IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 December 2023, accepted 20 February 2024, date of publication 26 February 2024, date of current version 5 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3370676

RESEARCH ARTICLE

Contrasting Multi-Modal Similarity Framework for Video Scene Segmentation

JINWOO PARK, (Student Member, IEEE), JUNGEUN KIM, (Graduate Student Member, IEEE), JAEGWANG SEOK, SUKHYUN LEE, AND JUNYEONG KIM, (Member, IEEE)
Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, Republic of Korea

Corresponding author: Junyeong Kim (junyeongkim@cau.ac.kr)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by Korean Government (MSIT) (Artificial Intelligence Graduate School Program, Chung-Ang University) under Grant 2021-0-01341; and in part by IITP grant funded by Korean Government (MSIT) (Development of Previsional Intelligence Based on Long-Term Visual Memory Network) under Grant 2020-0-00004.

ABSTRACT This paper proposes a video scene segmentation framework referred to as a Contrasting Multi-Modal Similarity (CMS). Video is composed of multiple scenes which are short stories or semantic units of video, with each scene consisting of multiple shots. The task of video scene segmentation aims to semantically segment long videos, such as movies, into the sequence of scenes by identifying the boundaries of each scene transition. Current video scene segmentation frameworks have primarily relied on comparing only the visual cues of adjacent shots to identify scene boundaries. These frameworks have focused on two major approaches: 1) comparing only the visual cues of adjacent frames to distinguish between scenes and 2) performing clustering based on visual cues for distinction among scenes. However, within videos, there exist numerous scenes that are difficult to distinguish using visual information alone, as they often appear similar or ambiguous. Taking inspiration from the aforementioned issues, we propose a framework referred to as CMS that leverages not only visual cues (i.e., shots) but also textual cues (i.e., captions) to semantically distinguish scenes. The new framework, CMS, leverages visual cues and text cues as follows: 1) Generate captions corresponding to each shot using a zero-shot captioning model (Caption Generation). 2) Construct similarity score matrices for each modality to measure semantic similarities (Similarity Score Calculation). 3) Based on the above matrix, select similar shots and dissimilar shots for contrastive training (Similarity Score-based Sampling). Our experiments show that the CMS framework advances the performance to exceed the previous state-of-the-art methods with a relatively simple approach without complex model architectures.

INDEX TERMS Visual scene segmentation, multi-modal reasoning, contrastive learning.

I. INTRODUCTION

With the increasing abundance of video data, the ability to comprehend videos effectively has become exceedingly important. Thus far, notable progress has been made toward understanding videos that include video action detection/segmentation [1], [2], [3], video question answering [4], [5], video-grounded dialogue [6], [7], video moment retrieval [8] and video scene segmentation [9], [10], [11], [12], [13], [14], [15]. Among those, we focus on video scene segmentation (VSS), which plays a crucial role in understanding and interpreting long-term videos and can serve as the fundamental building block for AI systems designed to comprehend lengthy videos. The task of VSS aims to semantically segment long videos, such as movies, into the sequence of scenes by identifying the boundaries of each scene transition.

Videos can be hierarchically divided into scenes, shots, and frames. Depending on the criteria for division, frames can be merged into shots and scenes [10], [12]. A "shot" is a sequence of frames captured with a single camera movement, without any interruption. A "scene" can be viewed as a single

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

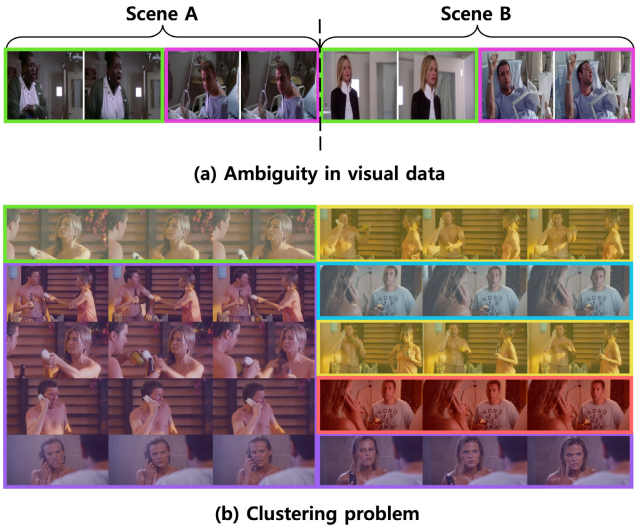

FIGURE 1. Limitation of visual data. (a) shows the ambiguity of visual data. Comparing the sections highlighted in green and pink, it can be observed that although they are visually different, they belong to the same scene. Conversely, sections of the same color may appear visually similar but belong to different scenes. (b) illustrates the shortcomings of clustering. The shots in (b) are all consecutive shots. However, when looking at the results of clustering (each color represents a cluster group), we can see that even within what appears to be the same scene, multiple clusters are formed.

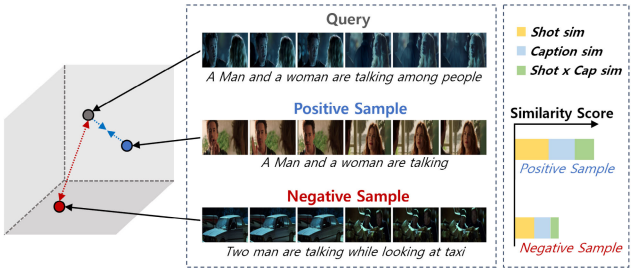

FIGURE 2. Sampling based on data similarity. Based on shot, caption, and shot x caption data, samples with a high similarity score to the query are selected as positive samples, while those with a low score are selected as negative samples. In the embedding space, the distance between the query and positive samples is trained to be closer, while the distance between the query and negative samples is trained to be farther apart.

short story or meaningful unit formed by the accumulation of these shots. While short-term videos may consist of only a few shots or a few scenes, long-term videos, such as those lasting over an hour, are composed of multiple scenes. Therefore, in order to understand lengthy videos effectively, it is important to grasp not just the overall content of the video but also its separation of meaningful units.

VSS is the task of identifying the transitions between scenes within the video, which allows for understanding the short stories within the video and, subsequently, contributes to comprehending the entire video. However, unlike humans, teaching AI agents to understand semantically separable scenes and learn their transition boundaries is a challenging task. This difficulty can be attributed to two main aspects. First, generating video scene boundary labels requires significant human and time costs. To perform video scene segmentation, it is necessary to label the shots within the video, indicating whether they are part of a scene boundary or not. But, labeling for long-term videos composed of thousands of shots can be costly in terms of human resources and time. Second, the boundaries between scenes can be ambiguous in cases where scenes are formed by the combination of multiple shots. For instance, as can be seen in Fig. 1 (a), if the background, context, or situation in which a character appears are different, it can be easily classified as different scenes even for the same scene. Conversely, adjacent shots in different scenes with similar backgrounds or visual appearances can be categorized as the same scene.

In order to overcome the aforementioned limitations and perform video scene segmentation, recent studies has heavily focused on utilizing self-supervised learning [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. Among self-supervised learning approaches, contrastive learning [12], [13], [15] is utilized to distinguish between positive and negative samples (i.e., shots) and has demonstrated promising results. These studies classify unlabeled shots into positive samples that contain similar visual cues, and negative samples that contain different visual cues using the adjacent shot [13], clustering method [15], and pseudo-boundary decision [14]. Contrastive loss is posed to make representations of positive samples to be close and that of negative samples to be far away. While every existing method on VSS relied only on visual cues, due to the ambiguity of visual cues (Fig. 1 (a)), understanding the inherent meaning of each shot solely through visual data is challenging. As another example, as can be seen in Fig. 1 (b), there are cases in which they are classified into different clusters even though they are shots that should be classified into the same cluster. As illustrated in the example, dividing video into semantically similar units solely based on visual data without providing direct scene information is a challenging task.

In this paper, we propose a novel framework called CMS(Contrasting Multi-modal Similarity Framework) for contrastive learning utilizing both visual cues and textual cues to address the aforementioned limitations and extract representations that can be used to meaningfully distinguish scenes. Reference [26] have conducted contrast learning using text data such as the genre of video, but we use a generated caption that contains information on the specific situation of the scene to focus on distinguishing the meaningful part of the scene. The proposed framework is divided into two stages: (1) the video representation learning stage and (2) the video scene segmentation stage. In the video representation learning stage, we propose a new method of classifying positive and negative samples using both visual cues and textual cues for effective contrastive learning. We first generate captions corresponding to each shot using a zero-shot captioning model (Caption Generation). Then we construct similarity score matrices for each modality to measure semantic similarities (Similarity Score Calculation). We obtain similarity matrices for video, caption, and mixture

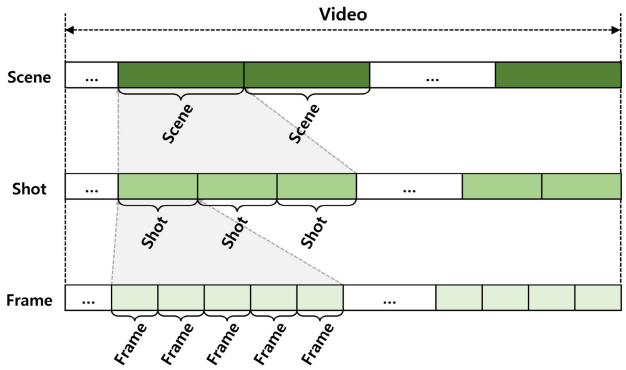

FIGURE 3. Overview of video component. A video is composed of frames. A shot consists of multiple frames, and a scene is composed of multiple shots.

of video and caption. Finally, based on the similarity matrices, we select similar shots and dissimilar shots for contrastive training (Similarity Score-based Sampling). Selected positive and negative samples are utilized to calculate contrastive loss. The overall flow of our method is shown in Fig 2. In the video scene segmentation stage, feature representations of video are extracted through the encoder learned in the previous step, and the classifier is fine-tuned using scene labels which indicates whether each adjacent shot is a scene boundary or not.

The contributions of our paper are as follows. 1) We identified the issues when utilizing visual data only for video scene segmentation. 2) To solve the limitations of visual data, we propose a method of constructing a similarity score matrix using visual cues and textual cues. In addition to this, we construct a new framework, CMS, for contrastive learning by selecting positive and negative samples based on a similarity score matrix. 3) We demonstrated superior performance on public benchmark compared to state-of-the-art methods. 4) We discovered the limitations of the learning method through task-specific sampling in the representation learning stage.

The rest of the paper is structured as follows. Section II describes studies with regard to the video scene segmentation task. Section III elaborates on the motivations for using captions and the newly proposed CMS framework. Section IV describes the dataset, the experimental setting used in the experiment, the performance comparison with existing methods, and limitations of the proposed method. Section VI concludes the paper with future work.

II. RELATED WORK

A. COMPONENT OF VIDEO

In this subsection, we define the word 'video' used in this paper and the terminology related to the video (e.g., shot, scene) to avoid confusion. There are many ways to define a video, but from a technical point of view, a video is a sequence of image frames taken in a short moment (1 second) and played back. The unit of video (i.e., fps) is more intuitive, where fps represents how many images were taken per second (e.g., 1 fps represents 1 image taken per second). Based on this perspective, the term 'video' used in this paper is defined as data in the form of multiple images listed.

Video can be divided into three units: frame, shot, and scene as shown in Fig.3. A frame is the smallest unit of video as an image. A shot is a series of frames taken by the camera at once without any interruption [15]. A scene is a unit of semantic meaning that consists of several shots and constitutes one short story in the video. A short video consists of a small amount of shots and scenes, while a long video such as a movie consists of a large amount of shots and scenes. It is important to understand the overall story to understand a long video, but it is also important to understand the short story that constitutes the video, that is the scene.

B. VIDEO SCENE SEGMENTATION WITH SELF-SUPERVISED LEARNING

Labeling work on large amounts of data, such as images and videos, has always been accompanied by human, time, and cost problems, which have led to a lack of labeled data. In order to overcome the lack of labeled data, studies using self-supervised learning to extract meaningful information from unlabeled data were conducted, and these studies showed good performance beyond the performance of supervised learning. Self-supervised learning proceeds by defining a pretext task to extract features from unlabeled data. Looking at the pretext tasks employed in previous research, they utilize techniques such as image rotation, inpainting, colorizing, jigsaw puzzles, pseudo-boundary determination, and more to extract meaningful information from unlabeled data. Recently, many studies have been conducted using contrastive learning through a pretext task to obtain contrast similarity between data by applying augmentation to original image data.

Video scene segmentation (VSS) is a task that finds the boundary at which the scene is converted. Therefore, for VSS, labels [0: not a boundary where the scene is converted, and 1: a boundary where the scene is converted] are required to represent the boundary where the scene is converted for each shot. However, labeling every frame for VSS entails significant human and time costs. Accordingly, many studies have been conducted to solve the problem of unlabeled data in video by using self-supervised learning [13], [14], [15]. The learning strategy of self-supervised learning is carried out by applying self-supervised learning to unlabeled video data to learn the overall feature of the video and then fine-tune it through labeled video data. Recently, among self-supervised learning, many studies have been conducted using contrasting learning. These studies learn the representation of the video by selecting query, positive samples, and negative samples from the shots in the video and calculating the contrastive loss between them. Methods of selecting positive samples vary, such as using adjacent shots as positive samples [13] or selecting positive samples within a cluster formed through the clustering algorithm [15]. All of these methods use only

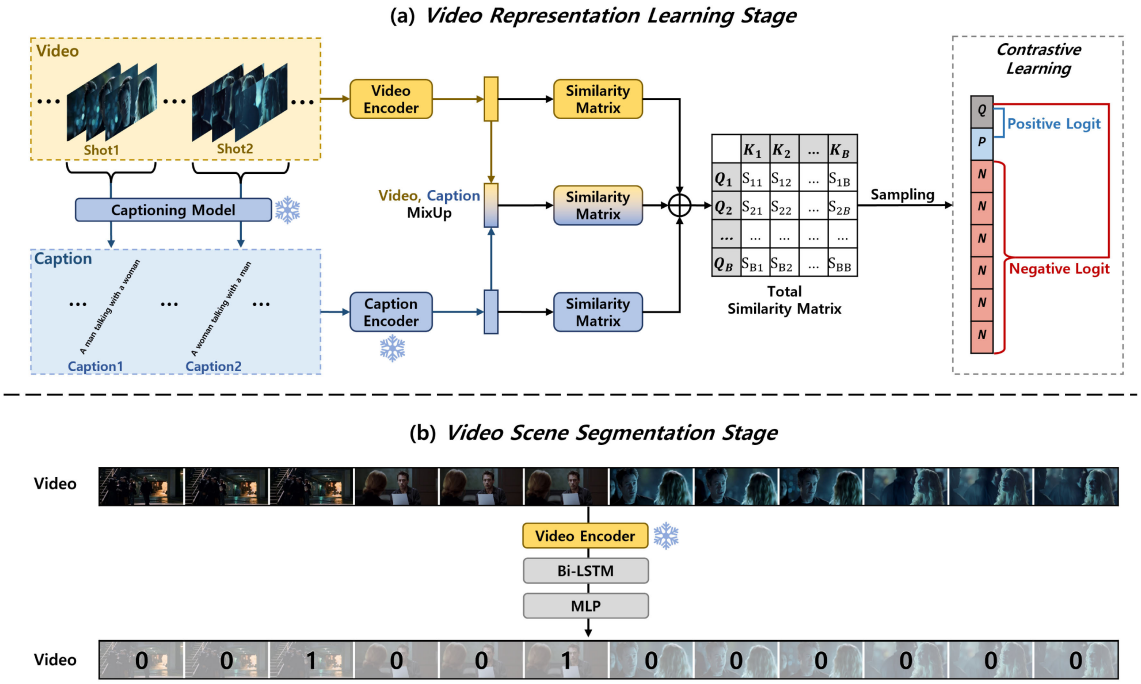

FIGURE 4. Overview of our model for video scene segmentation. There are two stages in our model. (a) is the stage of video representation learning, where the video encoder is trained through contrastive learning. It takes video and caption data as input, constructs similarity matrices for each modality, and combines them to form a total similarity matrix. Sampling is performed based on the total similarity matrix to select positive and negative samples for contrastive learning. (b) represents the video scene segmentation stage, where the encoder trained in (a) is used to extract features from the video. These features are then used to determine whether each shot is a scene boundary(1) or not(0). In the Total Similarity Matrix, Q, K, and S represent Query, Key(=Sample), and Score, respectively.

visual data. However, there is a limit to distinguishing scenes using only visual data. We find problems when using visual data only and present ways to use visual data and text data together to solve this problem.

C. MULTI-MODAL LEARNING

When you get information about an object, you can get more information from multiple perspectives than from one perspective. From this point of view, there is a study about learning video presentation by using text modality in addition to visual modality [26]. Reference [26] proposes contrastive learning method using text data such as the genre of the movie to extract positive samples within the same genre. However, text data such as genre is not suitable as text data to distinguish scenes in the video because it deals with comprehensive information about the whole. We introduce a method of utilizing caption data that can express video scenes well among text data.

III. METHOD

In this section, we introduce a model called CMS(Contrasting Multi-modal Similarity) that utilizes unlabeled video data and caption data to extract semantically meaningful features for scene segmentation. As shown in Fig.4, the model's architecture is divided into two stages: the video representation learning stage for extracting video features and the video scene segmentation stage for fine-tuning. The video representation learning stage is the phase where an encoder is trained to extract features from the video. In this subsection, we cover the reasons for utilizing captions in the training phase, the methods for generating captions, the approach for fusing information from shots and captions, the procedure for selecting positive and negative samples based on the generated captions, and how all of this contributes to the execution of contrastive learning in the model. The video scene segmentation stage involves fine-tuning the model to align with the scene segmentation task. In this subsection, we introduce a binary classification learning model based on the features extracted through the encoder and labeled data.

A. VIDEO REPRESENTATION LEARNING

1) BASELINE MODEL

As a base model for video representation learning, MoCo [17] is used for self-supervised learning. MoCo [17] is contrastive learning method that utilizes queries, positive samples, and negative samples. Overall process of MoCo is as follows. 1) Receive input image data x, which could be a sample or batch of samples. 2) Apply two augmentations (query augmentation: Aug_q, key augmentation: Aug_k) to input x for query and key sample($x_q = \text{Aug}_q(x)$, $x_k = \text{Aug}_k(x)$). 3) Utilize separate encoders(query encoder: f_q, key encoder: f_k) to extract feature embedding of augmented query and key samples.($\mathbf{x}_q = f_q(x_q)$, $\mathbf{x}_k = f_k(x_k)$) 4) Construct positive,

and negative samples. The positive samples are equal to key samples. The key samples from the previous step are updated to a fixed size(65536) queue structure to construct negative samples. 5) Compute the contrastive loss by using query, positive and negative samples. 6) Update the parameters of query encoder only. Key encoder's parameters are updated by momentum of query parameters as described in 1

$$\theta_k = m\theta_k + (1-m)\theta_q \tag{1}$$

θ_k, θ_q, and m represent parameters of key encoder, parameters of query encoder, and momentum value respectively.

Query-specific augmentation is applied for generating queries, while for positive and negative samples, key-specific augmentation is applied. For query-specific augmentation, operations such as resize and crop, color jitter, grayscale, gaussian blur, and horizontal flip are used. For key-specific augmentation, the same augmentations are applied, except for color jitter and grayscale, which are related to color transformation. Our model also uses the augmentation methods employed in MoCo [17]. Augmentation is applied exclusively to video data and is not employed for caption data.

2) USAGE OF CAPTION

Before delving into an explanation of the model, we provide a description of the limitations of visual data and the rationale behind choosing captions from various text data sources. Previous research related to Video Scene Segmentation (VSS) has focused on using visual data to distinguish positive and negative samples. The method of extracting positive samples from adjacent shots [13], has a limitation in that it does not consider the relevance of shots that are farther apart. The clustering-based approach [15] as seen in Fig.1 (b) can result in shots that should belong to the same cluster being classified into different clusters. As pointed out in [14], it's worth noting that within a video, there can be shots belonging to the same scene that exhibit different visual similarities, and conversely, shots from different scenes that share similar visual similarities (Fig. 1 (a)). We believe that these issues stem from the inherent limitations of visual data due to its ambiguity.

To address the ambiguity in visual data, we considered a method of utilizing information from text data to distinguish aspects that may be challenging to differentiate using visual data alone. The MovieNet [27] dataset, used in this paper, primarily provides various text data such as genre, synopsis, subtitles, and more. Reference [26] proposed that utilizes the genre of movie to select positive samples and performs contrastive learning. However, genre data, being information about the entire movie, is not well-suited for learning to distinguish scenes within a movie. Text data for the VSS task should indeed contain information about each scene in the movie in order to effectively distinguish between scenes. Using data like synopsis and subtitle from MovieNet [27], which may not be well-aligned with the shot and scene in the movies, is not suitable for the task. Therefore, in order to utilize data aligned with scenes while representing information about the scenes, we use generated captions. Considering that unlabeled videos lack information about scene boundaries, we decided to utilize captions for shots that make up the scenes rather than captions specifically describing the scenes.

3) CAPTION GENERATION

In this section, we provide an explanation of the model used to generate captions for shots. Unlike the typical video shots, which consist of hundreds or thousands of frames, the shot data provided by MovieNet [27] is structured with three frames selected from the entire frames. We utilize a captioning model to generate captions for each individual frame. The captioning model is used only for generating captions, and there was no need for additional training. We use a zero-shot captioning model that generates captions directly from the given video data. For the zero-shot captioning model, we utilized a model [28] that combines a vision transformer [29] and GPT-2 [30]. Furthermore, we employed a sentence transformer [31] to extract embedding vectors for each caption.

To transform the generated captions, which are originally frame-specific, into captions for shots, we use the method as follows: 1) Use the caption embedding from the middle of the shot($f_{mid}(\mathrm{fc}_1, \cdots, \mathrm{fc}_F)$). 2) Use the mean of each caption embedding as the embedding vector for the corresponding shot($f_{mean}(\mathrm{fc}_i, \cdots, \mathrm{fc}_F)$). 3) Concatenate the embedding of the three frames to create the embedding for the shot($f_{concat}(\mathrm{fc}_i, \cdots, \mathrm{fc}_F)$). fc represents 'frame caption' and i represents number of frame in a shot. The results for these three methods are detailed in the experiment section.

4) SHOT-CAPTION CROSS MODALITY REPRESENTATION

To achieve better efficiency in semantically distinguishing scenes, it's essential to consider not only the information of shots and captions separately but also the combined information when both modalities are merged. Reference [32] proposes a method to combine video and caption representations to gain complementary advantages from both modalities for enhanced scene understanding. Taking inspiration from the approach used in the [32] paper, this paper employs a straightforward method for the interaction of the two modalities by adding shot embeddings and caption embeddings. To be more specific about the method, batch size(B) of shot embeddings $\mathbf{e_s} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots \mathbf{s}_B\}(\mathbf{s}_i \in \mathbb{R}^d)$ are fed into the mixup function f_{mixup} along with caption embeddings $\mathbf{e_c} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots \mathbf{c}_B\}(\mathbf{c}_i \in \mathbb{R}^d)$. $\mathbf{s}_i, \mathbf{c}_i$ represent the embedding vector for the i-th shot and caption in batch, d represents the embedding dimension, and B represents the batch size. The overall process is in the equation (2).

$$\begin{aligned} \mathbf{s}_i &= \mathrm{Concat}(\mathrm{f}_1, \cdots, \mathrm{f}_\mathrm{F}) \\ \mathbf{c}_i &= f_{\mathrm{frame}\to\mathrm{shot}}(\mathrm{fc}_1, \cdots, \mathrm{fc}_\mathrm{F}) \\ f_{\mathrm{mixup}}(\mathbf{e}_s, \mathbf{e}_c) &= \mathbf{s}_i + \mathrm{c}_i \qquad (i = 1, 2, \cdots, B) \end{aligned} \tag{2}$$

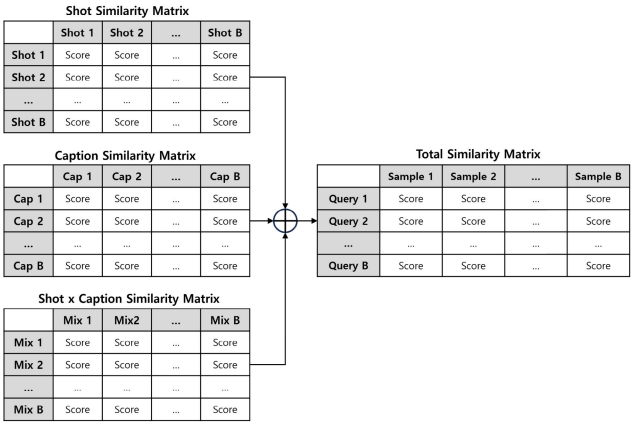

FIGURE 5. Similarity matrix. The left three matrices represent the similarity score matrices constructed for each modality (shot, caption, shot × caption), and the right matrix is the total similarity score matrix obtained by combining them. Each row of the leftmost column represents the queries, and the other columns contain the similarity scores between the queries and the respective samples.

F represents the number of frames in a shot, f and fc represent frame embedding and caption embedding of the frame, respectively. $f_{\text{frame}\rightarrow\text{shot}}$ refers to the transforming the embedding of a caption into an embedding for a shot as discussed in Sec.III-A3 We symbolize the mixup form of shot and caption as "shot × caption".

5) SAMPLE SELECTION

In this section, we introduce the methods for selecting positive and negative samples for contrastive learning using the shot, caption, and shot × caption. A new sampling method based on the similarity matrix is proposed in this section. The similarity score matrix, as defined in this paper, is a matrix composed of similarity scores that measure how similar each shot, caption, and shot × caption is. Based on the similarity score matrix, the sample with the highest similarity score is selected as the positive sample, while the rest are chosen as negative samples for calculating the contrastive loss. The specific method for computing the similarity score and sampling is as follows:

Similarity Score Matrix: The similarity score is represented as shown in (3), where it is calculated as the cosine similarity by taking the dot product of the embedding vectors for each modality. $\{\mathbf{e_s}, \mathbf{e_c}, \mathbf{e_{sc}}\}$ represent the embedding vectors for shot, caption, and shot × caption, each consisting of a batch size B number of values. For example, when calculating the similarity matrix for shot embeddings $\mathbf{e_s} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_B\}(\mathbf{v}_i \in \mathbb{R}^{d\times 1}, \ \mathbf{e}_s \in \mathbb{R}^{d\times B})$, a $\mathbf{M}_s \in \mathbb{R}^{B\times B}$ matrix is created, where B represents the batch size and d represents the embedding dimension.

$$\begin{aligned} \mathbf{M}_s &= \mathbf{e_s}^T \cdot \mathbf{e_s} \\ \mathbf{M}_c &= \mathbf{e_c}^T \cdot \mathbf{e_c} \\ \mathbf{M}_{sv} &= \mathbf{e_{sc}}^T \cdot \mathbf{e_{sc}} \end{aligned} \tag{3}$$

$$\mathbf{M}_{total} = \mathbf{M}_s + \mathbf{M}_c + \mathbf{M}_{sv} \tag{4}$$

The final similarity matrix is computed as described in (4), where each modality's similarity matrix is normalized and then added together. In the similarity matrix (Fig. 5), the leftmost column represents the query, the top row represents samples relative to the query, and the remaining cells indicate the similarity scores for each sample with respect to each query. The resulting similarity matrix, constructed by considering both visual and text information, encapsulates not only the visual aspects of shots but also their semantic aspects.

Similarity Score-Based Sampling: Sampling is based on the similarity scores in the similarity matrix. As in equation (5), the sample with the highest similarity score in similarity matrix is selected as the positive sample for each anchor. During the selection of positive samples, samples that are the same as the anchor are excluded from being chosen as positive samples.

$$\text{positive index} = \mathbf{argmax}(\mathbf{M}_{total}, \text{axis} = 1) \tag{5}$$

Negative samples are composed of all samples except the positive sample. The overall structure of this model is built upon MoCo [16], [17] architecture; hence, it utilizes a pre-defined queue dictionary for negative samples. Negative samples are updated at each step by adding new samples to the queue dictionary, and the oldest negative samples are removed to keep the dictionary up-to-date.

6) OBJECTIVE FUNCTION

In this paper, contrastive loss [33] using positive and negative samples is used as an object function. Similarity between the query and positive sample, as well as the similarity between the query and negative sample(queue), are computed as in (6), and then calculate contrastive loss [33] like (7)

$$\begin{aligned} \mathbf{sim}(q, k^+) &= \mathbf{e}_q \cdot \mathbf{e}_{k^+} \\ \mathbf{sim}(q, k^-) &= \mathbf{e}_q \cdot \mathbf{e}_{k^-} \end{aligned} \tag{6}$$

$$L_{contrastive} = -\log \frac{\sum_{k\in\{k^+\}} e^{(\text{sim(q,k)}/\tau)}}{\sum_{k\in\{k^+ k^-\}} e^{(\text{sim(q,k)}/\tau)}} \tag{7}$$

q, k^+, k^-, τ represent the query, positive sample, negative sample, and the temperature term respectively.

B. VIDEO SCENE SEGMENTATION

In this step, fine-tuning of the classifier model is performed using labeled data to fit it with the requirements of scene segmentation. Fine-tuning is conducted as a binary classification task, distinguishing between whether it is a scene boundary (1) or not (0) by taking into account adjacent shots and the overall context. Reference [15] proposes a model that takes into account long-term dependencies through seq2seq learning using Bi-LSTM. Since this paper aims to demonstrate the effectiveness of representation learning based on shots and captions, the model in the scene segmentation stage remains the same as the previous model [15]. The overall process is depicted as shown in Fig.4 (b). First, the encoder trained in the previous stage is

used to extract features for all shots in the input data. Next, the extracted shot features are classified as either representing scene transition boundary(1) or not(0) using Bi-LSTM and MLP for each shot.

IV. EXPERIMENTS

A. DATASETS

This paper uses the MovieNet [27] dataset which consists of 1,100 movies. Out of the 1,100 movies, they are divided into 660 for training, 220 for validation, and 220 for testing. There are 318 labeled data where it has been indicated whether each shot represents a scene boundary(1) or not(0) for the Video Scene Segmentation(VSS) task. Labeled data are divided into 190 for training, 64 for validation, and 64 for testing purposes. Each movie dataset consists of image files, and each shot is composed of three frames. In MovieNet, besides the movie data, various text data related to each movie are also provided such as IMDb ID, genre, synopsis, subtitles, scene boundaries, and more. Although there are dataset like BBC [11], OVSD [34], and HiREST [35] that can be used for scene segmentation, this paper chose to use MovieNet as a base to facilitate comparisons with previous models [13], [15].

B. EXPERIMENT SETUP

1) MODEL SETTING

In the representation learning stage, training was conducted using 660 unlabeled video data from MovieNet. In previous papers [13], [15], there were experiments where all 1,100 movie data were used for training. However, in a typical scenario, validation and test data are not used during the training phase, so they were excluded from this experiment.

In video representation learning stage, we used two different augmentation scheme(query, key augmentation). Query augmentation is applied for generating query. For query augmentation, resize and crop, color jitter, grayscale, gaussian blur, and horizontal flip are used. Key augmentation is applied using a different method than query augmentation. Key augmentation uses resize and crop, gaussian blur and horizontal flip, excluding color jitter and gray scale, which are methods of changing color.

For extracting features from augmented video shots, ResNet50 [36] is used as the backbone model. The purpose of this paper is to show that the CMS model proposed in this paper is more meaningful than the method proposed in previous studies. In this paper, we propose a method for learning features from videos that encapsulate the inter-scene relationships through contrastive learning, utilizing the similarity between video data and text data. Therefore, to validate the effectiveness of this method, it is necessary to minimize the impact of factors other than the contrastive learning methodology we proposed during the representation learning stage. All previous papers [13], [15] used ResNet50 [36] as the encoder, with variations observed in the methods of positive and negative sampling for contrastive learning. Therefore, for a fair comparison with the new methodology, this paper also employed the same encoder as used in previous studies.

A zero-shot captioning model [28] is used for generating captions, and [31] is employed as the model for extracting caption embeddings. Among these models, ResNet50 [36], which extracts features from the video, is the one that undergoes training. The other models related to captions are in a frozen state.

In the video scene segmentation stage, we use two forms of classifier(Bi-LSTM with MLP and MLP only. For Bi-LSTM + MLP, it consists of three fully connected layers and two Bi-LSTM layers(fc layer1(2048 $\rightarrow$ 1024), Bi-LSTM(1024 $\rightarrow$ 1024), fc layer2(1024 $\rightarrow$ 512), fc layer3(512 $\rightarrow$ 2). This structure is same as SCRL [15]. For MLP only, it consists of three fully connected layers(fc layer1(2048 $\rightarrow$ 4096), fc layer2(4096 $\rightarrow$ 1024, fc layer3(1024 $\rightarrow$ 2). In this stage, the encoder, which was trained in the previous stage(Video representation learning stage), is frozen, and only the classifier(Bi-LSTM with MLP and MLP only) is trained to evaluate the performance of the encoder.

2) METRICS

The video scene segmentation task is a binary classification problem that divides the input video into boundary(1) points and non-boundary(0) points. Given the characteristics of the long video, there are more instances of non-boundary (0) points than those that represent boundaries(1). Therefore, to evaluate the imbalanced labels accurately, it is necessary to consider the bias of labels(0). Taking these factors into consideration, this paper employs the mean of Average Precision(mAP) and F1-score as evaluation metrics, which are robust to precision and recall and suitable for assessing imbalanced data.

The method of calculating each metric is as follows. F1 score and mAP are calculated using precision and recall. Precision calculates the ratio of the correctly predicted instances among the predicted values as boundaries(1) as described in 8, while recall calculates the ratio of the correctly predicted instances among the total number of boundaries(1) as described in 9. TP(True Positive) is the cases where the model correctly predicted the positive class(predicted: 1, label: 1). FP(False Positive) is the cases where the model incorrectly predicted the positive class when the actual class is negative(predicted: 1, label: 0). FN(False Negative) is the cases where the model incorrectly predicted the negative class when the actual class is positive(predicted: 0, label: 1).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

To calculate average precision(AP) we need to interpolate the Precision-Recall curve by computing the precision at each unique recall level and then taking the average. Average

TABLE 1. Result of comparing CMS with other models. This is a comparison table between the previous models and CMS proposed in this paper. The performance values for the previous papers in this table are based on the results reported in the SCRL [15]. M.S 318 refers to the 318 labeled movies from the MovieNet [27] dataset used for scene segmentation.

	Representation Learning				Scene Segmentation		Result	
Methods	Pretrain Data	Backbone	Caption	Dim	Classifier	Eval Data	mAP	F1
ShotCoL [13]	Train	-	-	-	MLP	M.S 318	46.77	45.78
SCRL [15]	Train	Resnet-50	-	2048	MLP	M.S 318	53.74	50.4
ShotCoL [13]	Train	-	-	-	Bi-LSTM + MLP	M.S 318	48.21	46.52
SCRL [15]	Train	Resnet-50	-	2048	Bi-LSTM + MLP	M.S 318	**54.66**	**51.39**
CMS	Train	Resnet-50	ViT+GPT2 [28], sBERT [31]	768	MLP	M.S 318	54.03	51.14
CMS	Train	Resnet-50	ViT+GPT2 [28], sBERT [31]	768	Bi-LSTM + MLP	M.S 318	**55.73**	**52.05**

precision can be calculated by formula as described in 10 Mean of average precision is averaged AP values across all classes. F1 score is a single value combined by precision and recall which provides a balance between the two. F1 score can be calculated by formula as described in 11

$$\text{Average Precision} = \sum_{k} (R_k - R_{k-1}) \cdot P_k \tag{10}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

R_k represents the recall at the k-th point. R_{k-1} represents the recall at the previous operating point. P_k represents the precision at the k-th point.

3) IMPLEMENTATION DETAILS

In the video representation learning stage, the learning rate is set to 0.03, the batch size is 1024(number of shots per iteration), and the epochs are set to 100. For the learning rate, cosine learning rate decay is used. The Encoder(ReNet-50) is initialized with pre-trained parameters from the ImageNet1k [37]. The caption generator [28] generates captions consisting of a maximum of 20 words. The queue size for negative samples is set to 65,536, the momentum value is 0.999, and the temperature term is set to 0.07. For the video scene segmentation stage, epochs are set to 200, train batch size is set to 12, test batch size is set to 1, the initial learning rate is set to 0.1 and the number of shots for MLP and Bi-LSTM is set to 4 and 40, respectively. Both visual data (shot) and text data (caption) have a feature dimension of 768. The training was conducted using two NVIDIA RTX A6000 GPUs, and it took approximately 80 hours to complete.

C. EXPERIMENT RESULT

In this section, we compare the performance of our proposed model, CMS(Contrasting Multi-modal Similarity), with that of existing models [13], [15]. The main focus of this experiment was to determine whether the sampling method that utilizes the similarity of visual data and text data is more effective compared to using only visual data for sampling. Therefore, to demonstrate the effectiveness of CMS, it was necessary to keep the other settings consistent with those used in previous models [13], [15], excluding the sampling method.

Table 1 compares the performance of the CMS with previous models. As indicated in the table 1, the differences between CMS and previous models include the addition of a captioning model and a smaller embedding dimension compared to previous models. When looking at the results, the mAP score is +**7.52** higher than ShotCol [13] and +**1.07** higher than SCRL [15]. In terms of the F1 score, it is +**5.53** higher than ShotCol [13] and +**0.66** higher than SCRL [15]. The results are quite meaningful when considering the smaller embedding size compared to the previous models.

TABLE 2. Ablation study on caption for shot generating method.

	mAP	F1
Middle of Frame	53.65	50.41
Mean of Frame	**55.73**	**52.05**
Concat of Frame	55.16	51.60

D. ABLATION STUDY

This section will delve into an ablation study regarding caption generation for shots and the mixup method for shots and captions.

1) METHODS FOR GENERATING CAPTIONS FOR SHOTS

As mentioned earlier in the model description, the captions we used were applied to individual frames, so they need to be transformed into captions for shots. This paper presents three methods for converting frame captions into caption for shot. 1) Use the embedding vector from the middle of the 3 frames as the embedding vector for the corresponding shot(Middle of Frame). 2) Use the average value of each frame embedding as the embedding vector for the corresponding shot(Mean of Frame). 3) Use the concatenation embedding vector of the three frames as the embedding vector for the corresponding shot(Concat of Frame). Table 2 shows the results for each of the methods. The results show that using the average value of frames performs the best in terms of performance.

TABLE 3. Ablation study on with and without Shot x Caption. Shot × Caption refers to the mixup of shot features and caption features.

	mAP	F1
W Shot × Caption	**55.73**	**52.05**
W.O Shot × Caption	54.44	50.92

TABLE 4. Ablation study on using modality-specific similarity score matrices. Shot × Caption refers to the mixup of shot features and caption features.

Similarity Score Matrix	mAP	F1
Shot	53.79	50.48
Caption	53.54	50.02
Shot, Caption	54.44	50.92
Shot, Caption, Shot × Caption	**55.73**	**52.05**

2) USE OF SHOT AND CAPTION MIXUP REPRESENTATION

As seen in Sec. III-A4, CMS not only uses individual modality data for shot and caption but also combines both modalities(shot × caption) for utilization. We assess whether the method of combining two modalities is effective. Table 3 present the results when using both shot and caption modalities together(shot × caption) and when using each modality separately without their combination. The table shows that when using mixed data combining both shot and caption information, the performance is better. Specifically, there is an increase of **+1.29** in mAP score and **+1.13** in F1 score compared to using each modality separately.

3) USE OF SIMILARITY SCORE

CMS is a similarity score based framework. CMS calculates similarity scores about each modality(shot, caption, shot × caption) and constructs a similarity score matrix based on the calculated score. We assess whether the similarity score based method is effective. Table. 4 compares the performance when using modality-specific similarity score matrices. The results table reveals that even using a single modality for the similarity score matrix yields respectable performance, while the best performance is achieved when all three modalities are utilized.

V. DISCUSSION

A. RESULT ANALYSIS

In this section, we analyze the CMS framework based on the experimental results presented so far. CMS employs a simple sampling method in the network that leverages the similarity between shots and captions, leading to excellent performance even with a smaller dimension size. As seen in Sec.III-A4 and Sec.III-A5, CMS utilizes simple operations (dot product, sum, mean) when leveraging both visual data and text data. In Sec.III-A4, mean and concatenate operations were used to combine shot and caption information, while in Sec.III-A5, dot product was employed to calculate the similarity matrix and sum was utilized to obtain the total similarity matrix for the three modalities (shot, caption, shot × caption). CMS has demonstrated performance exceed to the previous state-of-the-art results, which were achieved through intricate operations such as clustering, using a network comprised of simple operations (Table. 1). Furthermore, CMS offers advantages in terms of dimensionality. As observed in Table. 1, CMS conducted training with a fixed dimension size of 768 for both shots and captions. The results indicate that CMS achieved a notable improvement over ShotCol [13] and displayed a performance higher than that of SCRL [15], which both employed larger dimension sizes.

TABLE 5. Comparison results of the representation learning purpose.

Representation Learning Purpose		mAP	F1
General Representation	MoCo	54.54	51.21
Task-specific Representation	ShotCol	48.21	46.52
	SCRL	54.66	51.14
	CMS	**55.73**	**52.05**

B. METHOD ANALYSIS

In this section, we analyze the methodologies employed in the model. ShotCol [13], SCRL [15], and CMS all follow the same framework as the baseline model, employing MOCO [17], a contrastive learning method explained in the III-A1. The difference lies in the sampling method used for selecting positive and negative samples.

ShotCol [13] selects adjacent shots as positive samples when determining positive samples, while SCRL [15] employs clustering to choose positive samples within the same cluster. CMS proposed in this paper selects positive samples based on the similarity between the input video data(shot) and text data(caption). These methods employ different sampling approaches to learn features that capture the inter-scene relationships necessary for successful video scene segmentation during the representation learning phase. In essence, different sampling methods are employed to learn task-specific features for the video scene segmentation task.

On the other hand, MoCo [17], in contrast, does not undergo a specific sampling process for positive and negative samples. It generates queries for each input image in a batch by applying query augmentation and generates keys by applying key augmentation. Contrastive learning is then performed using these query and key pairs. In other words, MoCo [17] can be characterized as a method that learns a general representation across the entire input data, rather than focusing on learning task-specific representations. The overall learning process of MoCo [17] can be found in III-A1.

We compare these methods, which aim to learn task-specific representations through sampling strategies(ShotCol [13], SCRL [15], CMS), with an approach that aims to learn general representations without specific sampling strategies(MoCo [17]). The results are as described in Table. 5 The results reveal that MoCo outperforms ShotCol(+**6.33** at mAP, +**4.69** at F1) and similar to

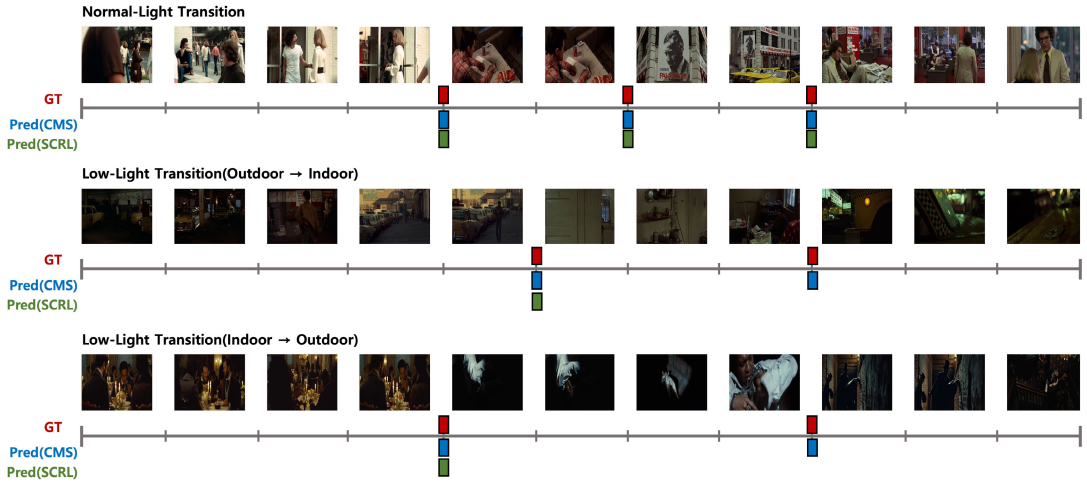

FIGURE 6. Comparison of scene boundary detection results for three situations. The results depict the boundary predictions of each model in three scenarios: normal light transition, low light transition (outdoor → indoor), and low-light transition (indoor → outdoor). In the case of low light transition (outdoor → indoor), it refers to scenes transitioning from outdoor to indoor under low light conditions, while low-light transition (indoor → outdoor) refers scenes transitioning from indoor to outdoor under low light conditions. Ground truth (GT) is represented in red, the predicted boundaries of the CMS model in blue, and the predicted boundaries of the SCRL [15]model in green. The comparison is conducted using SCRL [15], which exhibits the best performance, and the CMS model proposed in this paper.

SCRL [15](**−0.12** at mAP, **+0.07** at F1. In other words, it is observed that methods focusing on learning task-specific representations during the pretraining phase do not yield significantly superior performance compared to methods aiming for general representation learning.

C. LIMITATIONS

While CMS has demonstrated meaningful performance through a new sampling method based on the similarity of shots and captions, there remain challenges in the method for caption generation, the method of mixing the shot and caption representation, and the sampling strategy itself. CMS generates captions for individual frames and subsequently constructs caption features for shots by aggregating the features of each frame along with its corresponding caption. However, the simple summation of embeddings from individual frames may result in information loss and inefficiency. To mitigate information loss and better capture inter-scene relationships, generating captions for sequences of shots instead of individual frames could be more advantageous. In the process of combining shots and captions, it is believed that employing a method aligning the information of each shot and caption effectively like transformer [38], rather than simple summation, could yield better performance.

As observed in V-B, learning a general representation during the representation learning (pretraining) stage tends to achieve superior performance. As a future avenue of exploration, one could attempt methods excluding task-specific sampling during the representation learning stage to enhance the learning of a general representation. Additionally, exploring contrastive learning methods that exclude negative samples could be another interesting avenue. All of the models compared in this paper, utilize a predefined queue structure (with a size of 65565) for negative samples, updating previous key samples into the queue at each step during the selection of negative samples. This method may introduce confusion due to the similarity between samples within the queue and positive samples. Therefore, exploring contrastive learning methods that exclude negative samples could also be a meaningful research direction.

D. VISUALIZATION

To assess the model's performance, we visualized the actual results. Transitions between scenes in movies often occur, particularly from outdoor to indoor or vice versa. In addition, there are many scenes in the film, not only bright scenes such as during the day, but also scenes in dark indoor lighting or scenes in the dark outdoors. Considering these aspects, we examined whether the models accurately predicted scene boundaries in three environments: normal-light transitions, low-light transitions(indoor to outdoor, outdoor to indoor). We compared two models, SCRL [15] and the CMS.

Fig. 6 illustrates the results of this analysis. Both models perform well in predicting scene transition boundaries under normal-light conditions. However, in low-light environments, the CMS model demonstrates the ability to predict the transition boundaries that the SCRL [15] model fails to anticipate. The results indicate that the CMS model exhibits relatively better performance in low-light environments compared to SCRL [15].

Through this, we can observe that the method utilizing both video and text data learns information about aspects that are challenging to distinguish with video data alone.

Consequently, it performs better in predicting boundaries in low-light environments.

VI. CONCLUSION

In this paper, we identify limitations in existing methods that rely solely on visual cues for performing the video scene segmentation task. To address these limitations, we propose a novel contrastive learning framework called Contrasting Multi-Modal Similarity framework (CMS), which is based on measuring the similarity between visual and textual cues. CMS leverages visual cues and text cues as follows: 1) Generate captions corresponding to each shot(Caption Generation). 2) Construct similarity score matrices for each modality(Similarity Score Calculation). 3) Based on this matrix, select similar shots and dissimilar shots(Similarity Score-based Sampling). CMS demonstrated notable performance to existing approaches, despite employing relatively simple techniques(Sec.III-A4, Sec.III-A5) to leverage visual and textual cues. As future research, one could explore new methods for caption generation, techniques for mixing shot and caption representations, and approaches for learning general representations.

REFERENCES

[1] S. Wang, Z. Miao, W. Xu, C. Ma, and M. Li, "Boundary sensitive and category sensitive network for temporal action proposal generation," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 3–19.

[2] A. Cioppa, A. Deliège, S. Giancola, B. Ghanem, M. Van Droogenbroeck, R. Gade, and T. B. Moeslund, "A context-aware loss function for action spotting in soccer videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13123–13133.

[3] J. Tan, J. Tang, L. Wang, and G. Wu, "Relaxed transformer decoders for direct action proposal generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13526–13535.

[4] J. Kim, M. Ma, K. Kim, S. Kim, and C. D. Yoo, "Progressive attention memory network for movie story question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8329–8338.

[5] J. Kim, M. Ma, T. Pham, K. Kim, and C. D. Yoo, "Modality shifting attention network for multi-modal video question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10103–10112.

[6] J. Kim, S. Yoon, D. Kim, and C. D. Yoo, "Structured co-reference graph attention for video-grounded dialogue," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 1789–1797.

[7] S. Yoon, E. Yoon, H. Suk Yoon, J. Kim, and C. D. Yoo, "Information-theoretic text hallucination reduction for video-grounded dialogue," 2022, *arXiv:2212.05765*.

[8] S. Yoon, J. W. Hong, E. Yoon, D. Kim, J. Kim, H. S. Yoon, and C. D. Yoo, "Selective query-guided debiasing for video corpus moment retrieval," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 185–200.

[9] V. T. Chasanis, A. C. Likas, and N. P. Galatsanos, "Scene detection in videos using shot clustering and sequence alignment," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 89–100, Jan. 2009.

[10] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso, "Temporal video segmentation to scenes using high-level audiovisual features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 8, pp. 1163–1177, Aug. 2011.

[11] L. Baraldi, C. Grana, and R. Cucchiara, "A deep Siamese network for scene detection in broadcast videos," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 1199–1202.

[12] A. Rao, L. Xu, Y. Xiong, G. Xu, Q. Huang, B. Zhou, and D. Lin, "A local-to-global approach to multi-modal movie scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10143–10152.

[13] S. Chen, X. Nie, D. Fan, D. Zhang, V. Bhat, and R. Hamid, "Shot contrastive self-supervised learning for scene boundary detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9791–9800.

[14] J. Mun, M. Shin, G. Han, S. Lee, S. Ha, J. Lee, and E.-S. Kim, "Bassl: Boundary-aware self-supervised learning for video scene segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2022, pp. 4027–4043.

[15] H. Wu, K. Chen, Y. Luo, R. Qiao, B. Ren, H. Liu, W. Xie, and L. Shen, "Scene consistency representation learning for video scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14001–14010.

[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9726–9735.

[17] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[18] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE InCVF Int. Conf. Comput. Vis.*, Apr. 2021, pp. 9620–9629.

[19] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1597–1607.

[20] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 22243–22255.

[21] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15745–15753.

[22] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. NIPS*, Dec. 2020, pp. 9912–9924.

[23] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, "Spatiotemporal contrastive video representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6960–6970.

[24] H. Kuang, Y. Zhu, Z. Zhang, X. Li, J. Tighe, S. Schwertfeger, C. Stachniss, and M. Li, "Video contrastive learning with global context," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, p. 3188.

[25] C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, and K. He, "A large-scale study on unsupervised spatiotemporal representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3298–3308.

[26] S. Chen, C.-H. Liu, X. Hao, X. Nie, M. Arap, and R. Hamid, "Movies2Scenes: Using movie metadata to learn scene representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 6535–6544.

[27] Q. Huang, Y. Xiong, A. Rao, J. Wang, and D. Lin, "MovieNet: A holistic dataset for movie understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 709–727.

[28] Y. Tewel, Y. Shalev, R. Nadler, I. Schwartz, and L. Wolf, "Zero-shot video captioning with evolving pseudo-tokens," 2022, *arXiv:2207.11100*.

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[31] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, *arXiv:1908.10084*.

[32] W. Wu, H. Luo, B. Fang, J. Wang, and W. Ouyang, "Cap4video: What can auxiliary captions do for text-video retrieval?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 10704–10713.

[33] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[34] D. Rotman, D. Porat, and G. Ashour, "Optimal sequential grouping for robust video scene detection using multiple modalities," *Int. J. Semantic Comput.*, vol. 11, no. 2, pp. 193–208, Jun. 2017.

[35] A. Zala, J. Cho, S. Kottur, X. Chen, B. Oguz, Y. Mehdad, and M. Bansal, "Hierarchical video-moment retrieval and step-captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 23056–23065.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[37] A. Krizhevsky, I. Sutskever, and G. E Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.
[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

JINWOO PARK (Student Member, IEEE) received the B.A. degree in urban planning and real estate from Chung-Ang University, Seoul, Republic of Korea, in 2018, where he is currently pursuing the M.S. degree.

His research interests include visual-language reasoning and various multi-modal reasoning problems.

JUNGEUN KIM (Graduate Student Member, IEEE) received the B.S. degree in intelligent mechatronics engineering from Sejong University, Seoul, Republic of Korea, in 2022. She is currently pursuing the M.S. degree with Chung-Ang University. Her research interests include causal reasoning, visual-language reasoning, and various multi-modal reasoning.

JAEGWANG SEOK received the B.S. degree in mathematics from Chung-Ang University, Seoul, Republic of Korea, in 2023. He is currently pursuing the M.S. degree in artificial intelligence. His research interests include disentangling representation learning, fairness, and multi-modal reasoning.

SUKHYUN LEE currently pursuing the B.A. degree in artificial intelligence with Chung-Ang University, Seoul, Republic of Korea. His research interests include computer vision and various multi-modal problems.

JUNYEONG KIM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from KAIST, Republic of Korea, in 2015, 2017, and 2021, respectively. He was a Postdoctoral Research Associate with the Artificial Intelligence and Machine Learning Laboratory, School of Electrical Engineering, KAIST. He is currently an Assistant Professor with the Department of AI, Chung-Ang University, Republic of Korea. His research interests include visual-language reasoning, visual question answering, and various video-based problems.