised video summarization framework using keyframe extraction and video skimming. *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pages 140–145, 2020.

[32] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[33] Matthew J. Johnson and Alan S. Willsky. Bayesian nonparametric hidden semi-markov models. *J. Mach. Learn. Res.*, 14:673–701, 2013.

[34] Hussain Kanafani, Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. Unsupervised video summarization via multi-source features. *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 2021.

[35] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.

[36] Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal Process. Image Commun.*, 16:477–500, 2001.

[37] Hilde Kuehne, Alexander Richard, and Juergen Gall. A hybrid rnn-hmm approach for weakly supervised temporal action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:765–779, 2020.

[38] Colin S. Lea, Michael D. Flynn, René Vidal, Austin Reiter, and Gregory Hager. Temporal convolutional networks for action segmentation and detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012, 2017.

[39] John Lee, Max Dabagia, Eva L. Dyer, and Christopher J. Rozell. Hierarchical optimal transport for multimodal distribution alignment. *ArXiv*, abs/1906.11768, 2019.

[40] Chi-Heng Lin, Mehdi Azabou, and Eva L. Dyer. Making transport more robust and interpretable by moving data through a small number of anchor points. *Proceedings of machine learning research*, 139:6631–6641, 2021.

[41] Fenglin Liu, Yuanxin Liu, Xuancheng Ren, Xiaodong He, and Xu Sun. Aligning visual regions and textual concepts for semantic-grounded image representations. In *NeurIPS*, 2019.

[42] Wei Liu, Jie-Lin Qiu, Wei-Long Zheng, and Bao-Liang Lu. Multimodal emotion recognition using deep canonical correlation analysis. *ArXiv*, abs/1908.05349, 2019.

[43] Wei Liu, Jie-Lin Qiu, Wei-Long Zheng, and Bao-Liang Lu. Comparing recognition performance and robustness of multimodal deep learning models for multimodal emotion recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 14:715–729, 2022.

[44] Masatoshi Nagano, Tomoaki Nakamura, Takayuki Nagai, Daichi Mochihashi, Ichiro Kobayashi, and Wataru Takano. Hvgh: Unsupervised segmentation for high-dimensional time series using deep neural compression and statistical generative model. *Frontiers in Robotics and AI*, 6, 2019.

[45] John William Paisley, Aimee K. Zaas, Christopher W. Woods, Geoffrey S. Ginsburg, and Lawrence Carin. A stick-breaking construction of the beta process. In *ICML*, 2010.

[46] Gabriel Peyré, Marco Cuturi, and Justin M. Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *ICML*, 2016.

[47] Jim Pitman. Poisson–dirichlet and gem invariant distributions for split-and-merge transformations of an interval partition. *Combinatorics, Probability and Computing*, 11:501 – 514, 2002.

[48] Yair Poleg, Chetan Arora, and Shmuel Peleg. Temporal segmentation of egocentric videos. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2544, 2014.

[49] Xuqiang Qiao, Ling Zheng, Yinong Li, Yuqing Ren, Zhida Zhang, Ziwei Zhang, and Lihong Qiu. Characterization of the driving style by state–action semantic plane based on the bayesian nonparametric approach. *Applied Sciences*, 2021.

[50] Jielin Qiu, Jiacheng Zhu, Mengdi Xu, Franck Dernoncourt, Trung Bui, Zhaowen Wang, Bo Li, Ding Zhao, and Hailin Jin. Semantics-consistent cross-domain summarization via optimal transport alignment. volume abs/2210.04722, 2022.

[51] Jie-Lin Qiu, W. Liu, and Bao-Liang Lu. Multi-view emotion recognition using deep canonical correlation analysis. In *ICONIP*, 2018.

[52] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10143–10152, 2020.

[53] Ivegen Redko, Titouan Vayer, Rémi Flamary, and Nicolas Courty. Co-optimal transport. *ArXiv*, abs/2002.03731, 2020.

[54] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019.

[55] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[56] Daniel Rotman, Dror Porat, and Gal Ashour. Robust and efficient video scene detection using optimal sequential grouping. `https://github.com/ivi-ru/video-scene-detection`, 2016.

[57] Daniel Rotman, Dror Porat, Gal Ashour, and Udi Barzelay. Optimally grouped deep features using normalized cost for video scene detection. `https://github.com/ivi-ru/video-scene-detection`, 2018.

[58] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[59] M. Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11220–11229, 2021.

[60] Jay Sethuraman. A constructive definition of dirichlet priors. 1991.

[61] Panagiotis Sidiropoulos, Vasileios Mezaris, Yiannis Kompatsiaris, Hugo Meinedo, Miguel M. F. Bugalho, and Isabel Trancoso. Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, 21:1163–1177, 2011.

[62] Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Ndedi Monekosso, and Paolo Remagnino. Superframes, a temporal video segmentation. *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 566–571, 2018.

[63] Yale Song, Miriam Redi, Jordi Vallmitjana, and Alejandro Jaimes. To click or not to click: Automatic selection of beautiful thumbnails from videos. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016.

[64] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187, 2015.

[65] Tom'avs Souvcek and Jakub Lokovc. Transnet v2: An effective deep network architecture for fast shot transition detection. volume abs/2008.04838, 2020.

[66] Shitao Tang, Litong Feng, Zhanghui Kuang, Yimin Chen, and Wayne Zhang. Fast video shot transition localization with deep structured models. In *ACCV*, 2018.

[67] Yee Whye Teh, Gatsby, and Michael I. Jordan. Hierarchical bayesian nonparametric models with applications. 2008.

[68] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566 – 1581, 2006.

[69] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[70] Marilyn A. Walker, Ricky Grant, Jennifer Sawyer, Grace I. Lin, Noah Wardrip-Fruin, and Michael Buell. Perceived or not perceived: Film character models for expressive nlg. In *ICIDS*, 2011.

[71] Marilyn A. Walker, Grace I. Lin, and Jennifer Sawyer. An annotated corpus of film dialogue for learning and characterizing character style. In *LREC*, 2012.

[72] Jianfeng Wang, Xiaowei Hu, Zhe Gan, Zhengyuan Yang, Xiyang Dai, Zicheng Liu, Yumao Lu, and Lijuan Wang. Ufo: A unified transformer for vision-language representation learning. *ArXiv*, abs/2111.10023, 2021.

[73] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. *ArXiv*, abs/1608.00859, 2016.

[74] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2740–2755, 2019.

[75] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *ArXiv*, abs/2002.10957, 2020.

[76] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. Boundary-aware cascade networks for temporal action segmentation. In *ECCV*, 2020.

[77] Wayne Xiong, L. Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5934–5938, 2018.

[78] Ran Xu, Caiming Xiong, Wei Chen, and Jason J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015.

[79] Fei Yan and Krystian Mikolajczyk. Deep correlation for matching images and text. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3441–3450, 2015.

[80] Siyang Yuan, Ke Bai, Liqun Chen, Yizhe Zhang, Chenyang Tao, Chunyuan Li, Guoyin Wang, Ricardo Henao, and Lawrence Carin. Advancing weakly supervised cross-domain alignment with optimal transport. *ArXiv*, abs/2008.06597, 2020.

[81] Haoxin Zhang, Zhimin Li, and Qinglin Lu. Better learning shot boundary detection via multi-task. *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[82] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, 2016.

[83] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2933–2942, 2017.

[84] Feng Zhou, Fernando De la Torre, and Jessica K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:582–596, 2013.

[85] Jian Zhou, Fei-Yue Wang, and Da jun Zeng. Hierarchical dirichlet processes and their applications: a survey. *Acta Automatica Sinica*, 37:389–407, 2011.