BaSSL: Boundary-aware Self-Supervised Learning for Video Scene Segmentation

Jonghwan Mun[1,⋆,†] Minchul Shin[1,⋆] Gunsoo Han[1]
Sangho Lee[2] Seongsu Ha[2] Joonseok Lee[2,†] Eun-Sol Kim[3,†]

[1]Kakao Brain
[2]Graduate School of Data Science, Seoul National University
[3]Department of Computer Science, Hanyang University
[1]{jason.mun,craig.starr,coco.han}@kakaobrain.com
[2]{sangho.lee,sha17,joonseok}@snu.ac.kr [3]eunsolkim@hanyang.ac.kr

Abstract. Self-supervised learning has drawn attention through its effectiveness in learning in-domain representations with no ground-truth annotations; in particular, it is shown that properly designed pretext tasks bring significant performance gains for downstream tasks. Inspired from this, we tackle video scene segmentation, which is a task of temporally localizing scene boundaries in a long video, with a self-supervised learning framework where we mainly focus on designing effective pretext tasks. In our framework, given a long video, we adopt a sliding window scheme; from a sequence of shots in each window, we discover a moment with a maximum semantic transition and leverage it as pseudo-boundary to facilitate the pre-training. Specifically, we introduce three novel boundary-aware pretext tasks: 1) Shot-Scene Matching (SSM), 2) Contextual Group Matching (CGM) and 3) Pseudo-boundary Prediction (PP); SSM and CGM guide the model to maximize intra-scene similarity and inter-scene discrimination by capturing contextual relation between shots while PP encourages the model to identify transitional moments. We perform an extensive analysis to validate effectiveness of our method and achieve the new state-of-the-art on the MovieNet-SSeg benchmark. The code is available at https://github.com/kakaobrain/bassl

Keywords: Video scene segmentation · Self-supervised learning

1 Introduction

Understanding long videos such as movies, for an AI system, has been viewed as an extremely challenging task. [1] In contrast, for humans, as studies in cognitive science [49] tell us it is naturally achieved by breaking down a video into meaningful units (*e.g.*, event) and reasoning about these units and their relation [42]. From this point of view, dividing a long video into a series of shorter temporal segments can be considered as an essential step towards the high-level video understanding. Motivated by this, in this paper, we tackle the video scene

⋆ Equal contribution † Corresponding authors

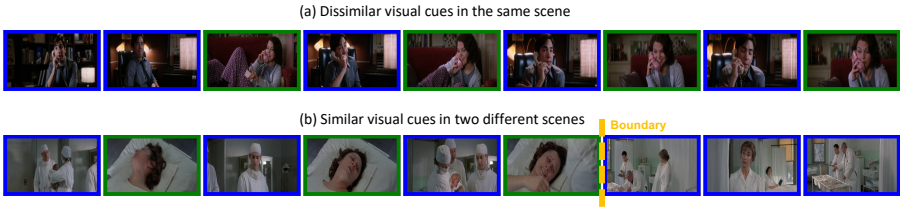

Fig. 1. Examples of the video scene segmentation. In each row, we visualize the shots including similar visual cues (*e.g.*, characters, places, etc.) with the same colored border.

segmentation task, temporally localizing scene boundaries from a long video; the term scene is widely used in filmmaking and scene (a series of semantically cohesive shots) is considered as a basic unit for understanding the story of movies.

One of the biggest challenges in video scene segmentation is that it is not achieved simply by detecting changes in visual cues. As shown in Fig. 1(a), we present an example of nine shots, all of which belong to the same scene, where two characters are talking on the phone; the overall visual cues within the scene do not stay the same but rather change repeatedly when each character appears. On the other hand, Fig. 1(b) shows two different scenes which contain visually similar shots (highlighted in blue) where the same character appears in the same place. Thus, it is expected that two adjacent scenes which share shots with similar visual cues need to be contextually discriminated. From this observation, it is important for the video scene segmentation task to model contextual relationship between shots by maximizing 1) *intra-scene similarity* (*i.e.*, the shots in the same scene should be close to each other) and 2) *inter-scene discrimination* across two adjacent scenes (*i.e.*, shots across the scene boundary should be distinguishable).

Supervised learning approaches (*e.g.*, [34]) are clearly limited due to the lack of large-scale datasets with reliable ground-truth annotations; in addition, collecting boundary annotations from long videos is extremely expensive. Recently, self-supervision [5, 9, 17, 37] is spotlighted through its effectiveness in learning in-domain representation without relying on costly ground-truth annotations. The self-supervised learning methods [11, 14, 33] in the video domain have been proposed to learn spatio-temporal patterns in a short term; inspired by this, ShotCoL [8] proposed shot-level representation pre-training algorithm based on contrastive prediction task. Although ShotCoL shows the remarkable performance, such shot-level representation learned without being aware of the semantic transition is insufficient for video scene segmentation. This is because the task requires not only a good representation for individual shots but also contextual representation considering neighboring shots at a higher level as observed in Fig. 1. Thus, we set our main goal to design effective pre-text tasks for video scene segmentation so that the model can learn the contextual relationship between shots across semantic transition during pre-training.

We introduce a novel **B**oundary-**a**ware **S**elf-**S**upervised **L**earning (BaSSL) framework where we learn boundary-aware contextualized representation effec-

tive in capturing semantic transtion during pre-training and adapt the learned representation for precise scene boundary detection through fine-tuning. The main idea during pre-training is identifying a moment with a maximum semantic transition and using it as psuedo-boundary. Then, we propose three boundary-aware pretext tasks that are beneficial to the video scene segmentation task as follows: 1) Shot-Scene Matching (SSM) matching shots with their associated scenes, 2) Contextual Group Matching (CGM) aligning shots whether they belong to the same scene or not and 3) Pseudo-boundary Prediction (PP) capturing semantic changes. SSM and CGM encourage the model to maximize intra-scene similarity and inter-scene discrimination, while PP enables the model to learn the capability of identifying transitional moments. In addition, we perform Masked Shot Modeling task inspired by CBT [46] to further learn temporal relationship between shots. The comprehensive analysis demonstrates the effectiveness of the boundary-aware pre-training compared to shot-level pre-training as well as the contribution of the individual proposed components (*i.e.*, pseudo-boundary discovery algorithm and boundary-aware pretext tasks).

Our main contributions are summarized as follows: (*i*) we introduce a novel boundary-aware pre-training framework which leverages pseudo-boundaries to learn contextual relationship between shots during the pre-training; (*ii*) we propose three boundary-aware pretext tasks, which are carefully designed to learn essential capabilities required for the video scene segmentation task; (*iii*) we perform extensive ablations to demonstrate the effectiveness of the proposed framework; (*iv*) we achieve the new state-of-the-art on the MovieNet-SSeg benchmark with large margins compared to existing methods.

2 Related Work

Video scene segmentation approaches formulate the task as a problem of temporal grouping of shots. In this formulation, the optimal grouping can be achieved by clustering-based [7, 35, 36, 40], dynamic programming-based [16, 39, 48] or multi-modal input-based [30, 43] methods. However, the aforementioned methods have been trained and evaluated on small-scale datasets such as OVSD [38] and BBC [3] which can produce a poorly generalized model. Recently, [19] introduce a large-scale video scene segmentation dataset (*i.e.*, MovieNet-SSeg) that contains hundreds of movies. Training with large-scale data, [34] proposes a strong supervised baseline model that performs a shot-level binary classification followed by grouping using the prediction scores. [8] proposes a shot contrastive pre-training method that learns shot-level representation. We found ShotCoL [8] to be the most similar to our method. However, our method is different from ShotCoL in that we focus on learning contextual representations by considering the relationship between shots through boundary-aware pre-text tasks.

Action segmentation in videos is one of the related works for video scene segmentation, which identifies action labels of individual frames, thus can divide a video into a series of action segments. Supervised methods [13, 24] proposed

CNN-based architectures to effectively capture temporal relationship between frames in order to address an over-segmentation issue. As frame-level annotations are prohibitively costly to acquire, weakly supervised methods [6, 15, 26, 27, 41, 44, 59] have been suggested to use an ordered list of actions occurring in a video as supervision. Most of the methods are trained to find (temporal) semantic alignment between frames and a given action list using an HMM-based architecture [21], a DP-based assignment algorithm [15] or a DTW-based temporal alignment method [6]. Recently, unsupervised methods [22, 23, 28, 51, 54] have been further proposed; in a nutshell, clustering-based prototypes (corresponding to one of the actions) are discovered from unlabeled videos, then the methods segment the videos by assigning prototypes into frames. Contrary to action segmentation localizing segments each of which represents a single action within an activity, video scene segmentation requires localizing more complex segments each of which may be composed of more than two actions (or activities).

Self-supervised learning in videos has been actively studied for the recent years with approaches proposing various pretext tasks such as future frame prediction [2, 45, 52], temporal ordering of frames [25, 31, 55], geometric transformations prediction [20], colorization of videos [53], multimodal correspondence [57] and contrastive prediction [11, 14, 33]. In addition, CBT [46, 47] proposes a pretext task of masked frame modeling to learn temporal dependency between frames (or clips). Note that since most of those methods are proposed for the classification task, they would be sub-optimal to the video scene segmentation task. On the other hand, BSP [56] proposes a pre-training algorithm based on pseudo-boundary synthesis for temporal localization tasks. However, the method still requires video-level class labels to synthesize pseudo-boundaries thus is not applicable to videos such as movies that are hard to define semantic labels. Also, note that we empirically show that pseudo-boundaries identified by our method are more effective for pre-training than synthesized pseudo-boundaries.

3 Preliminary

Terminologies A long video (*e.g.*, documentaries, TV episodes and movies) is assumed to have a hierarchical structure at three-level semantics: frame, shot and scene. A shot is a series of frames physically captured by the same camera during an uninterrupted period of time. A scene is a series of semantically cohesive shots and serves as a semantic unit for making a story. Note that, in this paper, our focus is on finding scene-level boundaries.

Video Scene Segmentation Task Given a long video, which contains a series of N shots $\{\mathbf{s}_1, ..., \mathbf{s}_N\}$ with class labels $\{y_1, ..., y_N\}$ where $y_i \in \{0, 1\}$ indicating if it is the last shot of a scene, the video scene segmentation task is formulated as a simple binary classification task at an individual shot level. Leveraging the local context from the neighbor shots, existing methods [8, 34] adopt a sliding window scheme. For n^{th} shot $\mathbf{s}_n$, the window is defined by $\mathbf{S}_n = \{\mathbf{s}_{n-K}, ..., \mathbf{s}_n, ..., \mathbf{s}_{n+K}\}$

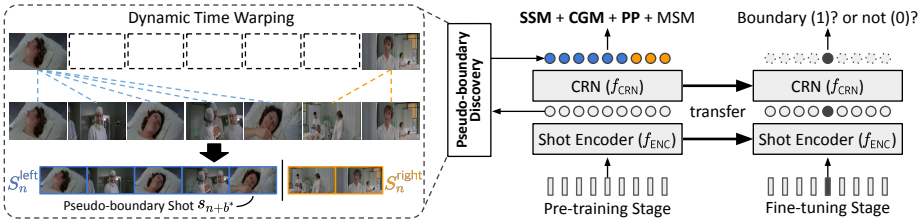

Fig. 2. Overall pipeline of our proposed framework, BaSSL.

containing a sequence of $2K+1$ shots where K is the number of neighbor shots before and after $\mathbf{s}_n$. Then, supervised learning methods typically train a parameterized (θ) model by maximizing the expected log-likelihood:

$$\theta^* = \arg\max_{\theta} \mathbb{E}\left[\log p_\theta(y_n|\mathbf{S}_n)\right]. \tag{1}$$

Note that each shot $\mathbf{s}$ is given by a set of N_k key-frames, resulting in a tensor with size of (N_k, C, H, W) where C, H and W are the RGB channels, the height and the width, respectively.

Model Architecture The model (θ) consists of two main components: 1) *shot encoder* embedding a shot by capturing its spatio-temporal patterns, and 2) *contextual relation network* (CRN) capturing contextual relation between shots. Taking a window $\mathbf{S}_n = \{\mathbf{s}_{n-K}, ..., \mathbf{s}_n, ..., \mathbf{s}_{n+K}\}$ centered at $\mathbf{s}_n$ as an input, two-level representations are extracted as follows:

$$\mathbf{e}_n = f_{\text{ENC}}(\mathbf{s}_n) \quad \text{and} \quad \mathbf{C}_n = f_{\text{CRN}}(\mathbf{E}_n), \tag{2}$$

where $f_{\text{ENC}}\colon \mathbb{R}^{N_k \times C \times H \times W} \to \mathbb{R}^{D_e}$ and $f_{\text{CRN}}\colon \mathbb{R}^{(2K+1)\times D_e} \to \mathbb{R}^{(2K+1)\times D_c}$ represent the shot encoder and CRN while D_e and D_c mean dimensions of encoded and contextualized features, respectively. $\mathbf{e}_n$ is an encoding of shot $\mathbf{s}_n$ by f_{ENC} while $\mathbf{E}_n = \{\mathbf{e}_{n-K}, ..., \mathbf{e}_n, ..., \mathbf{e}_{n+K}\}$ and $\mathbf{C}_n = \{\mathbf{c}_{n-K}, ..., \mathbf{c}_n, ..., \mathbf{c}_{n+K}\}$ correspond to the input and output feature sequence for f_{CRN}, respectively. In addition, the model employs additional pretext-specific heads for pre-training or a scene boundary detection head for fine-tuning.

Shot-level Self-supervised Learning ShotCoL [8] proposes a shot-level contrastive self-supervised learning algorithm for video scene segmentation, which learns to make representation of visually similar nearby shots—highly likely to belong to the same scene—similar. However, the method has following two limitations. First, since ShotCoL pre-trains a model without explicitly identifying semantic boundaries during pre-training, it may fail to properly maximize intra-scene similarity and inter-scene dissimilarity. For example, the visually similar shots in different scenes may be learned indistinguishable. Second, the method learns shot representation given by the shot encoder (f_{ENC}) only and does not learn temporal and contextual relation between shots given by the contextual

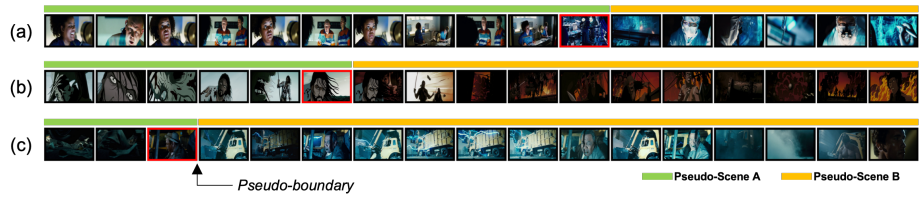

Fig. 3. An example in each row shows an input window sampled from the same scene where there exists no ground-truth scene-level boundary. Our method finds a pseudo-boundary shot (highlighted in red) that divides a sequence into two pseudo-scenes (represented by green and orange bars, respectively) so that semantics (*e.g.*, places, characters) maximally changes.

relation network (f_{CRN}). Contrary to such shot-level self-supervised learning, we propose boundary-aware self-supervised learning that trains both f_{ENC} and f_{CRN} while capturing the desired contextual relation between shots across semantic change. More detailed comparisons are given in appendix.

4 Boundary-aware Self-supervised Learning (BaSSL)

4.1 Overview

As illustrated in Fig. 2, our framework BaSSL is based on two-stage training following common practice [8]: pre-training on large-scale unlabeled data with self-supervision and fine-tuning on relatively small labeled data. Our main focus is in the pre-training stage, designing effective pretext tasks for video scene segmentation. Furthermore, we aim to train both shot encoder and contextual relation network while maximizing intra-scene similarity and inter-scene discrimination across semantic transition.

During pre-training, given an input window $\mathbf{S}_n$, BaSSL finds a shot across which the semantic transition becomes maximum and uses it as a pseudo-boundary to self-supervise the model. To be specific, we leverage the dynamic time warping technique to divide the shots in a window into two semantically disjoint sub-sequences, thus yielding a pseudo-boundary (Section 4.2). Then, we pre-train a model θ using three boundary-aware pre-text tasks and the masked shot modeling task adopted from CBT [46] to maximize intra-scene similarity and inter-scene dissimilarity (Section 4.3). After pre-trained with the four pretext tasks, the model is fine-tuned with labeled scene boundaries (Section 4.4).

4.2 Pseudo-boundary Discovery

The goal of our pre-training is to learn a capability of capturing semantic change before and after a semantic transition moment, thereby leading to higher performance in video scene segmentation. Specifically, we leverage a pseudo-boundary as a clue for self-supervision. However, extracting scene-level pseudo-boundaries

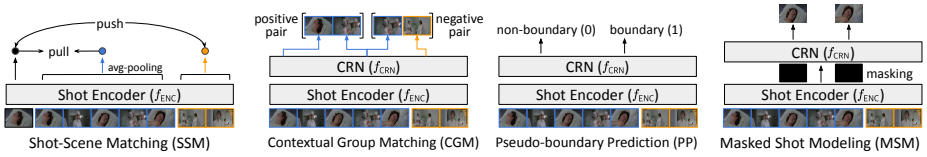

Fig. 4. Illustration of four pre-training pretext tasks.

from an input window is challenging. This is because there may be no scene boundary and it is also difficult to determine how many boundaries there are. Therefore, given an input window, we adopt a simple approach of finding a single moment where the semantics is maximally transitioning, and use it as pseudo-boundary. Although such identified moment may not correspond to scene-level boundary, it is still effective in learning a capability to capture semantic transition and the capability can be adapted to detect scene-level transition via fine-tuning. Fig. 3 shows identified pseudo-boundaries from input windows having no ground-truth scene boundary; we observe that the resulting two sub-sequences are still cognitively distinguishable. More examples are presented in appendix.

The process, dividing an input window $\mathbf{S}_n$ into two continuous, non-overlapping sub-sequences $\mathbf{S}_n^{\text{left}}$ and $\mathbf{S}_n^{\text{right}}$ with maximum semantic transition, can be seen as a temporal alignment problem between $\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$; specifically, observing the first shot should belong to $\mathbf{S}_n^{\text{left}}$ and the last one to $\mathbf{S}_n^{\text{right}}$, we define $\mathbf{S}_n^{\text{slow}} = \{\mathbf{s}_{n-K}, \mathbf{s}_{n+K}\}$, which can be seen as a same video with $\mathbf{S}_n$ with lower sampling frequency. Then, the problem becomes aligning intermediate shots either to the first shot $\mathbf{s}_{n-K}$ or the last shot $\mathbf{s}_{n+K}$ while preserving continuity.

Under the problem setting, we adopt dynamic time warping (DTW) [4] to find the optimal alignment between $\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$. DTW solves the following optimization problem using dynamic programming to maximize semantic coherence of the resulting two sub-sequences among all possible boundary candidates:

$$b^* = \underset{b=-K+1,\dots,K-1}{\arg\max} \left(\frac{1}{b+K} \sum_{i=-K+1}^{b} \text{sim}(\mathbf{e}_{n-K}, \mathbf{e}_{n+i}) + \frac{1}{K-b-1} \sum_{j=b+1}^{K-1} \text{sim}(\mathbf{e}_{n+K}, \mathbf{e}_{n+j}) \right), \tag{3}$$

where b and b^* are the candidate and optimal boundary offsets, respectively. $\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$ computes cosine similarity between two given shot encodings. Two sub-sequences are inferred as $\mathbf{S}_n^{\text{left}} = \{\mathbf{s}_{n-K}, ..., \mathbf{s}_{n+b^*}\}$ and $\mathbf{S}_n^{\text{right}} = \{\mathbf{s}_{n+b^*+1}, ..., \mathbf{s}_{n+K}\}$. $\mathbf{s}_{n+b^*}$ is the pseudo-boundary shot, which is the last shot of $\mathbf{S}_n^{\text{left}}$. The results are used for learning boundary-aware pretext tasks, which will be described Section 4.3.

Discussion on Single Pseudo-boundary One might question if identifying multiple pseudo-boundaries is more reasonable, since there may exist more than

two semantic transitions in an input window. However, we emphasize that the goal of our boundary-aware pre-training is learning a capability of capturing semantic transition, not lying on perfectly capturing all semantic transitions (or scene boundaries) at once; the capability to capture all scene-level boundaries is adapted via fine-tuning. In experiments, we verify that pre-training with one semantically strongest pseudo-boundary brings remarkable performance gain.

4.3 Pre-training Objectives

As shown in Fig. 4, we pre-train a model with three novel boundary-aware pre-text tasks—1) shot-scene matching ($\mathcal{L}_{\text{ssm}}$), 2) contextual group matching ($\mathcal{L}_{\text{cgm}}$) and 3) pseudo-boundary prediction ($\mathcal{L}_{\text{pp}}$)—and an additional one, masked shot modeling ($\mathcal{L}_{\text{msm}}$) as follows:

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{ssm}} + \mathcal{L}_{\text{cgm}} + \mathcal{L}_{\text{pp}} + \mathcal{L}_{\text{msm}}. \tag{4}$$

Shot-Scene Matching (SSM) The objective of this task is to make the representations of a shot and its associated scene similar to each other, while the representations of the shot and other scenes dissimilar. In other words, SSM encourages the model to maximize intra-scene similarity, while minimizing inter-scene similarity. Considering the splitted two sub-sequences ($\mathbf{S}_n^{\text{left}}$ and $\mathbf{S}_n^{\text{right}}$) as pseudo-scenes, we train the model using the InfoNCE loss [32]:

$$\mathcal{L}_{\text{ssm}} = \mathcal{L}_{\text{nce}}\left(h_{\text{ssm}}(\mathbf{e}_{n-K}), h_{\text{ssm}}(\mathbf{r}_n^{\text{left}})\right) + \mathcal{L}_{\text{nce}}\left(h_{\text{ssm}}(\mathbf{e}_{n+K}), h_{\text{ssm}}(\mathbf{r}_n^{\text{right}})\right), \tag{5}$$

$$\mathcal{L}_{\text{nce}}(\mathbf{e}, \mathbf{r}) = -\log \frac{e^{\text{sim}(\mathbf{e},\mathbf{r})/\tau}}{e^{\text{sim}(\mathbf{e},\mathbf{r})/\tau} + \sum_{\bar{\mathbf{e}}\in\mathcal{N}_e} e^{\text{sim}(\bar{\mathbf{e}},\mathbf{r})/\tau} + \sum_{\bar{\mathbf{r}}\in\mathcal{N}_r} e^{\text{sim}(\mathbf{e},\bar{\mathbf{r}})/\tau}}, \tag{6}$$

where h_{ssm} is a SSM head of a linear layer, τ is a temperature hyperparameter and $\mathbf{r}_n^{\text{left}}$ means a scene-level representation, which is defined by the averaged encoding of shots in the sub-sequence $\mathbf{S}_n^{\text{left}}$. $\mathcal{N}_e$ and $\mathcal{N}_r$ in Eq. (6) are constructed using other shots and pseudo-scenes in a mini-batch, respectively.

Contextual Group Matching (CGM) Since directly matching representations of shots and scenes would not be effective when the scenes are composed of visually dissimilar shots, CGM is introduced to bridge this gap. Similar to SSM, CGM is also designed to maximize intra-scene similarity and inter-scene discrimination. However, CGM measures semantic coherence of the shots rather than comparing visual cues. With CGM, the model learns to decide if the given two shots belong to the same group (*i.e.*, scene) or not. In detail, we use the center shot $\mathbf{s}_n$ in the input sequence as the anchor and construct a triplet of ($\mathbf{s}_n$, $\mathbf{s}_{\text{pos}}$, $\mathbf{s}_{\text{neg}}$). We sample each shot from $\mathbf{S}_n^{\text{left}}$ and $\mathbf{S}_n^{\text{right}}$; the one sampled within the same sub-sequence with $\mathbf{s}_n$ is used as the positive shot $\mathbf{s}_{\text{pos}}$, while the other as the negative $\mathbf{s}_{\text{neg}}$. CGM loss is defined using binary cross-entropy by

$$\mathcal{L}_{\text{cgm}} = -\log\left(h_{\text{cgm}}(\mathbf{c}_n, \mathbf{c}_{\text{pos}})\right) - \log\left(1 - h_{\text{cgm}}(\mathbf{c}_n, \mathbf{c}_{\text{neg}})\right), \tag{7}$$

where h_{cgm} is a CGM head taking two shots as input and predicting a matching score. $\mathbf{c}_n$, $\mathbf{c}_{\text{pos}}$ and $\mathbf{c}_{\text{neg}}$ are the contextualized features by f_{CRN} for the center, positive and negative shots, respectively.

Pseudo-boundary Prediction (PP) Through the above two pretext tasks, our model learns the contextual relationship between shots. In addition to these, we design an extra pretext task, PP, which is more directly related to boundary detection; PP makes the model have a capability of identifying transitional moments that semantic changes. Using the pseudo-boundary shot and one randomly sampled non-boundary shot, the PP loss is defined as a binary cross-entropy loss:

$$\mathcal{L}_{\text{pp}} = -\log\left(h_{\text{pp}}(\mathbf{c}_{n+b^*})\right) - \log\left(1 - h_{\text{pp}}(\mathbf{c}_{\bar{b}})\right), \tag{8}$$

where h_{pp} is a PP head that projects the contextualized shot representation to a probability distribution over binary class. $\mathbf{c}_{n+b^*}$ and $\mathbf{c}_{\bar{b}}$ indicate the contextualized representation from f_{CRN} for the pseudo-boundary shot $\mathbf{s}_{n+b^*}$ and randomly sampled non-boundary shot $\mathbf{s}_{\bar{b}}$, respectively.

Masked Shot Modeling (MSM) Inspired by masked frame modeling [46, 47], we adopt the MSM task whose goal is to reconstruct the representation of masked shots based on their surrounding shots. In this task, given a set of encoded shot representations, we randomly apply masking each of them with a probability of 15%. For a set $\mathcal{M}$ of masked shot offsets, we learn to regress the output on each masked shot to its encoded shot representation:

$$\mathcal{L}_{\text{msm}} = \sum_{m \in \mathcal{M}} \|\mathbf{e}_m - h_{\text{msm}}(\mathbf{c}_m)\|_2^2, \tag{9}$$

where h_{msm} is a MSM head to match the dimension of contextualized shot representation with that of encoded one. $\mathbf{e}_m$ and $\mathbf{c}_m$ denote the encoded and contextualized features by f_{ENC} and f_{CRN} for a masked shot $\mathbf{s}_m$, respectively.

4.4 Fine-tuning for Scene Boundary Detection

Recall that we formulate the video scene segmentation as a binary classification task to identify contextual transition across the scene. Different from the pre-training stage, given an input window $\mathbf{S}_n$, we employ a scene boundary detection head h_{sbd} to infer a prediction from the contextualized representation ($\mathbf{c}_n$) for the center shot $\mathbf{s}_n$. Following ShotCoL [8], we freeze the parameters of the shot encoder and then train only CRN and the scene boundary detection head using a binary cross-entropy loss with the ground truth label y_n as follows:

$$\mathcal{L}_{\text{finetune}} = -y_n \log(h_{\text{sbd}}(\mathbf{c}_n)) + (1 - y_n) \log(1 - h_{\text{sbd}}(\mathbf{c}_n)). \tag{10}$$

With a sidling window scheme, each shot is decided to be a scene boundary when its prediction score is higher than a pre-defined threshold (set to 0.5).

5 Experiment

5.1 Experimental Settings

Dataset For evaluation, we use the MovieNet-SSeg dataset [19] containing 1,100 movies with 1.6M shots. Only 318 out of 1,100 movies have scene boundary annotations, which are divided into 190, 64, and 64 movies for training, validation,

Table 1. Comparison with other algorithms. † and ‡ denote that the numbers are copied from [34] and [19], respectively. ⋆ indicates the methods exploiting additional information (*e.g.*, audio, place, cast, transcript). The best numbers are in bold.

Method	AP (↑)	mIoU (↑)	AUC-ROC (↑)	F1 (↑)
Supervised Learning				
Siamese [3]‡	35.80	39.60	-	-
MS-LSTM [19]‡⋆	46.50	46.20	-	-
LGSS [34]†⋆	47.10	48.80	-	-
Unsupervised Learning				
GraphCut [36]†	14.10	29.70		-
SCSA [7]†	14.70	30.50	-	-
DP [16]†	15.50	32.00	-	-
Story Graph [48]†	25.10	35.70	-	-
Grouping [39]‡⋆	33.60	37.20		-
BaSSL w/o fine-tuning (10 epochs)	31.55	39.36	71.67	32.55
Self-supervised Learning				
ShotCoL [8]	53.40	-	-	-
BaSSL (10 epochs)	56.26 ±0.04	49.50 ±0.11	90.27 ±0.02	45.70 ±0.24
BaSSL (40 epochs)	**57.40 ±0.08**	**50.69 ±0.45**	**90.54 ±0.03**	**47.02 ±0.87**

and test split, respectively. Following ShotCoL [8], we use the entire 1,100 movies with no ground truth labels for the pre-training and fine-tune the model on the training split. The performance is measured on the test split.

Metric Following [19], we compare algorithms using AP and mIoU. Also, we adopt F1 score [1] and AUC-ROC as additional evaluation metrics. We also report Meta-Sum metric inspired by [10, 29] for easy and straightforward comparison.

Implementation details We employ ResNet-50 [18] and Transformer [50] as shot encoder and CRN, respectively. We cross-validate the number of neighbor shots among $K = \{4, 8, 12, 16\}$ and $K = 8$ is selected due to its good performance and computational efficiency. In all experiments, we report mean and std from 5 fine-tuned models with random seeds. More details are presented in appendix.

5.2 Comparison with State-of-the-art Methods

We compare BaSSL with 1) supervised ones: LGSS [34], Siamese [3], MS-LSTM [19] and, 2) unsupervised ones: GraphCut [36], SCSA [7], DP [16], StoryGraph [48] and Grouping [39], and 3) self-supervised ones: ShotCoL [8]. Without fine-tuning, BaSSL can be seen as an unsupervised model in that it is trained to predict the pseudo-boundary by the PP task. Table 1 summarizes comparison against competing methods. BaSSL without fine-tuning shows competitive or outperforming performance based only on the basic visual cue compared to competing unsupervised ones. Furthermore, fine-tuning BaSSL with ground-truth scene boundaries

[1] Contrary to the previous works [34, 8] that report recall, we use F1 score to consider for balanced comparison between precision and recall.

Table 2. Average precision (AP) comparison with pre-training baselines. Note that SimCLR (NN) corresponds to our reproduced ShotCoL using SimCLR.

Method	Pre-training		Transfer		Architecture of f_{CRN} during fine-tuning		
	f_{ENC}	f_{CRN}	f_{ENC}	f_{CRN}	MLP	MS-LSTM	Transformer
Supervised pre-training using image dataset							
M1 ImageNet	✓		✓		43.12 ±0.14	45.10 ±0.55	47.13 ±1.04
M2 Places365	✓		✓		43.82 ±0.10	45.87 ±0.40	48.71 ±0.50
Shot-level pre-training							
M3 SimCLR (instance)	✓		✓		45.60 ±0.07	49.09 ±0.24	51.51 ±0.31
M4 SimCLR (temporal)	✓		✓		45.55 ±0.11	49.24 ±0.26	50.05 ±0.78
M5 SimCLR (NN)	✓		✓		45.99 ±0.13	50.73 ±0.19	51.17 ±0.69
Boundary-aware pre-training							
M6 BaSSL	✓	✓	✓		46.53 ±0.11	50.58 ±0.14	50.82 ±0.69
M7 BaSSL	✓	✓	✓	✓	-	-	56.26 ±0.04
M8 M5+M7	✓	✓	✓	✓	-	-	56.86 ±0.01

improves AP by 24.71%p and BaSSL outperforms all other algorithms. Finally, through longer pre-training (40 epochs), BaSSL surpasses the previous state-of-the-art method (*i.e.*, ShotCoL) by a large margin (4.00%p in AP).

5.3 Comparison with Pre-training Baselines

We perform extensive experiments to compare BaSSL with the pre-training baselines learning shot-level representation by f_{ENC} only. In the experiments, we compare the following three types of pre-training approaches. The first group (M1-2) trains f_{ENC} using image-level supervision on ImageNet [12] or place labels on Places365 [58]. The second group (M3-5) trains f_{ENC} through shot-level contrastive learning (*i.e.*, SimCLR [9]) with different positive pair sampling strategies. Specifically, *instance* (M3) takes an instance of the center shot with different augmentation, *temporal* (M4) takes one randomly sampled neighbor shot as positive pair in local temporal window, and *NN* (M5) takes the most visually similar shot among the neighbor shots as positive pair, which is also known as ShotCoL [8]. The last group (M6-8) learns both f_{ENC} and f_{CRN} through boundary-aware pretext tasks proposed in this paper. Given pre-trained representations of f_{ENC}, we train a video scene segmentation model with three different types of f_{CRN} including MLP [8], MS-LSTM [19][2] and Transformer. For fair comparison, all pre-training methods employ ResNet-50 as f_{ENC} and we pre-train the models for 10 epochs.

In Table 2, we found the following observations. First, when transferring pre-trained shot representation, employing MS-LSTM and Transformer as f_{CRN} is more effective than using MLP, as they are favorably designed to capture contextual relation between shots (see M1-6). Second, BaSSL (M7) outperforms all competing baselines (M1-5), which shows the importance of boundary-aware pre-training. Third, it turns out that transferring the representation through f_{CRN}

[2] https://github.com/AnyiRao/SceneSeg/tree/master/lgss

Table 3. Ablation study on varying combinations of pretext tasks for pre-training. The best scores are highlighted in bold.

	Pretext Tasks				Evaluation Metric				
	SSM	CGM	PP	MSM	AP	mIoU	AUC-ROC	F1	Sum
P1	✓				42.57 ±0.29	40.12 ±0.50	84.11 ±0.15	30.83 ±0.79	197.63
P2		✓			36.76 ±0.02	40.59 ±0.18	82.06 ±0.04	30.94 ±0.32	190.35
P3			✓		36.55 ±0.04	39.58 ±0.05	81.36 ±0.03	29.96 ±0.04	187.45
P4				✓	13.33 ±0.23	29.80 ±0.39	64.65 ±0.98	18.68 ±0.39	126.45
P5	✓	✓			55.77 ±0.05	48.19 ±0.21	90.19 ±0.03	43.17 ±0.39	237.32
P6	✓		✓		56.04 ±0.08	49.00 ±0.16	90.13 ±0.02	44.74 ±0.29	239.91
P7		✓	✓		38.09 ±0.03	41.25 ±0.10	82.85 ±0.01	32.24 ±0.24	195.43
P8	✓			✓	54.39 ±0.07	47.54 ±0.18	89.72 ±0.03	42.48 ±0.22	234.13
P9		✓		✓	39.49 ±0.04	41.71 ±0.12	83.27 ±0.02	32.85 ±0.20	197.32
P10			✓	✓	38.53 ±0.07	40.85 ±0.15	82.78 ±0.04	31.47 ±0.16	193.63
P11		✓	✓	✓	41.02 ±0.07	40.89 ±0.10	83.79 ±0.02	31.53 ±0.18	197.23
P12	✓		✓	✓	56.10 ±0.08	49.10 ±0.17	90.09 ±0.03	45.42 ±0.30	240.71
P13	✓	✓		✓	56.20 ±0.06	48.00 ±0.17	90.13 ±0.01	43.24 ±0.27	237.57
P14	✓	✓	✓		**56.26 ±0.02**	48.42 ±0.33	90.25 ±0.01	43.98 ±0.58	238.91
P15	✓	✓	✓	✓	**56.26 ±0.04**	**49.50 ±0.11**	**90.27 ±0.02**	**45.70 ±0.24**	**241.73**

is important for the boundary detection task where it leads to a performance gain of 5.44%p in AP (see M6-7). Finally, learning shot-level and contextual representations is complementary to each other; that is, incorporating ShotCoL (M5) and our framework (M7) provides further improved performance (M8).

5.4 Ablation Studies

Impact of individual pretext tasks We investigate the contribution of individual pretext tasks. In this experiment, we train models by varying the combinations of the pretext tasks. From Table 3, we can obtain following two observations. First, among models trained by a single pretext task (P1-4), the MSM leads to the worst performance compared to the others. This indicates that boundary-aware pretext tasks (*i.e.*, SSM, CGM and PP) are indeed important for scene boundary detection. Second, the more pretext tasks are used, the better the performance is, and the best one is obtained from all tasks (P15). This means all tasks are complementary to each other, contributing to improvement.

Psuedo-boundary discovery method To check the effectiveness of DTW-based pseudo-boundary discovery, we train three models with different pseudo-boundary decision strategies—1) *Random* defining one randomly sampled shot in the input window as a pseudo-boundary, 2) *Fixed* always taking the center shot as a pseudo-boundary, and 3) *Synthesized*, inspired by BSP [56], synthesizing the input window by concatenating two sub-sequences sampled from different movies and using the last shot of the first sub-sequence as a pseudo-boundary. Table 4(a) summarizes the results. Our approach to adopting DTW to find pseudo-boundaries achieves the best performance. It is notable that BaSSL with

Table 4. Ablations to check the impact of pseudo-boundary discovery strategies, the number of neighboring shots (K) and longer pre-training. The best scores are in bold.

Pseudo-boundary	AP
Random	46.64 ±0.37
Fixed	49.53 ±0.32
Synthesized	54.61 ±0.03
DTW (ours)	**56.26 ±0.04**

(a) Pseudo-boundary discovery methods.

# Neighbors	AP
4	55.98 ±0.10
8	56.26 ±0.04
12	**56.29 ±0.03**
16	55.31 ±0.04

(b) The number of neighbor shots.

Epochs	AP
10	56.26 ±0.04
20	56.74 ±0.04
30	56.74 ±0.07
40	**57.40 ±0.08**
50	57.15 ±0.08

(c) The number of pre-training epochs.

Table 5. Scene clustering quality measured by normalized mutual information (NMI).

Model	Scene Length			$\Delta \downarrow$ (Short $\rightarrow$ Long)
	Short (N_c=8)	Medium (N_c=16)	Long (N_c=32)	
ImageNet	67.50	61.60	56.25	-16.67%
SimCLR (temporal)	82.40	81.65	78.99	-4.14%
SimCLR (NN)	83.54	83.17	81.25	-2.75%
BaSSL (ours)	**86.22**	**86.72**	**85.63**	**-0.68%**

Synthesized pseudo-boundaries also outperforms the pre-training baselines in Table 2, which shows the importance of boundary-aware pre-training.

Hyperparameters We analyze the impact of two hyperparameters: the number of neighbor shots K and pre-training epochs. Table 4(b) shows that we achieve higher performance with more neighbor shots, saturating around $K = 12$. Table 4(c) shows the impact of longer pre-training. We find that performance increases until certain numbers (40 epochs) and decrease afterward. We conjecture that this is partly due to overfitting to noise from incorrect pseudo-boundaries.

5.5 Analysis on Pre-trained Shot Representation Quality

We analyze the quality of pre-trained shot representations using normalized mutual information (NMI) to measure the clustering quality. Specifically, we randomly sample 100 scenes from the test split of MovieNet-SSeg while we vary the length of scenes $N_c \in \{8, 16, 32\}$ (the number of shots included in a single scene). Then, we perform K-Means clustering on $N_c \times 100$ shot representations extracted by the pre-trained model with the number of classes (=100). This intends to form a single cluster for each scene, assuming that high-quality representation would locate the shot embeddings within the same scene close to each other. Considering the randomness in the K-Means clustering and scene sampling, we report the averaged score from 5 trials.

Table 5 shows the NMI score for different pre-trained models. BaSSL outperforms the shot-level pre-training baselines and the model pre-trained using ImageNet. With respect to different scene lengths (N_c), we found BaSSL is more

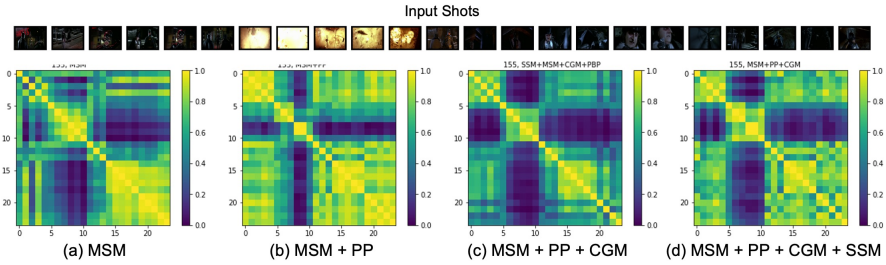

Fig. 5. Visualization of similarity (below) between shot representations in randomly sampled consecutive shots (above). We observe that the shot representations are clearly clustered as adding pretext tasks one by one.

robust than the others. When the visual diversity across the shots increases as the scenes become longer (N_c=8 $\rightarrow$ 32), the performance of BaSSL drops only -0.68% while the other baselines suffer from severe degradation. This implies the effectiveness of BaSSL in maximizing intra-scene similarity.

5.6 Qualitative Analysis

To qualitatively check the effect of individual pretext tasks, we visualize the matrix of cosine similarity between shot representations from the randomly sampled 16 consecutive shots in Fig. 5. The shot representations are computed by models without the fine-tuning in order to solely focus on the behavior of each pretext task at the pretraining stage. When the MSM is used only, approximately three clusterings are shown, but similarity around boundaries is smoothed. Next, when we add PP, dissimilarities around the boundaries are to be sharpened. Then, with additional CGM, the clusters are more clearly obtained. Finally, adding SSM makes the similarity of shots within the same cluster higher (*i.e.*, more yellow ones). On the other hand, we present more qualitative analysis for discovered pseudo-boundaries and scene boundary predictions in supplementary material.

6 Conclusion

We present BaSSL, a novel self-supervised framework for video scene segmentation, especially designed to learn contextual relationship between shots. Through the pseudo-boundary discovery, we can define and conduct boundary-aware pretext tasks that encourage the model to learn the contextual relational representation and a capability of capturing transitional moments. Comprehensive experiments demonstrate the effectiveness of our framework and we achieve outstanding performance in the MovieNet-SSeg dataset.

Acknowledgements. This work was supported by Kakao Brain and partly by Korea research grant from NRF (2021H1D3A2A03038607/10%, 2022R1C1C1010627/10%), and IITP (2021-0-01778/10%, 2022-0-00264/40%, 2022-0-00951/10%, 2022-0-00612/10%, 2020-0-01373/10%).

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: YouTube-8M: A large-scale video classification benchmark. arXiv:1609.08675 (2016)
2. Ahsan, U., Sun, C., Essa, I.: DiscrimNet: Semi-Supervised Action Recognition from Videos using Generative Adversarial Networks. arXiv:1801.07230 (2018)
3. Baraldi, L., Grana, C., Cucchiara, R.: A Deep Siamese Network for Scene Detection in Broadcast Videos. In: ACM MM (2015)
4. Berndt, D.J., Clifford, J.: Using Dynamic Time Warping to Find Patterns in Time Series. In: SIGKDD workshop (1994)
5. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. arXiv:2006.09882 (2020)
6. Chang, C.Y., Huang, D.A., Sui, Y., Fei-Fei, L., Niebles, J.C.: D3TW: Discriminative Differentiable Dynamic Time Warping for Weakly Supervised Action Alignment and Segmentation. In: CVPR (2019)
7. Chasanis, V.T., Likas, A.C., Galatsanos, N.P.: Scene Detection in Videos using Shot Clustering and Sequence Alignment. IEEE transactions on multimedia **11**(1), 89–100 (2008)
8. Chen, S., Nie, X., Fan, D., Zhang, D., Bhat, V., Hamid, R.: Shot Contrastive Self-Supervised Learning for Scene Boundary Detection. In: CVPR (2021)
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A Simple Framework for Contrastive Learning of Visual Representations. In: ICML (2020)
10. Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: UNITER: UNiversal Image-TExt Representation Learning. In: ECCV (2020)
11. Dave, I., Gupta, R., Rizve, M.N., Shah, M.: TCLR: Temporal Contrastive Learning for Video Representation. arXiv:2101.07974 (2021)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-scale Hierarchical Image Database. In: CVPR (2009)
13. Farha, Y.A., Gall, J.: MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation. In: CVPR (2019)
14. Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., He, K.: A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In: CVPR (2021)
15. Fried, D., Alayrac, J.B., Blunsom, P., Dyer, C., Clark, S., Nematzadeh, A.: Learning to Segment Actions from Observation and Narration. arXiv:2005.03684 (2020)
16. Han, B., Wu, W.: Video Scene Segmentation using A Novel Boundary Evaluation Criterion and Dynamic Programming. In: IEEE International conference on multimedia and expo (2011)
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum Contrast for Unsupervised Visual Representation Learning. In: CVPR (2020)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016)
19. Huang, Q., Xiong, Y., Rao, A., Wang, J., Lin, D.: MovieNet: A Holistic Dataset for Movie Understanding. In: ECCV (2020)
20. Jing, L., Tian, Y.: Self-Supervised Spatiotemporal Feature Learning by Video Geometric Transformations. arXiv preprint arXiv:1811.11387 (2018)
21. Kuehne, H., Richard, A., Gall, J.: A hybrid RNN-HMM Approach for Weakly Supervised Temporal Action Segmentation. IEEE transactions on pattern analysis and machine intelligence **42**(4), 765–779 (2018)

22. Kukleva, A., Kuehne, H., Sener, F., Gall, J.: Unsupervised Learning of Action Classes with Continuous Temporal Embedding. In: CVPR (2019)
23. Kumar, S., Haresh, S., Ahmed, A., Konin, A., Zia, M.Z., Tran, Q.H.: Unsupervised Activity Segmentation by Joint Representation Learning and Online Clustering. arXiv:2105.13353 (2021)
24. Lea, C., Reiter, A., Vidal, R., Hager, G.D.: Segmental Spatiotemporal CNNs for Fine-Grained Action Segmentation. In: ECCV (2016)
25. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised Representation Learning by Sorting Sequences. In: ICCV (2017)
26. Li, J., Lei, P., Todorovic, S.: Weakly Supervised Energy-based Learning for Action Segmentation. In: ICCV (2019)
27. Li, J., Todorovic, S.: Set-Constrained Viterbi for Set-Supervised Action Segmentation. In: CVPR (2020)
28. Li, J., Todorovic, S.: Action Shuffle Alternating Learning for Unsupervised Action Segmentation. In: CVPR (2021)
29. Li, L., Lei, J., Gan, Z., Yu, L., Chen, Y.C., Pillai, R., Cheng, Y., Zhou, L., Wang, X.E., Wang, W.Y., et al.: VALUE: A Multi-Task Benchmark for Video-and-Language Understanding Evaluation. In: NeurIPS (2021)
30. Liang, C., Zhang, Y., Cheng, J., Xu, C., Lu, H.: A Novel Role-Based Movie Scene Segmentation Method. In: Pacific-Rim Conference on Multimedia (2009)
31. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and Learn: Unsupervised learning using temporal order verification. In: ECCV (2016)
32. Oord, A.v.d., Li, Y., Vinyals, O.: Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748 (2018)
33. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal Contrastive Video Representation Learning. In: CVPR (2021)
34. Rao, A., Xu, L., Xiong, Y., Xu, G., Huang, Q., Zhou, B., Lin, D.: A Local-to-Global Approach to Multi-modal Movie Scene Segmentation. In: CVPR (2020)
35. Rasheed, Z., Shah, M.: Scene Detection in Hollywood Movies and TV Shows. In: CVPR (2003)
36. Rasheed, Z., Shah, M.: Detection and Representation of Scenes in Videos. IEEE transactions on Multimedia **7**(6), 1097–1105 (2005)
37. Roh, B., Shin, W., Kim, I., Kim, S.: Spatially Consistent Representation Learning. In: CVPR (2021)
38. Rotman, D., Porat, D., Ashour, G.: Robust and efficient video scene detection using optimal sequential grouping. In: IEEE international symposium on multimedia (ISM) (2016)
39. Rotman, D., Porat, D., Ashour, G.: Optimal Sequential Grouping for Robust Video Scene Detection using Multiple Modalities. International Journal of Semantic Computing **11**(02), 193–208 (2017)
40. Rui, Y., Huang, T.S., Mehrotra, S.: Exploring Video Structure beyond The Shots. In: Proc. of the IEEE International Conference on Multimedia Computing and Systems (1998)
41. Shen, Y., Wang, L., Elhamifar, E.: Learning To Segment Actions From Visual and Language Instructions via Differentiable Weak Sequence Alignment. In: CVPR (2021)
42. Shou, M.Z., Lei, S.W., Wang, W., Ghadiyaram, D., Feiszli, M.: Generic event boundary detection: A benchmark for event segmentation. In: ICCV (2021)

43. Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., Trancoso, I.: Temporal Video Segmentation to Scenes using High-level Audiovisual Features. IEEE Transactions on Circuits and Systems for Video Technology **21**(8), 1163–1177 (2011)
44. Souri, Y., Fayyaz, M., Minciullo, L., Francesca, G., Gall, J.: Fast Weakly Supervised Action Segmentation using Mutual Consistency. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
45. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using LSTMs. In: ICML (2015)
46. Sun, C., Baradel, F., Murphy, K., Schmid, C.: Learning Video Representations using Contrastive Bidirectional Transformer. arXiv:1906.05743 (2019)
47. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: VideoBERT: A Joint Model for Video and Language Representation Learning. In: ICCV (2019)
48. Tapaswi, M., Bauml, M., Stiefelhagen, R.: StoryGraphs: Visualizing Character Interactions as a Timeline. In: CVPR (2014)
49. Tversky, B., Zacks, J.M.: Event perception. Oxford handbook of cognitive psychology (2013)
50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention Is All You Need. In: NIPS (2017)
51. VidalMata, R.G., Scheirer, W.J., Kukleva, A., Cox, D., Kuehne, H.: Joint Visual-Temporal Embedding for Unsupervised Learning of Actions in Untrimmed Sequences. In: WACV (2021)
52. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating Videos with Scene Dynamics. NIPS (2016)
53. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking Emerges by Colorizing Videos. In: ECCV (2018)
54. Wang, Z., Chen, H., Li, X., Liu, C., Xiong, Y., Tighe, J., Fowlkes, C.: Unsupervised Action Segmentation with Self-supervised Feature Learning and Co-occurrence Parsing. arXiv:2105.14158 (2021)
55. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In: CVPR (2019)
56. Xu, M., Pérez-Rúa, J.M., Escorcia, V., Martinez, B., Zhu, X., Zhang, L., Ghanem, B., Xiang, T.: Boundary-sensitive Pre-training for Temporal Localization in Videos. arXiv:2011.10830 (2020)
57. Zhang, B., Hu, H., Lee, J., Zhao, M., Chammas, S., Jain, V., Ie, E., Sha, F.: A hierarchical multi-modal encoder for moment localization in video corpus. arXiv:2011.09046 (2020)
58. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million Image Database for Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
59. Zhukov, D., Alayrac, J.B., Cinbis, R.G., Fouhey, D., Laptev, I., Sivic, J.: Cross-Task Weakly Supervised Learning from Instructional Videos. In: CVPR (2019)