2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.01748

Neighbor Relations Matter in Video Scene Detection

Jiawei Tan, Hongxing Wang*, Jiaxin Li, Zhilong Ou, Zhangbin Qian
Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, China
School of Big Data and Software Engineering, Chongqing University, China
{jwtan, ihxwang, jiaxin_li}@cqu.edu.cn, zlou@stu.cqu.edu.cn, zbqian@cqu.edu.cn

Abstract

Video scene detection aims to temporally link shots for obtaining semantically compact scenes. It is essential for this task to capture scene-distinguishable affinity among shots by similarity assessment. However, most methods relies on ordinary shot-to-shot similarities, which may inveigle similar shots into being linked even though they are from different scenes, and meanwhile hinder dissimilar shots from being blended into a complete scene. In this paper, we propose NeighborNet to inject shot contexts into shot-to-shot similarities through carefully exploring the relations between semantic/temporal neighbors of shots over a local time period. In this way, shot-to-shot similarities are remeasured as semantic/temporal neighbor-aware similarities so that NeighborNet can learn context embedding into shot features using graph convolutional network. As a result, not only do the learned shot features suppress the affinity among similar shots from different scenes, but they also promote the affinity among dissimilar shots in the same scene. Experimental results on public benchmark datasets show that our proposed NeighborNet yields substantial improvements in video scene detection, especially outperforms released state-of-the-arts by at least 6% in Average Precision (AP). The code is available at https://github.com/ExMorgan-Alter/NeighborNet.

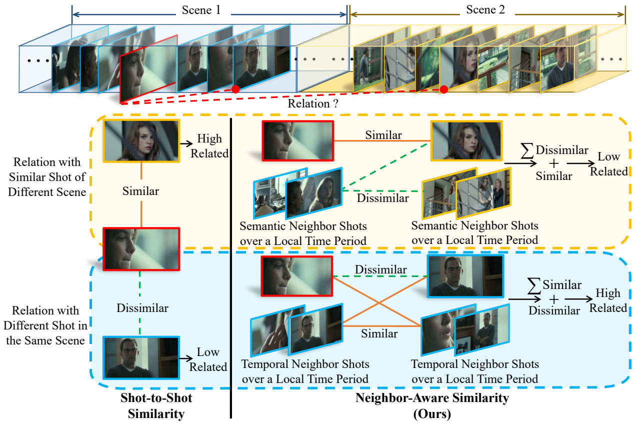

Scene 1
Scene 2
Relation ?
High Related
Relation with Similar Shot of Different Scene
Similar
Similar
Dissimilar
Semantic Neighbor Shots over a Local Time Period
Semantic Neighbor Shots over a Local Time Period
∑Similar + Dissimilar
Low Related
Relation with Different Shot in the Same Scene
Dissimilar
Low Related
Dissimilar
Similar
Temporal Neighbor Shots over a Local Time Period
Temporal Neighbor Shots over a Local Time Period
∑Similar + Dissimilar
High Related
Shot-to-Shot Similarity
Neighbor-Aware Similarity (Ours)

Figure 1. Shot-to-Shot Similarity based methods may deceive themselves into (i) relating different scenes that have similar shots into a single and (ii) relating little with dissimilar shots even though they are in the same scene. In contrast, the proposed Neighbor-Aware Similarity introduces affinity relations between semantic/temporal neighbor shots over a local time period to make shot relations align better with scene relations.

1. Introduction

Video scene detection aims to determine whether the scene happens to change from a shot to the next [24]. It facilitates long video to be truncated into and understood by multiple short yet storytelling units, i.e., scenes, thereby holding great significance across diverse applications, such as text-to-video retrieval [1] and human-centric storyline construction [40].

Like humans, affinity relations among shots are pivotal for computers in discerning whether adjacent shots belong to the same scene or not. Most efforts [23, 29, 42] have been made on ordinary shot-to-shot similarity to explore scene-distinguishable affinity relations among shots. However, we illustrate in Fig. 1 shot-to-shot similarities prefer bracketing similar shots together even though they are from different scenes. On the other hand, a complete scene is highly likely to be fractured into incomplete segments by dissimilar shots. Both circumstances can imply inferior scene detection results.

In fact, shot-to-shot similarities neglect the contribution of context information to the affinity relations among shots. For instance, information from other shots within the same scene can aid in measuring the relations between any two shots. Although it is impossible to know the exact shot contexts at scene scale in the absence of scene

*Corresponding author: Hongxing Wang. This work is supported in part by the National Natural Science Foundation of China under Grant 61976029 and the Key Project of Chongqing Technology Innovation and Application Development under Grant cstc2021jscx-gksbX0033.

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.01748

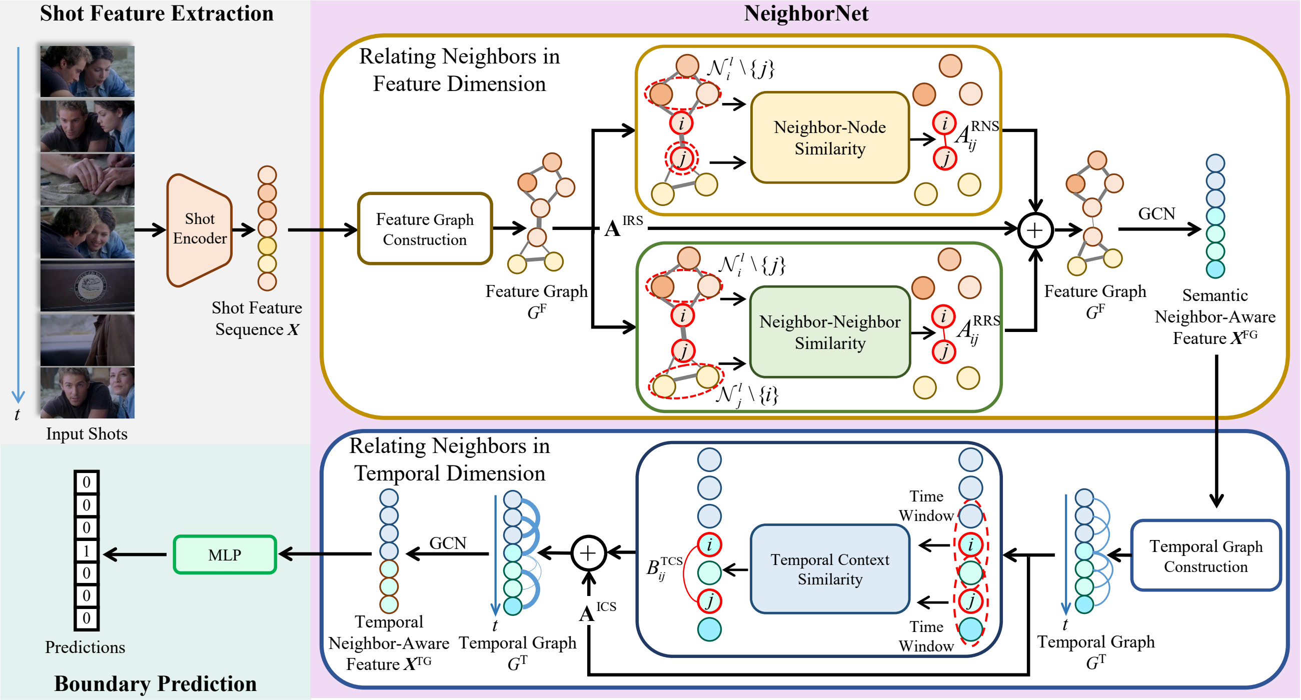

Shot Feature Extraction
NeighborNet
t
Input Shots
Shot Encoder
Shot Feature Sequence X
Relating Neighbors in Feature Dimension
Feature Graph Construction
Feature Graph G^{F}
$\mathbf{A}^{\mathrm{IRS}}$
$\mathcal{N}_i^l \setminus \{j\}$
$\mathcal{N}_j^l \setminus \{i\}$
i
j
Neighbor-Node Similarity
A_{ij}^{RNS}
Neighbor-Neighbor Similarity
A_{ij}^{RRS}
Feature Graph G^{F}
GCN
Semantic Neighbor-Aware Feature X^{FG}
Relating Neighbors in Temporal Dimension
Temporal Graph Construction
Temporal Graph G^{T}
Time Window
Time Window
Temporal Context Similarity
B_{ij}^{TCS}
$\mathbf{A}^{\mathrm{ICS}}$
Temporal Graph G^{T}
GCN
Temporal Neighbor-Aware Feature X^{TG}
MLP
0
0
0
1
0
0
0
Predictions
Boundary Prediction

Figure 2. The proposed NeighborNet for video scene detection. It involves propagating information on the elaborately established feature and temporal graphs to obtain context features for each shot. $\mathcal{N}$ denotes semantically similar shots over a local time period, and $\oplus$ signifies element-wise addition.

information, we are able to cut the contexts in a local time period. Due to the local time constraint, similar shots will have quite different semantic neighbor shots when they are from different scenes. Likewise, dissimilar shots will have quite similar temporal neighbor shots when they are from the same scene. Driven by this insight, we remeasure the shot-to-shot similarity as neighbor-aware similarity. As depicted in Fig. 1, neighbor-aware similarity leverages relations between semantic neighbor shots to reduce the relevance between similar shots from different scenes, as well as relations between temporal neighbors to strengthen the correlation among dissimilar shots within the same scene.

To inject context information captured by neighbor-aware similarity into shot representation for video scene detection, we propose NeighborNet, which undergoes a cascaded graph reasoning process in feature and temporal dimensions. Fig. 2 shows its overview. As can be seen, in feature dimension, NeighborNet implements neighbor-aware similarity using semantic neighbors in two ways combined neighbor-node similarity and neighbor-neighbor similarity. We conduct similarity propagation on a feature graph to generate semantic neighbor-aware features, which will be more discriminative in identifying whether similar shots are from different scenes. In temporal dimension, neighbor relations occur in temporal contexts, by which temporal neighbor-aware similarity is induced. To improve semantic neighbor-aware features further, we depend on them to build a temporal graph and propagate temporal neighbor-aware similarities to generate temporal neighbor-aware features, which become more similar on dissimilar shots from the same scene.

In a nutshell, our contributions include:

- For video scene detection, we are the first to utilize inter-neighbor relations to measure the shot-to-shot affinity relations in both feature and temporal dimensions, which make shot relations align better with scene relations.
- Relying on the proposed shot-to-shot affinity relations, we present NeighborNet to learn shot features discriminative to different scenes through cascaded graph reasoning in feature and temporal dimensions.
- We perform comprehensive evaluations on three video scene detection datasets: MovieNet [21], BBC [2], and OVSD [32]. Our proposed method surpasses previous approaches by large margins in various learning settings.

2. Related Work

Video Scene Detection: Early approaches [8, 17, 31, 36] predominantly rely on unsupervised learning to cluster neighboring shots into scenes. For example, Chasanis *et al.* [7] develop an improved spectral clustering method and employ the fast global k-means algorithm to group shots. Similarly, Rasheed *et al.* [31] utilize color and motion information between shots to generate a similarity graph and partition the graph to identify video scenes. However, these methods exhibit limited performance due to manually designed similarity mechanisms. Recently, supervised methods [21, 34, 35, 41] have demonstrated higher accuracy compared to their earlier unsupervised counterparts. For instance, Rao *et al.* [30] utilize the similarities between adjacent shots as boundary features to detect scenes. Tan *et al.* [34] and Mun *et al.* [29] utilize shot-to-shot similarity to incorporate other shot information into each shot for capturing affinity relations among shots. Building upon this progress, Yang *et al.* [45] introduce a local time window mask to force the attention mechanism to pay attention to the relations between short-range shots. Additionally, Islam *et al.* [23] employ G4S [28] to capture the long-range shot-to-shot relation. However, recent methods depend on raw shot-to-shot similarity to assess relations among shots, potentially leading to the linking of similar shots from different scenes and impeding the integration of dissimilar shots into a complete scene.

Temporal Context for Video Understanding: Temporal context plays a crucial role in various video understanding tasks, including action recognition [37, 44] and action detection [9, 13]. Numerous methods [15, 25, 43] have dedicated significant efforts to modeling temporal context for long videos. For instance, Zhang *et al.* [46] employ a combination of multiscale features and local self-attention to model the longer-range temporal context. Xu *et al.* [43] utilize self-attention to compress long-term memory into a fixed-length latent representation. Islam *et al.* [22] stack multiple S4 [28] with pooling layers to capture the multiscale temporal contexts. However, these methods focus on associating long-range relevant frames/segments, which may not be optimal for video scene segmentation. This is because video scene detection necessitates reinforcement of correlations among shots within the same scene while weakening the correlations among shots from different scenes.

3. Method

3.1. Problem Formulation

Given a video, we manipulate it at shot level. As frames in the same shot belong to a single camera take, we represent each shot using one randomly selected frame inside like the common practice [29, 45]. After that, we treat video scene boundary detection as a binary classification task on shots. The goal is to identify whether each shot is at the end of a scene or not. For convenience, we refer to the ending shot of a scene as a "boundary shot", as illustrated in Fig. 3. In the following, we learn neighbor-aware shot features to make them discriminative in categorizing boundary shots into Class 1, and other shots into Class 0.

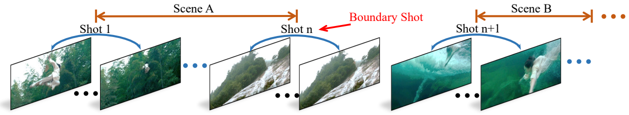


Figure 3. Hierarchical structure of a video sequence. It strings together different scene units, each of which consists of a sequence of shots. The shot ending a scene and abutting next is referred to as a boundary shot.

3.2. NeighborNet

As illustrated in Fig. 2, NeighborNet sequentially propagates information on feature graph and temporal graph to obtain context-embedded shot representation. The feature graph connects similar shots over a local time period. In this context, we introduce semantic neighbors for each shot to attenuate the connections between similar shots from different scenes. After propagating on the feature graph, we have semantic neighbor-aware features that make similar shots from different scenes more distinguishable. Subsequently, we construct the temporal graph based on similar shots generated by the feature graph, where similar shots and shots between each pair of them in time are connected. In the temporal graph, we estimate edge weights by the similarity of their temporal neighbor shots. After that, we propagate the semantic neighbor-aware features on the temporal graph to merge the information of dissimilar shots in the same scene.

Relating Neighbors in Feature Dimension. In accordance with previous methods [23, 29], our model takes as input an N temporally adjacent shots. We extract shot features using a ResNet-50 [19] pre-trained on ImageNet [33], resulting in shot feature sequence $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$. Based these extracted features, we construct a feature graph $G^{\mathrm{F}} = \left\langle \mathbf{X}, \mathbf{E}^{\mathrm{F}} \right\rangle$ to link shots with similar semantics in a local time period, where shots with features $\mathbf{X}$ act as nodes and edges $\mathbf{E}^{\mathrm{F}} = \{E^{\mathrm{F}}_{ij}\}$ between nodes are determined by the semantic similarity between shots as follows:

$$E^{\mathrm{F}}_{ij} = \begin{cases} A^{\mathrm{IRS}}_{ij}, & j \in \mathcal{N}^l_i, \\ -\infty, & \text{otherwise}, \end{cases} \tag{1}$$

where A^{IRS}_{ij} denotes the **i**n-neighbo**r** node **s**imilarity (IRS) derived from the cosine similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$; $\mathcal{N}^l_i$ signifies the top-k similar shots to the shot i within a time window centered on the shot i with a length of l. If the time window does not provide enough shots, we pad it with zero vectors.

The edge weights E^{F}_{ij} are computed based on A^{IRS}_{ij}, which sometimes could result in high similarity values for connections between similar shots from different scenes. To

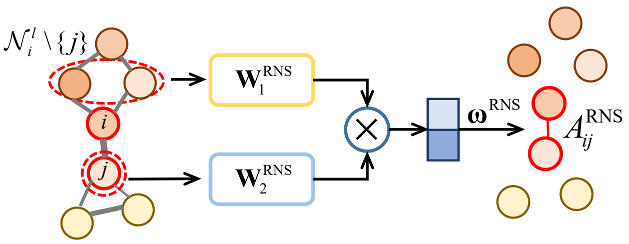

$\mathcal{N}_i^l \setminus \{j\}$
$\mathbf{W}_1^{\text{RNS}}$
$\mathbf{W}_2^{\text{RNS}}$
i
j
$\boldsymbol{\omega}^{\text{RNS}}$
A_{ij}^{RNS}

Figure 4. Neighbor-node similarity (RNS) of node j with respect to node i. $\otimes$ denotes matrix multiplication.

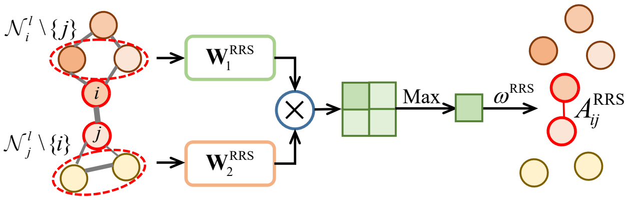

$\mathcal{N}_i^l \setminus \{j\}$
$\mathcal{N}_j^l \setminus \{i\}$
$\mathbf{W}_1^{\text{RRS}}$
$\mathbf{W}_2^{\text{RRS}}$
i
j
Max
ω^{RRS}
A_{ij}^{RRS}

Figure 5. Neighbor-neighbor similarity (RRS) of node j with respect to node i. $\otimes$ denotes matrix multiplication.

suppress the noisy connections, we propose neighbo**r**-**n**ode **s**imilarity (RNS) and neighbo**r**-neighbo**r** **s**imilarity (RRS).

Intuitively, semantic neighbors of a shot are similar to each other. For two shots connected in the feature graph, their neighbors also share similarity with each other. Motivated by these facts, we measure the relations of shot j to shot i by introducing the remaining neighbors of shot i. As depicted by Fig. 4, we have RNS of shot j with respect to the shot i:

$$A_{ij}^{\text{RNS}} = \sum_{g \in \mathcal{N}_i^l \setminus \{j\}} \omega_g^{\text{RNS}} \cdot ((\mathbf{W}_1^{\text{RNS}} \mathbf{x}_g)^\top \mathbf{W}_2^{\text{RNS}} \mathbf{x}_j), \quad (2)$$

where $\mathbf{W}_1^{\text{RNS}}$ and $\mathbf{W}_2^{\text{RNS}}$ are learnable matrices, and ω_g^{RNS} denotes a learnable scale[1]. $\{\omega_g^{\text{RNS}}\}_{g \in \mathcal{N}_i^l \setminus \{j\}}$ can be stacked to form the vector representation $\boldsymbol{\omega}^{\text{RNS}}$.

For RRS, we illustrate it in Fig. 5 and formulate it as the relations between neighbors of two connected shots:

$$A_{ij}^{\text{RRS}} = \omega^{\text{RRS}} \cdot \max_{g,h}((\mathbf{W}_1^{\text{RRS}} \mathbf{x}_g)^\top \mathbf{W}_2^{\text{RRS}} \mathbf{x}_h), \quad (3)$$

where $g \in \mathcal{N}_i^l \setminus \{j\}$ and $h \in \mathcal{N}_j^l \setminus \{i\}$. Notably, when two shots share the same semantic neighbors, it is likely that the pair of nodes belong to the same video scene. We use the "max" operation to select the shared neighbors between shot i and shot j.

Once obtaining RNS and RRS, we proceed to re-measure the edge weights E_{ij}^{F} as:

$$E_{ij}^{\text{F}} = \begin{cases} A_{ij}^{\text{IRS}} + A_{ij}^{\text{RNS}} + A_{ij}^{\text{RRS}}, & j \in \mathcal{N}_i^l, \\ -\infty, & \text{otherwise.} \end{cases} \quad (4)$$

Before RNS and RRS are added, the IRS among similar shots is high. After adding RNS and RRS, the sum of RNS and RRS for similar shots within the same scene should surpass that for similar shots from different scenes. This weakens the connection between similar shots in different scenes and strengthens the connection between similar shots in the same scene.

Next, we perform message passing for the shot nodes on the feature graph G^{F} using the shot features $\mathbf{X}$. Concretely, the shot nodes receive messages from its connected semantically similar shots in the graph G^{F}, increasing the similarity between similar shots within the same scenes. We apply a one-layer graph convolution network (GCN) [27] to conduct graph message passing and inference for obtaining semantic neighbor-aware shot features $\mathbf{X}^{\text{FG}}$:

$$\mathbf{X}^{\text{FG}} = \sigma(\mathbf{X} + \text{GCN}(\mathbf{X}, \text{softmax}(\mathbf{E}^{\text{F}}))), \quad (5)$$

where σ denotes the activation function. Note that we employ softmax for normalization to make the edge weights learnable [39].

Relating Neighbors in Temporal Dimension. In the prior stage, information is propagated solely between similar shots within the same scene, with no effort to enhance relations between dissimilar shots within the same scene. The challenge in solving this problem lies in how to ensure that the enhanced shots indeed belong to the same scene. We find that if two shots are similar, then it is plausible that the shot between them may belong to the same scene. To this end, we build the temporal graph $G^{\text{T}} = \langle \mathbf{X}^{\text{FG}}, \mathbf{E}^{\text{T}} \rangle$ based on similar shots selected by the feature graph G^{F}. The edge weight $\mathbf{E}^{\text{T}} = \{E_{ij}^{\text{T}}\}$ can be formulated as:

$$E_{ij}^{\text{T}} = \begin{cases} B_{ij}^{\text{ICS}}, & j \in S(\mathcal{N}_i^l), \\ -\infty, & \text{otherwise,} \end{cases} \quad (6)$$

where B_{ij}^{ICS} denotes **i**n-**c**ontext node **s**imilarity (ICS) calculated by the cosine similarity between selected semantic neighbor-aware features $\mathbf{x}_i^{\text{FG}}$ and $\mathbf{x}_j^{\text{FG}}$. The valid set of shot j when given shot i complies with $S(\mathcal{N}_i^l) = \{(\min(i,u), \max(i,u)) \,|\, u \in \mathcal{N}_i^l\}$. In particular, we construct the temporal graph using semantic neighbor-aware shot features $\mathbf{X}^{\text{FG}}$ as they endow a higher similarity to similar shots of the same scene compared to the original shot features $\mathbf{X}$. In this way, it ensures that the temporal graph keeps resistant to the influence of connection with shots from different scenes.

To enhance relations between dissimilar shots from the same scene, we measure the relations between shots by

[1] It is worth noting that here and hereinafter, notations $\mathbf{W}$ or ω refers to a learnable matrix or scale, and $\mathbf{W}$ do not change the dimensions of the multiplied vector.

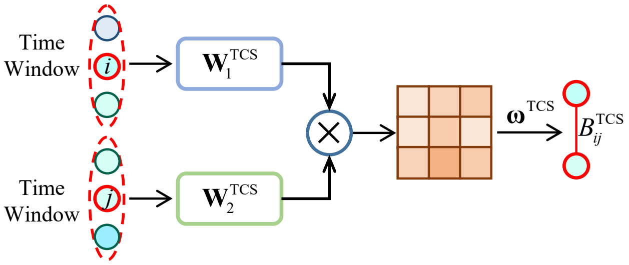

Time
Window
i
$\mathbf{W}_1^{\text{TCS}}$
Time
Window
j
$\mathbf{W}_2^{\text{TCS}}$
$\boldsymbol{\omega}^{\text{TCS}}$
B_{ij}^{TCS}

Figure 6. Temporal context similarity (TCS) of node j with respect to node i. $\boldsymbol{\omega}^{\text{TCS}} = (\omega_{gh}^{\text{TCS}})$, and $\otimes$ denotes matrix multiplication.

utilizing the temporal contexts. Intuitively, if the temporal context surrounding two shots is similar, they are very likely to be in the same scene. We thus reinforce the edge weight between them. Specifically, we design **t**emporal **c**ontext **s**imilarity (TCS) for two connected nodes in temporal graph as shown in Fig. 6,

$$B_{ij}^{\text{TCS}} = \sum_{g,h} \omega_{gh}^{\text{TCS}} \cdot ((\mathbf{W}_1^{\text{TCS}} \mathbf{x}_g^{\text{FG}})^\top \mathbf{W}_2^{\text{TCS}} \mathbf{x}_h^{\text{FG}}), \quad (7)$$

where $g \in \mathcal{T}_r(i)$, $h \in \mathcal{T}_r(j)$, and $\mathcal{T}_r(i)$ denotes a time window with length r centered on shot i. If there is not enough input shots in the time window, we pad it with border shot replications.

Having the temporal context similarities, we re-measure E_{ij}^{T} as follow:

$$E_{ij}^{\text{T}} = \begin{cases} B_{ij}^{\text{ICS}} + B_{ij}^{\text{TCS}}, & j \in S(\mathcal{N}_i^l), \\ -\infty, & \text{otherwise}. \end{cases} \quad (8)$$

After that, we propagate semantic neighbor-aware shot features $\mathbf{X}^{\text{FG}}$ on the temporal graph G^{T}. In the process, each shot node receives information from its connected dissimilar shot in the graph G^{T}. The shot nodes will update its features as temporal neighbor-aware features $\mathbf{X}^{\text{TG}}$ by another one-layer GCN that is different from (5):

$$\mathbf{X}^{\text{TG}} = \sigma(\mathbf{X}^{\text{FG}} + \text{GCN}(\mathbf{X}^{\text{FG}}, \text{softmax}(\mathbf{E}^{\text{T}}))). \quad (9)$$

3.3. Training and Loss Functions

There are two loss functions employed in our model training: the self-supervised loss and the supervised loss. Below, we detail the two losses and the usage of them. It is worth noting that we do not claim technical novelty over the loss functions. Instead, we employ the losses to train our devised NeighborNet model for video scene detection.

Self-supervised loss. For fair comparison with previous methods [23, 29], our self-supervised loss is the same as theirs. It is actually a pseudo-scene boundary prediction loss, where pseudo-scene boundaries are generated by the Modified Dynamic Warping algorithm [29]. The loss here is defined as a binary cross-entropy loss on temporal neighbor-aware features $\mathbf{X}^{\text{TG}}$ of shots:

$$L_p = -\log(h_p(\mathbf{x}_p^{\text{TG}})) - \log(1 - h_p(\mathbf{x}_{p^*}^{\text{TG}})) \quad (10)$$

where $h_p(*)$ represents a multilayer perceptron (MLP) that outputs the probability of a shot being the pseudo-boundary, $\mathbf{x}_{p^*}^{\text{TG}}$ denotes the feature of a randomly selected non-pseudo-boundary shot, and $\mathbf{x}_{\mathbf{p}}^{\text{TG}}$ represents the feature of the pseudo-boundary shot.

Supervised loss. It utilizes ground-truth video scene boundaries to supervise the model training with the binary cross-entropy loss below:

$$L_f = -y_i \log(h_b(\mathbf{x}_i^{\text{TG}})) + (1 - y_i) \log(1 - h_b(\mathbf{x}_i^{\text{TG}})) \quad (11)$$

where $h_b(*)$ is a MLP that is trained from scratch and outputs the probability of a shot being the boundary, $\mathbf{x}_i^{\text{TG}}$ represents a feature for shot i, and $y_i \in \{0, 1\}$ denotes the ground-truth binary label of shot i.

Our proposed NeighborNet can combine the above two losses to achieve different learning ways: **self-supervised learning**, **fully supervised learning**, and **self-supervised transfer learning**. For self-supervised learning, we use the self-supervised loss for model training. In the case of fully supervised learning, a supervised loss is utilized. For self-supervised transfer learning, it has two phases, where we apply the self-supervised loss for pre-training, followed by the utilization of the supervised loss for model fine-tuning.

4. Experiments

4.1. Settings

Datasets: We assess the performance of our method on three widely used video scene detection datasets, *i.e.*, MovieNet [21], BBC [2], and OVSD [32]. **MovieNet** is a comprehensive dataset comprising 1,100 movies, totaling 1.6 million shots. Among these movies, 318 are equipped with ground-truth scene boundary annotations, while the remaining 782 lack such annotations. The 318 annotated movies make up the MovieScenes dataset [30] for video scene detection. MovieScenes is further divided into subsets of 190 movies for training, 64 for validation, and 64 for testing. For different learning ways, we always evaluate our model on the test split of MovieScenes. In the self-supervised scenario, we utilize all the 782 unlabeled videos from MovieNet for pre-training. In the supervised setting, we utilize 190 training videos from MovieScenes for training. For self-supervised transfer learning, we utilize all the 782 unlabeled videos from MovieNet for pre-training, 190 training videos from MovieScenes for fine-tuning. **OVSD** [32] comprises 21 short films, each with an average duration of 30 minutes. It encompasses a total of 10,000 shots and 300 scenes, extracted from movie scripts. As the dataset lacks predefined splits, we adopt a common practice as in prior studies [23, 29, 42]. That is, we train our model using the MovieNet dataset and subsequently evaluate it on the OVSD dataset without further fine-tuning. **BBC** [2] consists of 11 episodes from the BBC educational TV series *Planet Earth* [4]. These videos have an average duration of 50 minutes and encompass a total of 670 scenes and 4.8K shots. Similarly to our evaluation on the OVSD dataset, we train our model using the MovieNet dataset and evaluate it on the BBC dataset without additional fine-tuning, in line with prior research practices [23, 29, 42].

Metrics: To measure the performances, we use the same evaluation metrics used in prior methods [23, 29, 42], which include the Average Precision (AP), the mean Intersection over Union (mIoU), and the F1-score (F1). These metrics serve to evaluate the effectiveness of video scene detection, with higher values indicating superior performance.

Implementation Details: We take $N = 21$ neighboring shots as input to our model. In NeighborNet, we set the activation functions in Eqs. (5) and (9) as ReLU [18]. In Eq. (1), we specify the sliding window scale as $l = 10$, and feature neighbor scale as $k = 5$. In Eq. (7), we set the temporal neighbor scale r to 5. We employ the Adam [26] optimizer with a mini-batch size of 512. For fully supervised learning and self-supervised learning, we initialize the learning rate at 10^{-4}. In the case of self-supervised transfer learning, we set the initial learning rate to 10^{-4} for pre-training and reduce it to 10^{-5} for fine-tuning. Across all training stages, we apply a linear warm-up strategy during the initial epoch, followed by a learning rate decay according to a cosine schedule [20]. We train our NeighborNet on a NVIDIA RTX 3060 GPU.

4.2. Comparison with State-of-the-Art Methods

To demonstrate the advantage of our complete solution, we compare with the state-of-the-art methods, including TranS4mer [23] (CVPR'23), BaSSL [29] (ACCV'22), Movies2Scenes [12] (CVPR'23), SCRL [42] (CVPR'22), Temporal Perceiver [34] (TPAMI'23), and MHRT [41] (ICCV'23). To further justify the proposed design for NeighborNet, we implement its three alternative baselines according to state-of-the-art model architectures: Transformer [38] (NIPS'17), GATv2 [6] (ICLR'22), and HopGNN [10] (CVPR'23). For Transformer, we repeat it for two layers to replace NeighborNet. For GATv2 and HopGNN, we use each in place of the neighbor relation learning modules built upon Eq. (2), Eq. (3), and Eq. (7).

Results on MovieNet [21]. Table 1 illustrates that our method consistently reaches superior performance compared to all the counterparts across all evaluation metrics in various experimental scenarios. Notably, our method surpasses the latest state-of-the-art TranS4mer [23] by a margin of at least **11%** in all metrics. These improvements can be attributed to our effective approach in estimating shot-to-shot relations utilizing their neighbor shots.

Methods	AP	mIoU	F1
Self-Supervised Learning			
TimeSformer [5] (ICML'21)	32.5	37.8	31.5
BaSSL [29] (ACCV'22)	31.5	39.6	32.6
TranS4mer [23] (CVPR'23)	34.5	39.6	33.4
Transformer [38] (NIPS'17)	23.0	42.9	22.5
Our Graphs + GAT-v2 [6] (ICLR'22)	43.1	48.4	37.5
Our Graphs + HopGNN [10] (CVPR'23)	47.0	48.9	40.9
Ours	**51.2**	**52.9**	**46.4**
Self-Supervised Transfer Learning			
ShotCoL [11] (CVPR'21)	53.4	-	49.7
Movies2Scenes [12] (CVPR'23)	54.2	-	-
SCRL [42] (CVPR'22)	54.8	-	51.4
BaSSL [29] (ACCV'22)	57.4	50.6	47.0
TimeSformer [5] (ICML'21)	59.6	50.8	48.0
SSM [16] (ICLR'22)	59.7	51.3	48.4
TranS4mer [23] (CVPR'23)	60.8	52.0	48.4
Transformer [38] (NIPS'17)	53.1	54.5	48.9
Our Graphs + GAT-v2 [6] (ICLR'22)	67.0	60.8	55.7
Our Graphs + HopGNN [10] (CVPR'23)	67.9	58.4	54.0
Ours	**71.9**	**64.5**	**62.7**
Fully Supervised Learning			
Siamese [3] (MM'15)	28.1	36.0	-
MS-LSTM [21] (ECCV'20)	46.5	46.2	-
LGSS [30] (CVPR'20)	47.1	48.8	-
Temporal Perceiver [34] (TPAMI'23)	53.3	53.2	-
MHRT [41] (ICCV'23)	54.8	51.2	46.3
Transformer [38] (NIPS'17)	50.0	46.5	40.9
Our Graphs + GAT-v2 [6] (ICLR'22)	58.1	57.9	53.6
Our Graphs + HopGNN [10] (CVPR'23)	60.1	56.4	53.5
Ours	**64.0**	**61.2**	**57.8**

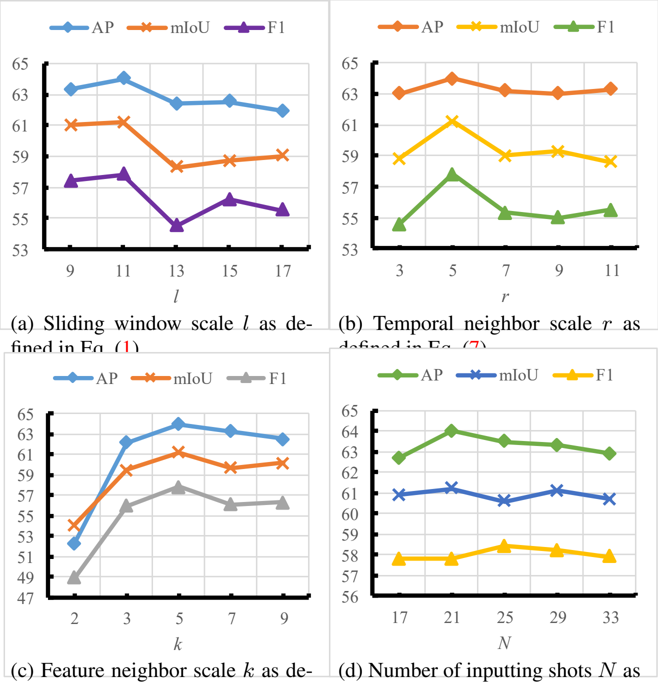
Table 1. Comparisons on MovieNet [21]. The best are indicated in **Bold**. "-" denotes that result does not get published.

It is evident that our method outperforms all baseline methods comprehensively. Specifically, our NeighborNet architechture surpasses Transformer [38] across a wide range of metrics, highlighting the effectiveness of our approach. Moreover, our method outperforms GATv2 [6] due to leveraging neighboring shots to capture the relations between shots. More importantly, our method performs much better than HopGNN [10] that also measures the relations between neighboring shots, indicating the suitability of our method for video scene segmentation.

Transfer Evaluation. We demonstrate the generalization capability of our NeighborNet in comparison with recent methods [11, 23, 29, 42] in Table 2. All models used have undergone self-supervised pre-training and fine-tuning on MovieNet [21]. We test these MovieNet-trained model without any additional fine-tuning on BBC [2] and OVSD [32]. The results demonstrate that our proposed method achieves the best performance among all the comparisons on the OVSD and BBC datasets, which verifies the

Method	OVSD			BBC		
	AP	mIoU	F1	AP	mIoU	F1
ShotCoL [11]	25.5	-	-	28.0	-	-
SCRL [42]	38.8	-	-	30.2	-	-
BaSSL [29]	29.0	-	-	40.2	-	-
Tran4mer [23]	36.0	-	-	43.6	-	-
Transformer [38]	25.8	43.0	25.4	32.2	33.9	18.8
Our Graphs + GATv2 [6]	38.7	49.6	28.7	37.8	44.0	31.5
Our Graphs + HopGNN [10]	41.9	47.6	28.8	44.0	46.0	34.7
Ours	**47.3**	**50.6**	**29.4**	**50.6**	**49.5**	**35.3**

Table 2. Evaluation on OVSD [32] and BBC [2]. "-" denotes that result does not get published.



(a) Sliding window scale l as defined in Eq. (1).

(b) Temporal neighbor scale r as defined in Eq. (7).

(c) Feature neighbor scale k as defined in Eq. (1).

(d) Number of inputting shots N as defined in Sec. 3.2.

Figure 7. Impact of hyperparameters.

Method	AP	mIoU	F1
ResNet	40.0	44.4	40.1
ResNet + NeighborNet	64.0	61.2	57.8
ViT	34.1	45.0	36.6
ViT + NeighborNet	65.5	62.7	58.9

Table 3. Impact of different shot encoders.

Feature Graph			Temporal Graph		AP
IRS	RNS	RRS	ICS	TCS	
-	-	-	-	-	40.0
✓	-	-	-	-	52.5
✓	✓	-	-	-	56.9
✓	✓	✓	-	-	60.0
-	-	-	✓	-	50.4
-	-	-	✓	✓	58.8
✓	-	-	✓	-	53.9
✓	✓	✓	✓	✓	64.0

Table 4. Ablation study on NeighborNet in different inclusions of in-neighbor node similarity (IRS), neighbor-node similarity (NNS), neighbor-neighbor similarity (RRS), in-context node similarity (ICS), and temporal context similarity (TCS). ✓signifies "included", while - "excluded".

strong generalization capability of NeighborNet.

4.3. Ablation Studies

We depend on MovieNet [21] to present various ablation analysis of our proposed method in video scene detection. The following experiments are conducted in fully supervised fashion. We have also included more ablation experiments in the supplementary.

Sliding Window Scale l. Fig. 7a illustrates the impact of the sliding window scale l as defined in Eq. (1). It is evident that all metrics reach their highest values when $l = 11$. This suggests that choosing $l = 11$ strikes a ideal capacity for our NeighborNet to capture sufficient contexts in both feature and temporal dimensions.

Temporal Neighbor Scale r. Fig. 7b presents the effects of the temporal neighbor scale r as defined in Eq. (7). It can be seen that the proposed method achieves the best performance when $r = 5$. This scale selection reaches an optimal efficacy to ensure adequate temporal contexts are from the same scene.

Feature Neighbor Scale k. In Fig. 7c, we evaluate the performance of our method under different top-k similar shots defined in Eq. (7). The metric curves show a rapid ascent from $k = 2$ to $k = 3$, reaching their peak at $k = 5$. Beyond this point, with further increases in k, the curves slowly decline. The result implies that top-5 feature neighbors include more semantically similar contexts while better avoiding semantic interference from different scenes.

Number of Inputting Shots N. Fig. 7d depicts the effect of different numbers of input shots on performance. Notably, the fluctuation in each metric curve remains within 1% as the number of input shots changes. This observation implies the robust adaptability of our proposed method to videos of varying lengths.

Different Shot Encoders. Table 3 provides the performance of our proposed NeighborNet when combined with different shot encoders including ResNet-50 [19] and ViT-S/16 [14]. As comparisons, we also implement the baseline video scene detection results, where only shot encoders ResNet and ViT of themselves are applied. It is obvious that our NeighborNet improves the baseline encoders by at least 24% AP, which highlights the effectiveness of the proposed method. Besides, our NeighborNet combined with either shot encoder achieves state-of-the-art performance, which implies that our method has a strong ability to learn shot contexts for video scene detection.

Network Component. Table 4 presents the results of an ablation study, which assesses the contributions of different components of our proposed method to overall performance. Notably, the first row of Table 4 reports the baseline

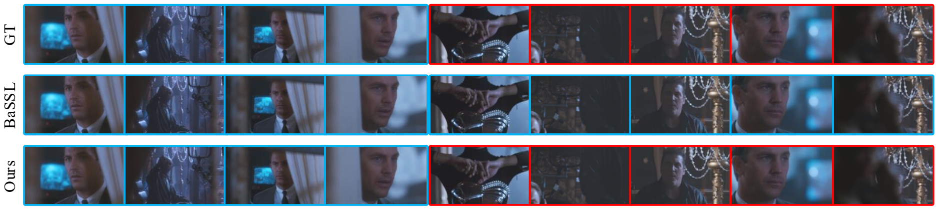


Figure 8. Qualitative comparison of the proposed method with the previous method BaSSL [29]. GT denotes ground-truth scene boundaries for reference. The borders of the same color indicate the shots from the same scene.

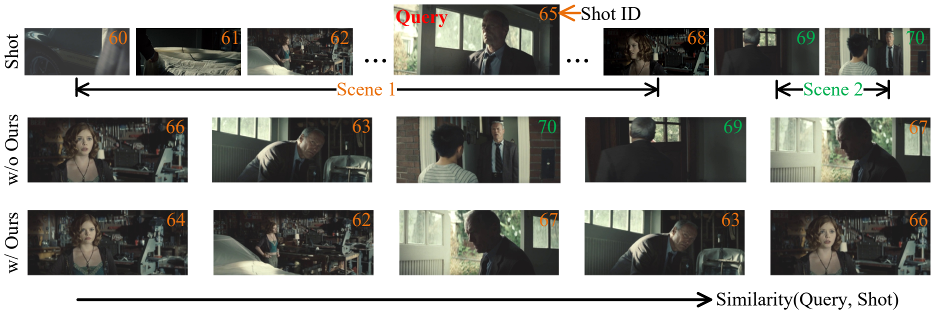


Figure 9. Visualization of relations between query shots and others. Shots whose IDs share the same color font stem from the same video scene. "w/o ours" indicates the result obtained without utilizing features learned by NeighborNet, while "w/ ours" represents the outcome achieved with the use of NeighborNet-learned features.

results of feeding the ResNet-encoded shot features $\mathbf{X}$ into the MLP for video scene detection prediction. Overall, we observe a consistent enhancement in Average Precision (AP) as more proposed modules are incorporated. Compared to using IRS in the feature graph, we add RNS and RRS to increase AP by 7.5 %, which may be attributed to their effects of mitigating noisy connections due to similar shots from different scenes. AP for the temporal graph with ICS (shot-to-shot similarity) is 50.4%. After incorporating the proposed TCS, AP increases to 58.8%, highlighting its validity of capturing the temporal relations between different shots.

4.4. Qualitative Results

All qualitative results are obtained from the model trained with self-supervised transfer learning. Additional visualizations can be found in the supplementary.

Visualization of Detection Results. In Fig. 8, we present a sample result of video scene detection on the MovieNet dataset [21]. We provide ground-truth annotations for reference and also include the previous state-of-the-art method, BaSSL [29], for comparison. BaSSL utilizes raw shot-to-shot similarities to capture shot contexts, which appears to confuse different scenes. In contrast, our method provides a correct detection, thanks to our proposed RNS and RRS to suppress the connections between similar shots from different scenes. The detection results that highlight the role of TCS are included in the supplementary.

Visualization of Shot Relations. Fig. 9 provides a deeper understanding of the relationships learned by the proposed NeighborNet. We visualize the top-5 similar shots to a query shot within a time window of 11 shots. The visualizations demonstrate that, when compared to the scenario w/o NeighborNet, w/ NeighborNet not only strengthens relationships between similar shots within the same scene, but also enhances connections between dissimilar shots. Additionally, NeighborNet effectively weakens the connections between similar shots in different scenes. These improvements can be attributed to the introduction of both semantic and temporal neighbors.

5. Conclusion

We propose a novel shot context learning method, NeighborNet, for video scene detection. It constructs a feature graph that links semantically similar shots over a local time period to enhance shot relations within the same scene, where the edge weights are guided by comparing semantic neighbor shots. Besides, to enhance the relations between dissimilar shots in the same scene, we further in the temporal dimension construct a temporal graph that connects each pair of similar shots to those intervening. The edge weights are induced by the similarities of the temporal neighbor shots. Experimental results demonstrate that our proposed NeighborNet can effectively capture shot contexts and thus achieving superior performance compared to state-of-the-art methods.

References

[1] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *ACCV*, pages 460–479, 2020. 1

[2] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *ACM Int. Conf. Multimedia*, pages 1199–1202, 2015. 2, 5, 6, 7

[3] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *ACM Int. Conf. Multimedia*, pages 1199–1202, 2015. 6

[4] BBC. Planet earth. https://www.bbc.co.uk/programmes/b006mywy. 6

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, pages 813–824, 2021. 6

[6] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *Int. Conf. Learn. Represent.*, 2022. 6, 7

[7] Vasileios Chasanis, Aristidis Likas, and Nikolas P. Galatsanos. Scene detection in videos using shot clustering and sequence alignment. *IEEE Trans. Multimedia*, 11(1):89–100, 2009. 2

[8] Vasileios Chasanis, Aristidis Likas, and Nikolas P. Galatsanos. Scene detection in videos using shot clustering and sequence alignment. *IEEE Trans. Multimedia*, 11(1):89–100, 2009. 2

[9] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. DCAN: improving temporal action detection via dual context aggregation. In *AAAI*, pages 248–257, 2022. 3

[10] Jie Chen, Zilong Li, Yin Zhu, Junping Zhang, and Jian Pu. From node interaction to hop interaction: New effective and scalable graph learning paradigm. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7876–7885, 2023. 6, 7

[11] Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot contrastive self-supervised learning for scene boundary detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9796–9805, 2021. 6, 7

[12] Shixing Chen, Chun-Hao Liu, Xiang Hao, Xiaohan Nie, Maxim Arap, and Raffay Hamid. Movies2scenes: Using movie metadata to learn scene representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6535–6544, 2023. 6

[13] Rui Dai, Srijan Das, Kumara Kahatapitiya, Michael S. Ryoo, and François Brémond. MS-TCT: multi-scale temporal convtransformer for action detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 20009–20019, 2022. 3

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021. 7

[15] Adriano Fragomeni, Michael Wray, and Dima Damen. Contra: (con)text (tra)nsformer for cross-modal video retrieval. In *ACCV*, pages 451–468, 2022. 3

[16] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *Int. Conf. Learn. Represent.*, 2022. 6

[17] Bo Han and Weiguo Wu. Video scene segmentation using a novel boundary evaluation criterion and dynamic programming. In *Int. Conf. Multimedia and Expo*, pages 1–6, 2011. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Int. Conf. Comput. Vis.*, pages 1026–1034, 2015. 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 3, 7

[20] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 558–567, 2019. 6

[21] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *Eur. Conf. Comput. Vis.*, pages 709–727, 2020. 2, 3, 5, 6, 7, 8

[22] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In *Eur. Conf. Comput. Vis.*, pages 87–104, 2022. 3

[23] Md Mohaiminul Islam, Mahmudul Hasan, Kishan Shamsundar Athrey, Tony Braskich, and Gedas Bertasius. Efficient movie scene detection using state-space transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18749–18758, 2023. 1, 3, 5, 6, 7

[24] E. Katz and R.D. Nolen. *The Film Encyclopedia 7th Edition: The Complete Guide to Film and the Film Industry*. HarperCollins, 2012. 1

[25] Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my temporal context: Multimodal egocentric action recognition. In *Brit. Mach. Vis. Conf.*, page 268, 2021. 3

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2015. 6

[27] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Int. Conf. Learn. Represent.*, 2017. 4

[28] Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. In *Int. Conf. Learn. Represent.*, 2023. 3

[29] Jonghwan Mun, Minchul Shin, Gunsoo Han, Sangho Lee, Seongsu Ha, Joonseok Lee, and Eun-Sol Kim. Bassl: Boundary-aware self-supervised learning for video scene segmentation. In *ACCV*, pages 485–501, 2022. 1, 3, 5, 6, 7, 8

[30] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10143–10152, 2020. 3, 5, 6

[31] Zeeshan Rasheed and Mubarak Shah. Detection and representation of scenes in videos. *IEEE Trans. Multimedia*, 7(6): 1097–1105, 2005. 2, 3

[32] Daniel Rotman, Dror Porat, and Gal Ashour. Robust and efficient video scene detection using optimal sequential grouping. In *ISM*, pages 275–280, 2016. 2, 5, 6, 7

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 3

[34] Jing Tan, Yuhong Wang, Gangshan Wu, and Limin Wang. Temporal perceiver: A general architecture for arbitrary boundary detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(10):12506–12520, 2023. 3, 6

[35] Jiawei Tan, Hongxing Wang, and Junsong Yuan. Characters link shots: Character attention network for movie scene segmentation. *ACM Trans. Multim. Comput. Commun. Appl.*, 20(4):94:1–94:23, 2024. 3

[36] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Storygraphs: Visualizing character interactions as a timeline. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 827–834, 2014. 2

[37] Anirudh Thatipelli, Sanath Narayan, Salman Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Bernard Ghanem. Spatio-temporal relation modeling for few-shot action recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 19926–19935, 2022. 3

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, pages 5998–6008, 2017. 6, 7

[39] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Int. Conf. Learn. Represent.*, 2018. 4

[40] Paul Vicol, Makarand Tapaswi, Lluís Castrejón, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8581–8590, 2018. 1

[41] Xi Wei, Zhangxiang Shi, Tianzhu Zhang, Xiaoyuan Yu, and Lei Xiao. Multimodal high-order relation transformer for scene boundary detection. In *Int. Conf. Comput. Vis.*, pages 22081–22090, 2023. 3, 6

[42] Haoqian Wu, Keyu Chen, Yanan Luo, Ruizhi Qiao, Bo Ren, Haozhe Liu, Weicheng Xie, and Linlin Shen. Scene consistency representation learning for video scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14001–14010, 2022. 1, 6, 7

[43] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. In *Adv. Neural Inform. Process. Syst.*, pages 1086–1099, 2021. 3

[44] Jiewen Yang, Xingbo Dong, Liujun Liu, Chao Zhang, Jiajun Shen, and Dahai Yu. Recurring the transformer for video action recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14043–14053, 2022. 3

[45] Yang Yang, Yurui Huang, Weili Guo, Baohua Xu, and Dingyin Xia. Towards global video scene segmentation with context-aware transformer. In *AAAI*, pages 3206–3213, 2023. 3

[46] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *Eur. Conf. Comput. Vis.*, pages 492–510, 2022. 3