Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Identify Video Scene Boundaries using Transformer Encoding Linker Network (TELNet)

SHU-MING TSENG[1], ZHI-TING YEH[1], CHIA-YANG WU[1], JIA-BIN CHANG[2], MEHDI NOROUZI[3]
[1]Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan, 10608
[2]Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan, 10608
[3]Department of Electrical and Computer Engineering, University of Cincinnati, Cincinnati, Ohio, USA, 45221

Corresponding author: SHU-MING TSENG(e-mail: shuming@ntut.edu.tw)

ABSTRACT Videos are made up of a series of scenes or chapters, which are each a sequence of semantically related shots. In this paper, we propose a Transformer Encoding Linker Network (TELNet), which learns the relationship among video shots and identifies scene boundaries without prior knowledge of video structure – such as the number of scenes in a video.
Shots are represented using deep features extracted from a fine-tuned 3D CNN model and encoded further using self-attention. TELNet uses the encoded features to link the semantically related shots and identify scene boundaries when there are no overhead links across consecutive shots..
TELNet was trained end-to-end on multiple video scene identification datasets, archives result on par with other state-of-the-art models in canonical setting, improves result in transfer setting significantly (F-score is doubled in the majority of cross-dataset evaluations), making the proposed model applicable to various video categories. Furthermore, TELNet's computational complexity grows linearly as the number of shots increases in a video; hence, it is efficient for scene boundary identification of long videos.

INDEX TERMS Enter key words or phrases in alphabetical order, separated by commas. For a list of suggested keywords, send a blank e-mail to keywords@ieee.org or visit http://www.ieee.org/organizations/pubs/ani_prod/keywrd98.txt

I. INTRODUCTION

VIDEO Scene boundary identification or video chaptering is a fundamental task in video structure analysis that facilitates extracting information from videos and enhancing user experience while browsing videos [1] [2]. It has shown that effective video temporal segmentation is as essential as predicting frame-level-importance-score in generating video summaries [3].

Considering an unbroken sequence of consecutive frames that are visually similar, videos can be split into shots – a sequence of frames recorded from a single camera. However, semantically meaningful storylines, scenes, or chapters that can be effectively used in information retrieval or video summarization can be a collection of consecutive video shots that may be visually different - recorded from different cameras and/or from different angles [4].

Various researchers have studied shot detection algorithms, and there are multiple established algorithms for shot detection [5] [6] [7] Given the similarity of frames within each shot, studying shallow visual features like color histograms or tracking changes in mutual information of consecutive frames can deliver results on par with similar studies of deep features extracted using object classification or motion detection models. [8].

Understanding videos at a high-level and detecting video scenes as a combination of semantically related shots is a challenging task that requires the integration of different information modalities – feature fusion [9]. Therefore, the majority of researchers have attempted to select a subset of features (such as texture, objects, motion, text, or voice) and learn the association between these features based on specific video content categories (e.g., surveillance videos, human activity [10]). Moreover, despite the great body of research in this area, there is no common framework for high-level temporal segmentation of generic video content – Scene boundary identification.

Having video shots and focusing on visual features, we propose a novel end-to-end Transformer Encoding Linker network (TELNet) that learns associations among shot representations, establishes links among correlated shots, and identifies scene boundaries based on the links (See Figure 1). The idea is to have a model which makes intra-scene shot features and shot features that belong to different scenes more distinguishable through feature encoding and generate a graph of shot links for scene boundary identification. TELNet relies on a pre-trained and fine tuned 3D CNN model for extracting video shot features and a stack of multi-head self-attention networks for learning associations among the extracted shot features – Transformer Encoder. The linker network establishes links among shots based on the encoded shot features and creates a graph of shots(nodes) and their edges(links). Scene boundaries are declared where no overhead links exist between consecutive shots.

Assuming that each scene can be represented by a key-shot which is the shot closest to the mean of shot features within each scene, video graphs are created and used as a label for the Linker network. Furthermore, TELNet was trained using a moving window technique, scanning sequences of shots in steps and aggregating results when reaching the end of the video.

TELNet was trained on multiple publicly available datasets, and compared to other state-of-the-art models [11] [9] [12] [13] [14] [15]. TELNet achieved results comparable to the other SOTA models in the canonical settings and improved results significantly in the transfer setting (the F-score is doubled in the majority of cross-dataset evaluations). To summarize, our main contributions are as follows:

- We proposed a Transformer Encoding Linker Network (TELNet) that models video shot correlations and identifies video scene boundaries without needing prior knowledge of video structure, such as the number of video scenes. The significant improvement of the result in the transfer setting confirms that the selected network models the video structure efficiently for video scene identification.
- The model was trained using a novel labeling methodology that uses links among intra-scene key-shot and shots as the label, and furthermore, process videos using a moving window methodology – process sequences of shots in steps. The model was trained end-to-end compared to the other SOTA models which were trained in increments.
- Given the simplicity of the model architecture, TELNets' computational complexity grows linearly as the number of shots increases; hence it is applicable to long videos in contrast to other models whose complexity grows linearly to the square of the number of shots [14] or NP-hard complexity of Ncut algorithm with an estimate of the number of scenes in a video [15].

II. RELATED WORK

A. SCENE BOUNDARY DETECTION

Scenes as a higher-level unit can be formed based on relationships among shots representations [16]. Rui et al. defined keyframe for each shot as the frame that can represent the shot's salient information [4]. Selected keyframes of different shots can be clustered for identifying scene boundaries. Chasanis et al. [17] used K-Means clustering to group the shot keyframes into scenes. Haroon et al. [18] used Scaled-Invariant Feature Transform (SIFT) [19] descriptor to represent the keyframe of the shot. Scene boundaries are detected by comparing the difference of those descriptors. Sidiropoulos et al. proposed a graph-based clustering - Scene Transition Graph (STG) - which links the shots based on the selected distance measure [20] and identified scene boundaries if there are no links within specific number of shots. Trojahn et al. [12] used VKFrameS^2 key-frames selection algorithm [21] to determine key-frame from shot. Every key-frame are represented by multiple features including visual, audio and textual. Those features are fed toward the LSTM model to further decide if a shot is boundary or not.

B. SHOT REPRESENTATION

Pre-trained image classification CNNs have been widely used for extracting image features. Baraldi et al. [13] extracted visual features within a shot using CNN, trained on ImageNet dataset [22] and Places dataset [23] , Protasov et al. extracted visual features using a pre-trained CNN on the Place205 dataset [24]. Rotman et al. used Inception-v3 to extract the visual features [25] [14].

It has been shown that 3D CNNs can represent transitions in a set of consecutive frames and hence, 3D CNNs (like C3D) perform well in motion detection/classification by tracking what appears in the first few frames (objects) of a sequence [26] [27]. Carreira el at. [28] proposed a 3D CNN architecture based-on Inception-v1 called I3D. Hara el at. In [29] found residual architecture useful on 3D CNN model and proposed a new 3D CNN model based on Resnet. Liu et al. used Resnet34 that was pre-trained on ImageNet and inflated into 3D and then further trained on Kinetic dataset for extracting shot features [15].

C. SHOT CLUSTERING

With the encoded features for each shot, we can cluster the shots into scenes and identify the scene boundaries. Baraldi et al. represented shots by concatenating the extracted visual features with the text features extracted using word2vec from the video transcripts [13]. They proposed the deep Siamese network which evaluates the similarity among shots. The deep Siamese network was trained to have a higher score if two shots come from the same scene. Once the deep Siamese network trained, the scene boundary was determined using the local minimum similarity score. [11] use perceptual features e.g. Visual appearance, audio feature, quantity of speech and time to represent shot. Triplet loss embedding network is employed to make those features more adaptive.

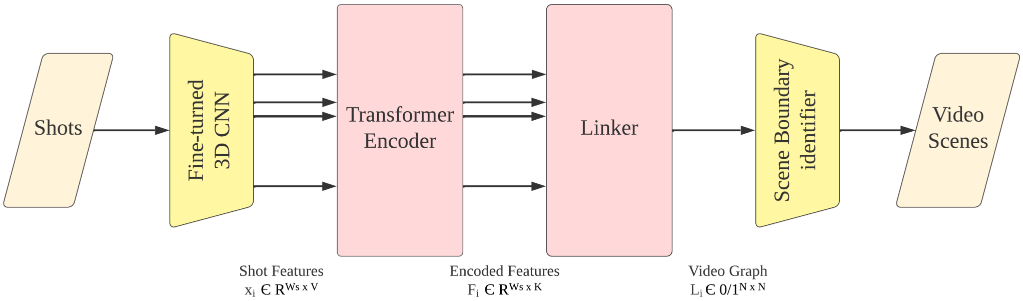
Shots
Fine-turned 3D CNN
Transformer Encoder
Linker
Scene Boundary identifier
Video Scenes
Shot Features $x_i \in R^{Ws \times V}$
Encoded Features $F_i \in R^{Ws \times K}$
Video Graph $L_i \in 0/1^{N \times N}$

FIGURE 1. TELNet overall architecture

	Shot feature	Feature Encoding	Shot clustering	Complexity	Prior Knowledge
[20]	Color Histograms + Audio		Scene Transition Graph(STG)		No required
[11]Triplet	2D CNN + MFCC audio + textual	DNN with triplet loss	Temporal Aware Clustering with minimized Total Within group Sum of Square	Cube to the number of shots	No required
[9]	Audio + visual		Off-the shelf STG etc.		No required
[12]	CNN, textual and audio		LSTM		No required
[13]SDN	2D CNN + textual		Siamese Network		No required
[24]SAK-18	2D CNN + textual		Overlapping Link		No require
[25]OSG	2D CNN + audio		Optimal Sequence Grouping(OSG)	square to the number of shots	number of scenes need to know/estimate
[14]OSG-Triplet	2D CNN	triplet loss	OSG	square to the number of shots	number of scenes need to know/estimate
[15]ACRNet	3D CNN	Self-Attention	Normalized Cuts (NCuts)	NP-hard	Number of scenes need to know/estimate
TELNet*	3D CNN	Transformer Encoding Linker		Linear to the number of shots	No required

TABLE 1. Comparison of state-of-the-art and our model

Shots are clustered thus minimizing the total within-group sum of squares(TWSS).

Protasov el at. linked the shots if their similarity of deep features is higher than a specific threshold.A boundary is declared when the two consecutive shots are not linked [24].

Rotman et al. [25] [14] proposed Optimal Sequential Grouping (OSG) for grouping shots. Shots were clustered by combining visual features with the audio features extracted from VGGish, minimizing intra-scene distance. Having the number of scenes, OSG clusters the shots using dynamic programming.

Liu et al. proposed an adaptive context reading network (ACRNet) model based on Transformer [15]. They selected each shot as the query and the other shots in a specific range as key-value pair. Using Q-function [30], the number of scenes were estimated and then the normalized-cuts (NCuts) [31] was used for shots clustering.

III. METHOD

A. SHOT FEATURES

Three-dimensional convolutional neural networks (3D CNN) can track transitions in consecutive frames; hence, they have been used widely for action recognition tasks [26] [27]. Given that the scenes are defined as semantically related shots that may not be visually similar, we've employed a pre-trained C3D network for representing video shots [27]. C3D model was trained on the Sports-1M dataset [32] which is a well-established sports action recognition dataset. Following the suggestion given by [28], each shot was sampled into 16 frames fulfilling the requirement of C3D model and represented by features extracted from the last layer of C3D, removing the softmax layer [27].

B. TRANSFORMER ENCODER

Shot features that are extracted from the pre-trained 3D CNN are considered raw representations which capture transitions in consecutive frames. Shot representations are then transformed into encoded features based on the desired labels (Links among shots). The idea is to learn the relationship among shot features and transform features to be tuned for the scene boundary identification task. We've employed Transformer encode as a combination of multi-head stacked self-attention networks as the feature encoder given the success Transformers had in sequence modeling [33]. Considering a video with n shots and shots represented by d features, X is denoted as shots feature matrix where X∈ $R^{n\times d}$. Each attention head transforms input features using $W_{Qi} \in R^{d\times dk}$, $W_{Ki} \in R^{d\times dk}$, $W_{Vi} \in R^{d\times dk}$ respectively. Attention weight for i-th head $e_i \in R^{n\times n}$ can be written as (1). Each attention head output denoted as $head_i$ is the softmax of e_i multiplied by XW_v (2). Multi-head output Z is the weighted sum of concatenated outputs of attention heads(3). Normalized residual output Z' is the multi-head output Z added to the original features X. Z' then passed through a feed forward network with weight W_{FF1} and W_{FF2}. Each stack of multi-head attention layer output denoted as Y is the normalized feed forward output plus Z' (4). Employing multiple layers of self-attention networks (Stacks), the final output is the encoded features denoted by F.

$$e_i = [(XW_{Qi})(XW_{Ki})^T] \tag{1}$$

$$head_i = softmax(e_i)V_i, where V_i = XW_{Vi} \tag{2}$$

$$Z = concat([head_1 head_2 head_3 ... head_h])W_O \tag{3}$$

$$Y = norm(Z' W_{FF1} W_{FF2} + Z') \tag{4}$$

C. LINKER

Having the encoded shot features, Linker establishes links among shots and generates a video graph. A video graph is a complete graph in which all nodes (shots) are linked through weighted edges, and the role of the Linker is to predict the edge weights. The Linker is constructed with fully connected layers, denoted by L function which takes S_j, the source shot encoded features located in the j-th row of F, θ, model parameters and predict the pairwise linking probability $Y_j = L(S_j, F, \theta)$ denoted as the linking probability of the j-th shot to other shots in F.

D. MERGE ALGORITHM

Because of the limited size of the computing memory, we set the input window size as 15. However, it leads to the result of fixed boundary of every 15 shots. In order to eliminate the boundaries of every input window, we came up with a merge algorithm using sliding-window method. By selecting stride step as 10, we overlap the last 5 shots from the current window with the first 5 shots from the succeeding window, which we believe are the most possible shots that may link to shots in different window. We combine the top five key shot candidates from both window, by linking the overlapping shots to their most related key shot from both windows, we therefore eliminate the fixed boundaries of every window.

E. LABEL GENERATION

Encoded shot features are expected to be related within each scene and therefore feature encoding should be able to encode shot features to have closer intra-scene distance in comparison to out of scene distance. Selecting the mean of the encoded features within a scene as a key representation of the scene (1-Mean Clustering), we have selected the closest shot to the encoded feature mean as the key-shot within each scene. Algorithm III-E describe the label generation procedure. Once the key-shot is selected, all shots within the same scene will be linked to key-shot. Figure 2 shows a sample of a key shot and how it is linked to the shots within the same scene.

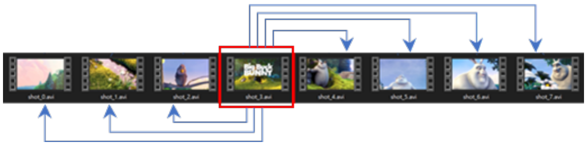

FIGURE 2. Training label generation; This diagram represents a sample scene in which the key-shot (red rectangle) is linked to all the other shots within the scene.

[tb] Label Generation **Input**: Encoded features in a scene **Output**: KeyShot

```
Let a scene start with k-th shot and n shots in that scene
Mean encoded feature $f_{mean} = \frac{1}{n}\sum_{i=k}^{k+n-1} f_i$
$miniDist = \infty$
for i = k to k+n-1 do
    Dist = EuclideanDistance($f_{mean}, f_i$)
    if Dist < minDist then
        minDist = Dist.
        KeyShot = $i$
    end if
end for
return KeyShot
```

IV. IMPLEMENTATION DETAIL AND EXPERIMENT RESULT

A. MODEL SETTING

Extracted shot features, $X \in R^{n\times 4096}$ are encoded to $F \in R^{n\times 4096}$ using a four-head, six stacks of self-attention networks - N is the number of shots in a video. The linker is implemented using a 3-layer fully connected neural network as detailed in Table 2.

B. MODEL EVALUATION METRIC

F-score calculated based on Overflow and Coverage has been widely used for evaluating the performance of the scene boundary identification models [34]. Having the predicted scenes Z = {Z1, Z2, Z3...Zm'} and the ground truth scenes Ž = {Ž1, Ž2, Ž3...Žm}, where Zi, Žj denote the scene index, i ∈ {1, . . ., m'}, and j ∈ {1, . . ., m}. #(Zi) is the number of shots in the predicted i-th scene. The metrics Coverage, Ct and Overflow, Ot for scene Žt are calculated as (5) and (6). F-score is the harmonic mean of Ct and $1 - O_t$ and given by (7)

$$C_t = \frac{Z_{i=1,...m}\#(Z_i \cap {}_t)}{\#({}_t)} \tag{5}$$

$$O_t = \frac{\sum_{i=1}^{m} \#(Z_i \setminus {}_t) * min(1, \#(Z_i \cap {}_t))}{\#({}_{t-1}) + \#({}_{t+1})} \tag{6}$$

$$F - score = \frac{2}{\frac{1}{C_t} + \frac{1}{1-O_t}} \tag{7}$$

Transformer Encoder	Head=4, number of stacks=6, d model=4096
Fully connect layer 1	(4096,2048), activate function=relu
Fully connect layer 2	(2048,1024), activate function=relu
Fully connect layer 3	(1024, *Ws*)

TABLE 2. Transformer Encoding Linker setting, d model is the output dimension of Transformer Encoder

Video Name	Video Length ($hh{:}mm{:}ss$)	Number of Shots	Number of Scenes	Average Scene Length ($mm{:}ss$)
From Pole to Pole	00:49:15	445	23	1:04
Mountains	00:48:05	383	36	1:05
Ice Worlds	00:49:17	421	33	1:01
Great Plains	00:49:03	472	30	0:52
Jungle	00:49:14	460	25	0:55
Seasonal Forests	00:49:19	526	33	0:57
Fresh Water	00:49:17	531	37	0:52
Ocean Deep	00:49:14	410	29	1:04
Shallow Seas	00:49:14	336	33	0:51
Caves	00:48:55	374	22	0:55
Deserts	00:49:00	467	26	0:55
Total	08:59:53	4855	327	1:39

TABLE 3. Detail of BBC Planet Earth Dataset

C. TRAINING

Through the training procedure, the Transformer Encoder and the Linker were trained together, minimizing the cross-entropy loss of the predicted linking probabilities and the ground truth label (generated labels) as shown in following equation:

$$Loss = -\sum Y_{Label} log(L(S_j, F, \theta)) \tag{8}$$

Y_{Label} is the one hot vector where 1 indicates the position of the key-shot – all shots within scene should be only connected to the scene key-shot.

Linker follows a moving window technique and slides through the sequence of *Ws* shots and predicts the pairwise linking probability within the range of *Ws*. Window size is selected as 15 after a comprehensive study provided in the supplementary material.

D. DATASET

We train our propose TELNet model on multiple public available datasets.

1) BBC Planet Earth Dataset

BBC Planet Earth dataset includes 11 episodes of BBC planet program, which are all about nature.Table 3 shows the details of the BBC Planet Earth dataset.

2) OVSD Dataset

Open Video Scene Detection (OVSD) dataset, proposed by Rotman et al., contains several movies and animations [25]. OVSD dataset originally had 21 videos, but we are not able to use all of them due to copyright problems. Table 4 shows the details of the OVSD videos we used in our study. OVSD dataset provides scene boundaries but shot boundaries were not provided.

Video Name	Video Length ($hh{:}mm{:}ss$)	Number of Shots	Number of Scenes	Average Scene Length ($mm{:}ss$)
BBB	00:09:56	113	15	0:40
BWNS	01:09:46	257	36	1:56
CL	00:12:10	98	7	0:53
FBW	01:16:06	686	62	1:13
Honey	01:26:49	315	20	4:08
Meridian	00:11:58	56	9	1:52
Route 66	01:43:25	701	55	1:53
Star Wreck	01:43:14	1055	55	1:53
Total	07:53:24	4250	259	1:50

TABLE 4. Detail of OVSD selected videos

3) MSC Dataset

Movie SceneClip (MSC) dataset is available through a YouTube Channel which contains 500+ videos [15]. Due to copyright problems, we only used 468 videos in our study. Unlike BBC Planet Earth or OVSD dataset, MSC dataset does not have complete videos, and each video is a highlight of a movie. Following [15], we integrate videos from the same movie into a video with pseudo scene boundaries (See Table 5)

Number of Video	Total Shot	Average Video Length	Average number of shots
468	16131	2:35	34

TABLE 5. Detail of MSC Dataset

E. PERFORMANCE COMPARISON

We compared TELNet's performance with the state-of-the-art methods that is detailed in Table 1 [13] [24] [25] [14] [15]. SDN applies the Siamese Neural Network on combination of visual and textual features [13]. SAK-18 estimates scene boundaries based on place change [24]. OSG detects scene boundaries using Optimal Sequential Grouping Algorithm [25]. OSG-Triplet further embeds original features using triplet embedding and applies OSG to detect the scene boundaries [14]. ACRNet [15] embeds features using a transformer and groups the shots using NCut algorithm [31].

Train \ Test	MSC	BBC	OVSD
MSC	0.67	0.64	**0.63**
BBC	0.28	**0.76**	0.22
OVSD	0.29	0.23	**0.73**

TABLE 6. F-score of ACRNet [15] on cross dataset

Train \ Test	MSC	BBC	OVSD
MSC	**0.69**	0.62	0.6
BBC	**0.64**	0.74	**0.56**
OVSD	**0.64**	**0.64**	0.72

TABLE 7. F-score of Proposed TELNet on cross dataset

Table 6 and Table 7 compared TELNet to ACRNet model [15] in transfer settings -cross-dataset evaluation. Note that the F-score on the BBC dataset and the OVSD dataset are based on leave-one-out training. For the MSC dataset, 70% of the videos were used for training and 30% were used for testing.

Table 8 shows the result on the BBC planet dataset. As shown in the Table 8, TELNet outperforms all models by relatively a large margin and achieves results on par with ACRNet [15].

Table 9 shows the result on the OVSD dataset. The F-score of OSG and OSG-triplet come from [25] [14].

TELNet achieves results on par with ACRNet [15] in the canonical setting and outperform other ACRNet significantly in transfer setting. Given the diversity of video categories among the dataset used, TELNet result in transfer setting verifies the effectiveness of the proposed model in learning video structure for scene boundary identification.

V. CONCLUSION

We've proposed a model integrating a Transformer Encoder and a Linker that are trained together to identify video scene boundaries. The proposed model relies on pre-trained 3D CNN models for shot representation and a link graph as the output - label. The link graph was created using a novel technique connecting intra-scene shots and the scene key-shot.

The proposed model was evaluated on various datasets, achieved results on par with the other state-of-the-art models in the canonical setting, and outperformed other models significantly in the transfer settings. The comparable results in the canonical setting were achieved without needing prior knowledge of the video structure, such as the number of scenes in a video. Furthermore, the significant improvement of the results in the transfer settings verifies the efficiency of the model in learning video structure for scene boundary identification.

TELNets' computational complexity grows linearly as the number of shots increases making it the simplest among the state-of-the-art models and applicable to long videos.

	[11] Triplet	[9]	[12]	[13] SDN $*1$	[24] SAK-18 $*2$	[25] OSG $*2*3$	[14] OSG-Triplet $*2*3*5$	[14] OSG-Triplet $*2*4*5$	[15] ACRNet $*1*5$	TELNet $*5$
B01	0.72	0.65	0.63	0.56	0.5	0.66	0.68	0.27	**0.83**	0.77
B02	0.75	0.65	0.65	0.63	0.54	0.65	0.65	0.31	**0.82**	0.68
B03	0.73	0.66	0.64	0.66	0.5	0.64	0.64	0.15	**0.77**	0.69
B04	0.63	0.7	0.68	0.61	0.54	0.6	0.6	0.29	0.72	**0.75**
B05	0.62	0.67	0.63	0.55	0.51	0.56	0.55	0.20	0.7	**0.74**
B06	0.65	0.69	0.64	0.64	0.51	0.58	0.61	0.29	0.7	**0.75**
B07	0.67	0.67	0.66	0.59	0.53	0.54	0.56	0.27	0.7	**0.74**
B08	0.65	0.64	0.67	0.64	0.38	0.65	0.66	0.3	0.73	**0.76**
B09	0.74	0.69	0.64	0.64	0.55	0.57	0.56	0.19	**0.8**	0.7
B10	0.62	0.65	0.67	0.64	0.43	0.59	0.61	0.19	0.75	**0.77**
B11	0.62	0.69	0.66	0.64	0.51	0.65	0.65	0.28	0.71	**0.77**
Average	0.67	0.67	0654	0.62	0.5	0.608	0.62	0.25	**0.76**	0.74

TABLE 8. F-score comparison in BBC Planet Earth dataset. The episodes B01 to B11 are respectively From Pole to Pole, Mountains, Fresh Water,Caves, Deserts, Ice Worlds, Great Plains, Jungles, Shallow Seas, Seasonal Forests and Ocean Deep.
*1 Result obtained from the published paper
*2 Result obtained from their soruce code
*3 OSG and OSG-Triplet are given correct number of scenes
*4 Number of video scenes are estimated
*5 Leave-one-out training result

Video Name	[12]^{*1}	[15] ACRNet^{*1*2}	[14]OSG-Triplet^{*1}	[25]OSG^{*1}	Proposed model TELNet
BBB	0.57	0.74	0.81	**0.83**	0.69
BWNS	0.53		**0.75**	0.63	0.6
CL	0.64	0.61	0.49	0.62	**0.88**
FBW	0.57		**0.76**	0.57	0.66
Honey	0.6		0.73	0.58	**0.77**
Meridian	0.45		0.69	0.63	**0.75**
LCDUP	0.63		0.72	0.73	**0.76**
Route 66	0.63	**0.72**	0.54		0.64
Star Wreck	0.62		0.66	0.55	**0.71**
Average	0.58^{*3}	**0.73**^{*4}	0.701	0.61	0.72

TABLE 9. F-score comparison in OVSD dataset
*1 Results obtained from the published paper
*2 Leave-one-out training result
*3 Average Fscore of the selected video
*4 From [15]

REFERENCES

[1] Hongcheng Wang, Jan Neumann, and Jonghyun Choi. Determining video highlights and chaptering, November 9 2021. US Patent 11,172,272.

[2] Amol Jindal and Ajay Bedi. Extracting session information from video content to facilitate seeking, June 30 2020. US Patent 10,701,434.

[3] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. Rethinking the evaluation of video summaries. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7596–7604, 2019.

[4] Yong Rui, Thomas S Huang, and Sharad Mehrotra. Constructing table-of-content for videos. Multimedia systems, 7(5):359–368, 1999.

[5] Costas Cotsaces, Nikos Nikolaidis, and Ioannis Pitas. Video shot detection and condensed representation. a review. IEEE signal processing magazine, 23(2):28–37, 2006.

[6] Sadiq H Abdulhussain, Abd Rahman Ramli, M Iqbal Saripan, Basheera M Mahmmod, Syed Abdul Rahman Al-Haddad, and Wissam A Jassim. Methods and challenges in shot boundary detection: a review. Entropy, 20(4):214, 2018.

[7] Shrikant Chavate, Ravi Mishra, and Pranay Yadav. A comparative analysis of video shot boundary detection using different approaches. In 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), pages 1–7. IEEE, 2021.

[8] Gautam Pal, Dwijen Rudrapaul, Suvojit Acharjee, Ruben Ray, Sayan Chakraborty, and Nilanjan Dey. Video shot boundary detection: a review. In Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2, pages 119–127. Springer, 2015.

[9] Rodrigo Mitsuo Kishi, Tiago Henrique Trojahn, and Rudinei Goularte. Correlation based feature fusion for the temporal video scene segmentation task. Multimedia Tools and Applications, 78(11):15623–15646, 2019.

[10] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the ieee conference on computer vision and pattern recognition, pages 961–970, 2015.

[11] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Recognizing and presenting the storytelling video structure with deep multimodal networks. IEEE Transactions on Multimedia, 19(5):955–968, 2016.

[12] Tiago Henrique Trojahn and Rudinei Goularte. Temporal video scene segmentation using deep-learning. Multimedia Tools and Applications, 80(12):17487–17513, 2021.

[13] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In Proceedings of the 23rd ACM international conference on Multimedia, pages 1199–1202, 2015.

[14] Daniel Rotman, Yevgeny Yaroker, Elad Amrani, Udi Barzelay, and Rami Ben-Ari. Learnable optimal sequential grouping for video scene detection. In Proceedings of the 28th ACM International Conference on Multimedia, pages 1958–1966, 2020.

[15] Dong Liu, Nagendra Kamath, Subhabrata Bhattacharya, and Rohit Puri. Adaptive context reading network for movie scene detection. IEEE Transactions on Circuits and Systems for Video Technology, 31(9):3559–3574, 2020.

[16] Alan Hanjalic, Reginald L Lagendijk, and Jan Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. IEEE transactions on circuits and systems for video technology, 9(4):580–588, 1999.

[17] Vasileios T Chasanis, Aristidis C Likas, and Nikolaos P Galatsanos. Scene detection in videos using shot clustering and sequence alignment. IEEE transactions on multimedia, 11(1):89–100, 2008.

[18] Muhammad Haroon, Junaid Baber, Ihsan Ullah, Sher Muhammad Daudpota, Maheen Bakhtyar, and Varsha Devi. Video scene detection using compact bag of visual word models. Advances in Multimedia, 2018, 2018.

[19] David G Lowe. Object recognition from local scale-invariant features. In Proceedings of the seventh IEEE international conference on computer vision, volume 2, pages 1150–1157. Ieee, 1999.

[20] Panagiotis Sidiropoulos, Vasileios Mezaris, Ioannis Kompatsiaris, Hugo Meinedo, Miguel Bugalho, and Isabel Trancoso. Temporal video segmentation to scenes using high-level audiovisual features. IEEE Transactions on Circuits and Systems for Video Technology, 21(8):1163–1177, 2011.

[21] Tiago H Trojahn, Rodrigo M Kishi, and Rudinei Goularte. A new multimodal deep-learning model to video scene segmentation. In Proceedings of the 24th Brazilian Symposium on Multimedia and the Web, pages 205–212, 2018.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

[23] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. Advances in neural information processing systems, 27, 2014.

[24] Stanislav Protasov, Adil Mehmood Khan, Konstantin Sozykin, and Muhammad Ahmad. Using deep features for video scene detection and annotation. Signal, Image and Video Processing, 12(5):991–999, 2018.

[25] Daniel Rotman, Dror Porat, and Gal Ashour. Robust video scene detection using multimodal fusion of optimally grouped features. In 2017 IEEE 19th international workshop on multimedia signal processing (MMSP), pages 1–6. IEEE, 2017.

[26] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence, 35(1):221–231, 2012.

[27] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 4489–4497, 2015.

[28] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6299–6308, 2017.

[29] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 3154–3160, 2017.

[30] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graphs. In Proceedings of the 2005 SIAM international conference on data mining, pages 274–285. SIAM, 2005.

[31] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence, 22(8):888–905, 2000.

[32] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1725–1732, 2014.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[34] Jeroen Vendrig and Marcel Worring. Systematic evaluation of logical story unit segmentation. IEEE Transactions on Multimedia, 4(4):492–499, 2002.