Jointly Learning the Attributes and Composition of Shots for Boundary Detection in Videos

Xuekun Jiang, Libiao Jin, Anyi Rao*, Linning Xu, and Dahua Lin

***Abstract*—In film making, shot has a profound influence on how the movie content is delivered and how the audiences are echoed, where different emotions and contents can be delivered through well-designed camera movements or shot editing. Therefore, in pursuit of high-level understanding of long videos, accurate shot detection from untrimmed videos should be considered as the first and the most fundamental step. Existing approaches address this problem based on the visual differences and content transitions between consecutive frames, while ignoring intrinsic shot attributes, *viz.*, camera movements, scales, and viewing angles, which essentially reveal how each shot is created. In this work, we propose a new learning framework (SCTSNet) for shot boundary detection by jointly recognizing the attributes and composition of shots in videos. To facilitate the analysis of shots and the evaluation of shot detection models, we collect a large-scale shot boundary dataset *MovieShots2*, which contains $15K$ shots from 282 movie clips. It is richly annotated with the temporal boundary between consecutive shots and individual shot attributes, including camera movements, scales, and viewing angles, which are the three most distinct shot attributes. Our experiments show that the joint learning framework can significantly boost the boundary detection performance, surpassing the previous scores by a large margin. SCTSNet improves shot boundary detection AP from 0.65 to 0.77, pushing the performance to a new level.**

***Index Terms*—Shot type; boundary detection; cinematic style.**

I. Introduction

The storytelling of a movie is heavily determined by its filming and editing style, where a variety of elements in film making are consolidated. As the basic unit of movie construction, shot, which is represented by a series of image frames that are recorded by a camera at certain times, plays an important role in delivering the underlying stories.

The ability to detect individual shots from untrimmed long videos is the first and most important step towards understanding movies and appreciating their artistic styles. However, existing approaches take a simple assumption that the shot continuity is equivalent to the frame's visual continuity. They design hand-crafted features [1]–[4] or take advantage of deep learning features [5], [6] to detect pixel-level changes and obtain shots. While these low-level bottom-up features could handle simple cases where visual information is continuous, they fail in complex situations, as this assumption on the equivalence between shot continuity and visual continuity does not hold in many real cases. For example, when the camera is occluded by an object, the visual continuity is broken while the shot should not be treated as disrupted; Also, when shots "dissolve", the adjacent shots are usually overlapped in the transition moments. Although these shots may share certain visual continuity, they still should be taken as two different shots. In both cases, visual continuity and shot continuity cannot directly imply each other. Using low-level appearance information among the pixels can instead severely hamper the performance of successful shot detection.

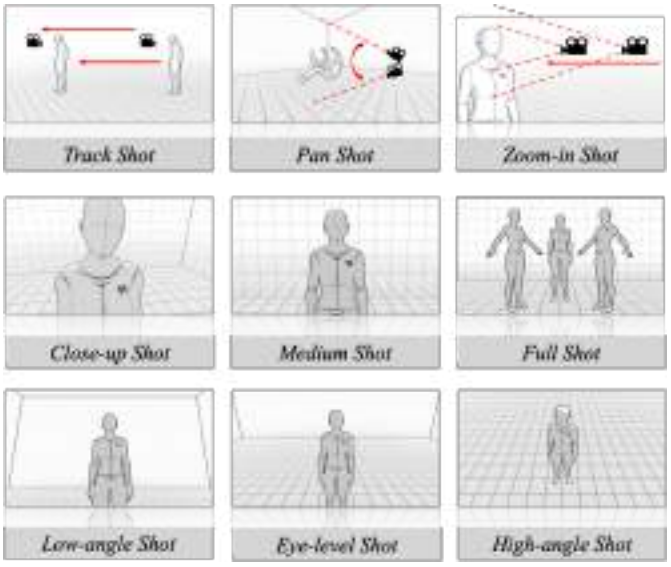

Fig. 1. Prototype demonstration of different shots in terms of camera movement, scale, and viewing angles, on nine selective categories.

To seek the answer of what determines the boundary of two shots, we resort to professional filming theory in cinematographic art [7] and start from the birthplace of the shot boundary. We find out that filmmakers follow certain editing theories, such as time/space ellipsis, match, montage, to connect different shots. The adjacent shots usually vary in attributes such as *camera movement*, *scale*, and *viewing angle*, as illustrated in Figure 1.

Inspired by this observation, we take the advantage of these intrinsic shot features to aid shot detection from videos. We propose a new framework to accurately detect shot boundary by jointly learning the attributes and composition of shots, believing that the understanding of shot attributes will contribute to the shot boundary detection. A preliminary version of this article has appeared in [8]. In that work, we collected a dataset *MovieShots* for shot type recognition and designed a

Xuekun Jiang and Libiao Jin are with the State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, China e-mail: ({xkjiang, libiao}@cuc.edu.cn).

*Anyi Rao (corresponding author), Linning Xu and Dahua Lin are with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China e-mail: ({ayrao, linningxu, dhlin}@ie.cuhk.edu.hk).

subject guidance network to classify shot movement type and scale type. In this paper, our extensions include: 1) SCTSNet, a novel joint classification and temporal segmentation network for shot boundary detection; and 2) a large-scale *MovieShots2* dataset, which extends the original *MovieShots* dataset [8] to a video *scene* setting, where each shot is annotated under the content of their affiliated movie scenes, thus individual shot can also be jointly studied with its adjacent shots within the same story plot. We provide accurate shot boundary annotation and introduce more shot attributes annotation including shot camera movement, scale, and viewing angle. The experiments show that SCTSNet greatly improves the shot detection performance comparing to existing methods [9]–[13] and can handle more complicated scenarios with ease.

The rest of this paper is organized as follows: a brief background on related work about shot boundary detection is provided in section II. Section III introduces *MovieShots2* dataset in details, including its collection and basic statistics. Section IV explains our joint classification and temporal segmentation framework (SCTSNet) in detail. Experiments are conducted to validate our framework in section V. Finally, we provide a conclusion of this work.

II. Related Work

A. Shot Attributes and Categories

Shot is the fundamental unit of video, which is a sequence of continuous images taken from a camera, from the moment that the camera starts rolling until the moment it stops. Most related works on shot studies mainly focus on two shot attributes: scale and movement types. Early works like [1] classified the shot scale type based on human-defined rules based on the ratio of the face box and the height of the frame. Conventional methods for video shot type classification use SVM with low-level texture features, color region, histogram or optical flow [5], [14]–[16]. Wang *et al* [17] redefined seven shot movement types based on camera motion and camera distance, and use probabilistic distance, motion descriptor and attention descriptor to classify them with SVM. Bhattacharya *et al* [18] focused on camera motion. It used homographic and Lie algebra to describe the motion type. With the development of deep learning, some researchers introduced deep learning methods into shot scale classification. Lin *et al* [19] introduced deep neural networks to classified shots in concert videos, extract features from VGG16 net [20].

Inspired from filming theories [7], we study three basic attributes *camera movement*, *shot scale* and *viewing angle*, which describe a shot from different perspectives comprehensively: 1) Camera movement explains how a shot is filmed; 2) Shot scale represents what content a shot has; 3) Viewing angle determines where a shot is viewed from. We include these three attributes in our *MovieShot2* annotation and explicitly model these attributes in SCTSNet to improve the shot representation.

B. Shot Boundary Detection

Early traditional methods are based on hand-craft features to represent visual contents, such as pixel [21], edge [2], texture [3] and color [22]. These features are sensitive to local changes in the frame, such as rapid motion. In order to solve these problems, researchers further proposed a block-based interframe comparison method. Some global features are also used to represent inter-frame differences, such as color histogram [4]. However, the histogram feature is also sensitive to image brightness. Later, local feature descriptors such as SIFT and SURF are applied to shot boundary detection algorithm. In [23], SIFT features of boundary frames were used to identify cut transition and gradual transition. Baber *et al* [24] proposed a shot detection framework based on entropy and SURF. In [25], SURF features, RGB histogram and RBG pixel value are combined to conduct shot boundary detection. In addition to inter-frame differences, some researchers took advantage of the characteristics of continuous frames. Lu *et al* [10] transform the input shot into a matrix, decomposed the shot matrix with SVD, and made the final prediction. Yuan *et al* and Luo *et al* [26], [27] construct a shot graph and detect the shot boundary with graph cut.

With the later development of deep learning, researchers introduce deep neural networks to solve the shot boundary detection problem. Wu *et al* [11] apply 3D convolution network to solve the classification problem of gradient shot. Xu *et al* and Tong *et al* [12], [13] introduced a convolution network to extract video frame features. Most applications of deep networks follow the three steps defined by [26].

Two main disadvantages are noticed in existing works. First, low-level bottom-up features only have poor representation to shot. Second, most of those works didn't consider the temporal feature of a shot. In this work, we propose an end-to-end network that takes in video shots, learns the relationship between adjacent shots in a long-term video clip, and finally outputs the shot boundary predictions. To get better shot representation, we focus on three shot attributes: camera movement, scale, and viewing angle, and introduce a comprehensive network to learn different shot attributes.

C. Video Temporal Feature Extraction

Various video studies have emerged recent years, such as action recognition [28], [29], person search [30], video scene temporal segmentation [31], video caption [32]–[34], and video generation [35]. There are three network structures that are widely used to extract video temporal features in recent studies. Tran *et al* and Carreira *et al* [36], [37] proposed 3D convolution, and indicates that 3D convolution has a better representation for video data. 3D convolution obtains state of the art performance in many video applications, *e.g.*, video description generation [38], action detection [39] and video classification [40]. Other studies used convolution network and Recurrent Neural Network (RNN) to extract video temporal features. The convolution network generated feature vectors for each frame, and then a sequence of feature vectors of each video clip was fed into a temporal model such as LSTM, GRU to make the final prediction. Venugopalan *et al* [41] trained an end-to-end video description model using CNN and LSTM. Lu *et al* [42] used SSD to extract the object of each frame, and applied LSTM to extract the temporal feature. Donahue *et*

Fig. 2. Examples on the five shot scale types.

Fig. 3. Examples on the three viewing angle types.

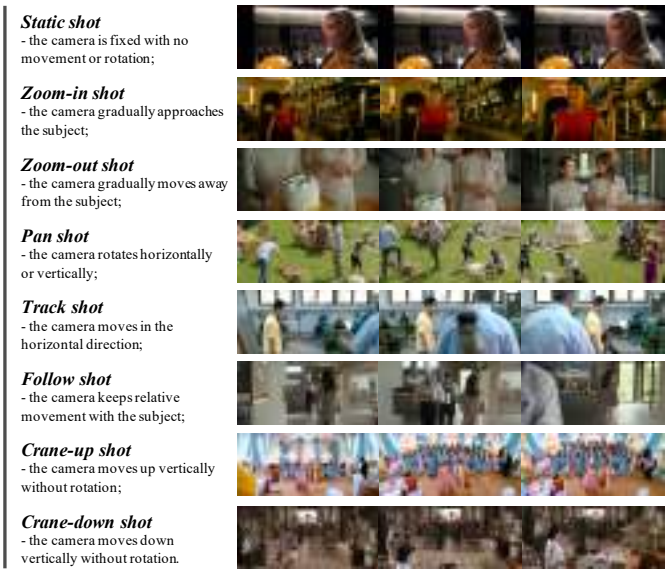

Fig. 4. Examples on the eight shot movement types.

al [43] adopted CNN and LSTM to extract video features. The third method is based on sampling, *e.g.*, TSN [44] extracted temporal feature by slice sampling. In this work, we follow the sampling strategy in TSN [44] to improve computational efficiency. We extract shot features and feed them into a sequence model for the downstream boundary detection task.

III. MovieShots2 Dataset

To facilitate the joint study shot attributes and the composition of shots. We collect *MovieShots2*, a shot boundary detection dataset with richly annotated shot types. *MovieShots2* contains $15,091$ shots collected from 282 movie clips. Each shot has three attributes, they are 1) camera movement, 2) scale, and 3) viewing angle.

To our best knowledge, *MovieShots2* is the first large-scale video shot datatset with complete shot attributes annotation including *movement*, *scale* and *angle*. Compared with our last work *MovieShots* [8], this version has two major improvements: 1) It contains more granular categories from 9 classes to 16 classes. Borrowing the professional domain knowledge from the film industry [7], we add a new shot attribute *angle* and further divide certain categories that have significantly more samples than other categories into more granular categories. Specifically, *motion shot* are specified as *pan*, *follow* and *crane-up* 2) The annotation is based on the shots within a scene context instead of independent shots, which can further support the study of adjacent shots and their joint effects in conveying a complete story scene. The details of this dataset are specified as follows.

A. Dataset Collection

To build such a large-scale shot detection dataset has two main challenges: shot segmentation and shot annotation. Considering that high-quality movies contain rich shot type usage and shot transition scenarios, we firstly collected 15 top-rated movies from IMDB as the pilot trial. However, to annotate a hours long movie is extremely time-consuming. It is even difficult for annotators to keep concentration in the whole process, which will diminish the quality of the annotation. To ease the difficulty, we firstly divided each movie into many minutes-long video clips with a complete story plot according to script and synopsis, and conduct annotation on these short clips. Meanwhile, we provide coarse initial boundaries for annotators using a traditional off-the-shelf shot detection algorithm [9]. This helps the annotators to improve efficiency on the per-frame annotation process. The annotators only need to check each shot, correct the faulty detected boundaries, and mark the boundaries that have not been identified by the shot detection algorithm. The whole annotation process is supervised by professional director and workers in the film academics and industry. Before the formal annotation phase, we designed a test phase that standardized annotators' criteria on each shot type to ensure consistency. Each shot is labeled with three rounds of annotation and the dataset finally reaches a high consistency of 90%.

B. Shot Categories

In this work, we study three shot attributes, namely, the movement, scale, and angle, each with 8, 5, 3 types respectively. See Figure 2, 3, 4 for illustrated examples.

Movement attribute describes the movement state of the camera. Movement attribute can to divided into eight types: 1) In *Static shot*, the camera is fixed with no movement or rotation, driving the audiences' attention to the characters; 2) The camera gradually approaches the subject in *Zoom-in shot*, while (3) gradually moves away from the subject in *Zoom-out shot*; 4) In *Pan shot*, the camera rotates horizontally or vertically, showing the audiences different parts of a large scene; 5) *Track shot* is recorded from a camera moving on a track, where the camera moves in the horizontal direction; 6) *Follow shot*, the camera keeps relative movement with the subject; 7) Crane-up and 8) Crane down shots gradually lift

TABLE I
COMPARISON OF DIFFERENT SHOT BOUNDARY DATASETS

	# Shot	Boundary	Scale	Angle	Movement
Unified 2005 [45]	430		✓		
TRECVID 2007 [46]	2463	✓			
Soccer 2009 [47]	1838		✓		
Taxon 2009 [17]	5054				✓
Content 2011 [14]	3206		✓		
Lie 2014 [18]	327				✓
MovieShots [8]	46857		✓		✓
MovieShots2	15091	✓	✓	✓	✓

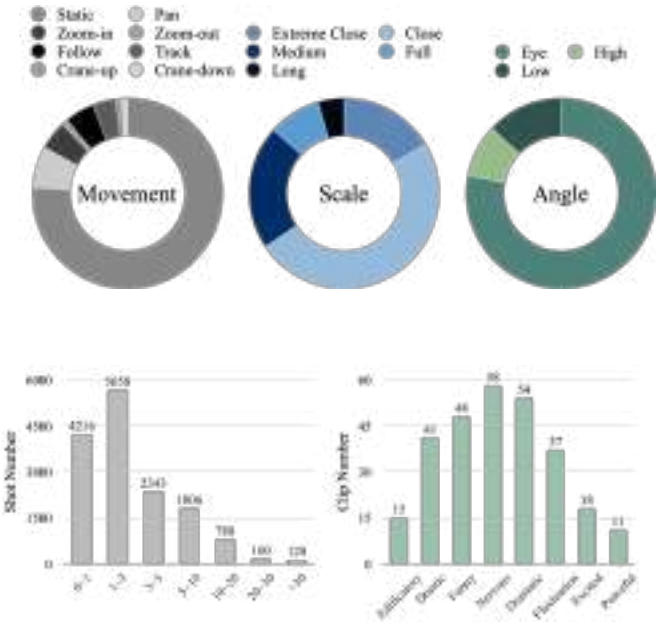

Fig. 5. Statistics of *MovieShots2*. The pie graphs show the distribution of categories within each shot attribute. The histograms show the distribution of shot duration and drama style among 282 movie clips.

up/down to show/leave the whole scene, where the camera is placed on equipment and move upward/downward.

Scale attribute is usually determined by the portion of the subject figure that is included within the frame. It has five types: 1) *Extreme close-up shot* (ECS) shows the details of the object or human body; 2) *Close-up shot* (CS) always be used to show the actor's facial expression; 3) *Medium shot* (MS) contains a figure from the knees or waist up; 4) *Full shot* (FS) barely includes the human body in full; 5) *Long shot* (LS) is taken to show a large space from a long distance

Angle attribute refers to the angle between the camera and the filmed objects. It marks the specific location at which the camera is placed to take a shot with three main types: 1) *Eye-level shot* is taken at the same height as the human eye and there is no obvious emotional tendency; 2) In *High-angle shot* and 3) *Low-angle shot*, the height of the camera is higher/lower than the height of the human eye. While high-angle shot always provides an omniscient view to audiences, low shots usually express strong emotions.

C. Dataset Statistics

The comparison between *MovieShots2* and existing shot datasets [8], [14], [17], [18], [45]–[47] is shown in Table I. [1] We compare against the total number of annotated shots, as well as their supported annotations on different shot attributes.

Note that, the previous dataset *MovieShots* is specifically designed for shot type classification. The data is collected from movie trailers and each shot is independent with each other. *MovieShots2* instead focuses on the video scene setting where each shot is annotated under the content of movie scenes. Therefore, each individual shot can also be jointly studied with its adjacent shots within the same plot.

MovieShots2 is large and comprehensive, containing more than $15k$ shots. It is annotated with detailed shot type categories, *i.e.*, 8 movement types, 5 scale types and 3 angle types. Each shot is connected with a movie clip that may help the study of the relationship between scene and shot. Furthermore, our new dataset is also of great diversity. Figure 5 shows some basic statistics of *MovieShots2*. The shot categories distribution corresponds to the natural distribution in real films and the shot length distribution covers a wider range. Shots coming from 282 movie clips differ in their drama style, *e.g.*, edificatory, drastic, or funny, containing rich shot types and shot transition styles, which is crucial to learn a robust shot detection model.

IV. SHOT CLASSIFICATION AND TEMPORAL SEGMENTATION NETWORK

Inspired from the editing theory that adjacent shots in a video usually belong to different types, we propose a new framework Shot Classification and Temporal Segmentation Network (SCTSNet) for video shot boundary detection. The overall framework is shown in Figure 6. The core idea of our framework is to simultaneously determine the attributes of a shot and the boundary between two shots in a video. SCTSNet is composed of three main parts: 1) multi-attribute feature extractor $\mathcal{E}$, 2) shot types classification network $\mathcal{F}$, and 3) video temporal segmentation network $\mathcal{T}$. Given a video clip, we first perform average sampling to get selective keyframes at a regular interval. The sampled frames are then fed into the extractor to provide shot intrinsic features, *i.e.*, movement, scale and angle with the supervision coming from the shot type classification network. The features coming from a sequence of frames are further fed into a multi-temporal-scale sequential network to predict the shot boundary in a context.

A. Multi-attribute Feature Learning

As we discussed before, the visual appearance change is not the essential reason for the existence of shot boundary. The intrinsic shot features, *viz.*, movement, scale, angle, instead are the keys for a video editor to decide two shots. To overcome the shortcomings of low-level features, we take the three classification supervisions coming from the movement, scale, and angle types to acquire a better shot feature representation.

[1]The comparing datasets are accessed on Oct. 10, 2020.

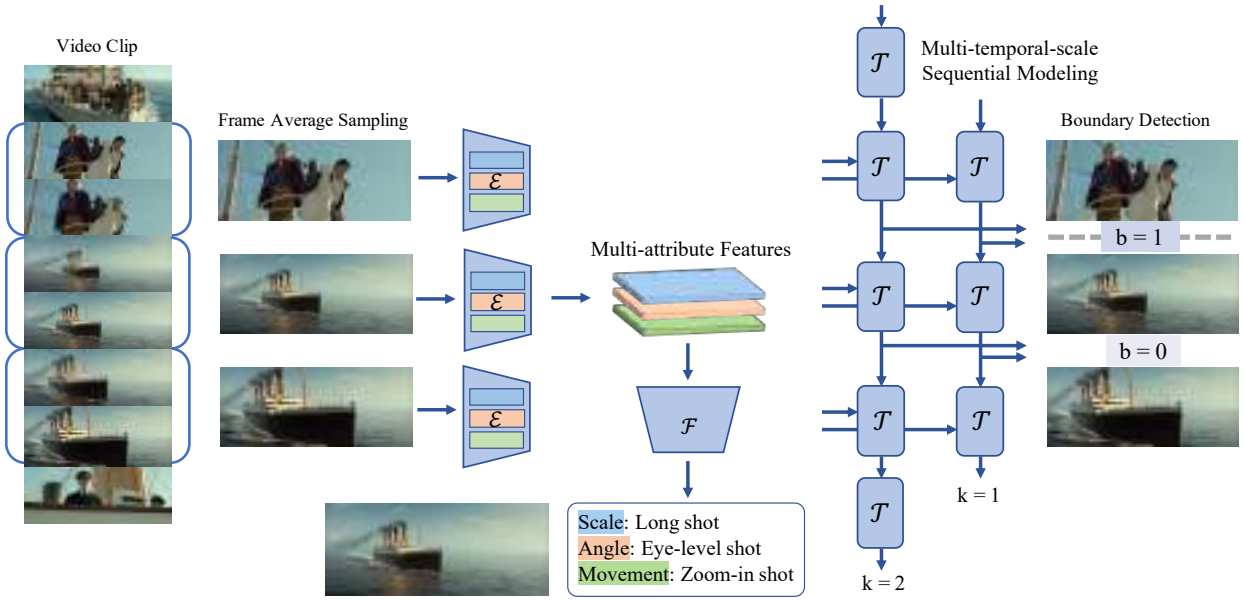

Fig. 6. The pipeline of shot classification and temporal segmentation network (SCTSNet). The number 1 between two clips represents there is a boundary and vice versa.

Given a video, we split it into several clips at regular intervals. The interval is set to be half a second in our experiment. The choice comes from the observation that almost all the shots last for at least one second long, and the attributes and contents of a clip generally do not change much within this short time interval. A dense sampling will not affect performance, but increase the computation cost. The central frame is then selected to represent each individual video clip.

A video with a sequence of N frames is denoted as $V = [I_1, I_2, \cdots, I_t, \cdots, I_N]$. Each frame I_t is fed into three feature extractors to handle movement, scale, and angle attributes respectively. Since the three aspects affect in different perspectives, they are expected to function in an independent fashion. The three feature extractors are trained separately to avoid different attributes to affect each other. Specifically, considering the characteristics of different shot attributes, there are two types of extractors that we used here. Since 3D convolution network can better model temporal features between frames, we use I3D [37] to extract the movement feature, and use ResNet50 [48] as the extractor for scale and angle features. The extractor module $\mathcal{E}$ outputs the features,

$$f_t = \mathcal{E}(I_t) = (f_t^s, f_t^m, f_t^a), \tag{1}$$

for each frame I_t. We use t denotes the frame index, and use subscripts s, m, a to distinguish three shot attributes.

B. Multi-temporal-scale Sequential Modeling

After acquiring the multi-attribute features $\{f_t\}$, we propose a multi-temporal-scale sequential model to handle different shot transition scenarios, *e.g.*, fade-in/fade-out and cut-in/cut-out cases. [2] We detect the boundaries by incorporating more contextual information and fusing different time-range scale information to improve its robustness.

Since a shot boundary connects two adjacent visual frames, the contextual information in the neighboring timestamps are also important to the boundary detection. We apply a sequence-to-sequence model LSTM [49] here to maintain contextual information while keeping the computation efficient. Fix index t and the half window size w, the shot boundary detection problem is now turned into the task of predicting a sequence binary labels through the network $\mathcal{T}$. Formally,

$$\begin{aligned} b_{t,w} &= [b_{t-w}, \cdots, b_t, \cdots, b_{t+w-1}], \\ f_{t,w} &= [f_{t-w}, \cdots, f_t, \cdots, f_{t+w}], \\ b_{t,w} &= \mathcal{T}(f_{t,w}), \end{aligned} \tag{2}$$

where $b_t \in \{0, 1\}$ indicates whether there is a shot boundary (*i.e.*, the shot transits) between the t-th and $(t+1)$-th frame.

Consider that in certain shot transition scenarios, *e.g.*, cut-in/out is abrupt while fade-in/out is smooth and lasts for a longer time, the required time-range reception field is different. The prediction is instantiated with a multi-temporal-scale network, which is composed of multiple sequential models with different reception fields. Each model $\mathcal{T}_{k,w}^i$ receives a specific kind of feature information, *viz*, scale, movement and angle, from a reception field of length $2kw$, where w is the half window size, $1 \le k \le K$, attribute type $i \in \mathcal{I} = \{m, s, a\}$, and outputs the central $2w$ prediction scores as follows,

$$[p_{t-w}^{i,k}, \cdots, p_{t+w-1}^{i,k}] = \mathcal{T}_{kw}^i([f_{t-kw}^i, \cdots, f_{t+kw}^i]), \tag{3}$$

where $p_t^{i,k} \in [0, 1]$ represents the probability of shot boundary existence.

The final prediction of multi-temporal-scale sequential models are assembled as a weighted sum over multi-attribute level

[2] Fade/cut-in/out means that there is a gradual or abrupt frame appearance change between neighboring shots.

and multi-temporal-scale predictions,

$$b_t = \sigma\left(\sum_{i\in\mathcal{I}}\lambda_i\sum_{k=1}^{K}\mu_k p_t^k\right). \quad (4)$$

We choose $K = 2$, $w = 5$ in our experiments, and σ is a binarization function with threshold 0.5. Specifically, μ_k is the weight to ensemble the prediction from different temporal scale model, which is set to 0.9 for the long model and 0.1 for the short one. λ_i is the weight to ensemble the results of different shot attributes, which represents their respective contributions to the boundary detection. The weights of movement, scale and angle are set to 0.7, 0.2, 0.1 respectively.

C. Joint Shot Type Classification and Temporal Segmentation

Using the supervision from shot boundary alone is too weak to support the learning of shot attributes, therefore, we provide additional supervision to learn the shot type classification and temporal segmentation jointly. A shot type classifier head $\mathcal{F}$ is appended after the feature extractor $\mathcal{E}$. The total loss is defined as follow,

$$\mathcal{L} = \alpha\mathcal{L}_{cls} + \beta\mathcal{L}_{seg}, \quad (5)$$

with

$$\begin{aligned}\mathcal{L}_{cls} &= -\sum_{i\in\{m,s,a\}}\sum_{j=1}^{M_i} y_{i,j}log(q_{i,j}),\\ \mathcal{L}_{seg} &= -(ylog(q) + (1-y)log(1-q)),\end{aligned} \quad (6)$$

where y is a binary indicator to show whether the class label is the correct classification, q is the predicted probability, M_i is the number of classes for attribute i, α and β are the weights for the two term respectively.

V. Experiments

A. Experiments Setup

Dataset. All the comparing methods are conducted on our *MovieShots2* dataset and TRECVID 2007 [46]. *MovieShots2* is split into *Train* and *Test* sets with a ratio of 7:3, as shown in Table II. Some basic statistics, *e.g.*, the ground truth shot number, average shot number and average frame number are also presented. As TRECVID 2007 dataset doesn't provide a training set and shot type annotation, it is taken as an additional evaluation set to compare all methods.

Implementation details. We take the cross-entropy loss for the classification. We train these models for 100 epochs with mini-batch SGD, where the batch size is set to 64 and the momentum is set to 0.9. The network weights are initialized with pre-trained models from ImageNet [50]. The initial learning rate of $\mathcal{T}$ is 0.001. The learning rate of $\mathcal{E}$ and $\mathcal{F}$ is 100 times lower than the one of $\mathcal{T}$. All the learning rate will be divided by 10 at the $30th$ and $80th$ epoch. The experiments are trained with 32 Tesla K80 16GB using PyTorch.

Evaluation metrics. We take three commonly used metrics: 1) Average Precision (AP): specifically in our experiment, it is the mean of AP of $b_t = 1$ for each movie clip. 2) Miou: a symmetric measure based on intersection over union to assess the quality of detected shots. 3) Accuracy (Acc): the accuracy of the binary classification of b_t.

TABLE II
Statistics of the *MovieShots2* dataset

	Train	Test	Total
Number of clips	225	57	282
Number of shots	11944	3147	15091
Avg. shots of each clip	53	55	54
Avg. frames of each clip	5652	5040	5532

B. Overall Results

We reproduce existing methods [10]–[12], [26], [51], [52] according to their papers since their codes are not publicly available. PerframeContent [9] is experimented with using their provided original code repository.

a) Analysis of Overall Results: The overall results are shown in Table III. Traditional methods [9], [51], [52] make boundary prediction based on the difference between two adjacent frames and the frames are represented by handcraft features. Additionally, such methods rely on predefined thresholds, making it sensitive to the local changes of pixels within the frames. Although these methods achieve better performance compared to the random baseline in terms of accuracy, they still create a large number of wrong boundaries, which is reflected from the lower recall. To better depict the relationship among frames, some other methods, *e.g.*, Graph and SVD [10], [26] model the frame sequences into a graph or a matrix, but the performances of these methods are still not satisfactory. Deep learning methods [11], [53] use 2D/3D convolution neural networks to represent shot features and detect shot boundary. A better representation ability helps these networks achieve better results than traditional methods.

Our full model SCTSNet (full, mf + mt), explicitly learns the boundary prediction by using the multi-attribute feature and multi-scale-temporal sequence model. It achieves the best result among all the competing methods and improves the AP from 0.65 to 0.77 (relatively 20%) compared to the SOTA methods [11], [53]. Similar conclusion can be made from the results on TRECVID 2007 [46], as shown in Table IV.

b) Analysis of Our Framework: Our base model of SCTSNet (base) uses a single shot attribute extractor (movement) and a single sequential model ($K = 1$). The base model SCTSNet (base) is comparable to traditional methods. As we add multi-attribute feature learning to it, the performance is greatly improved, the AP raises from 0.60 to 0.69 (relatively 15%), and M$_{iou}$ improves from 0.72 to 0.77 (relatively 7%) and Acc increases from 89.67 to 91.89 (relatively 2%). With the help of multi-temporal-scale sequential model, the SCTSNet (+ mt) improves the AP from 0.60 to 0.75 (relatively 25%) and Acc from 89.67 to 93.82 (relatively 5%). Finally, the full model SCTSNet (fill, mf + mt) achieves the best result with higher AP (0.77), M$_{iou}$ (0.78) and accuracy (94.11), which shows the effectiveness of multi-attribute feature learning and multi-scale-temporal designs.

C. Ablation Studies

Four ablation studies on different module designs are presented to show their effectiveness: 1) different attribute learn-

TABLE III
RESULTS ON MOVIESHOTS2. HERE MF MEANS MULTI-ATTRIBUTE FEATURE AND MT MEANS MULTI-TEMPORAL-SCALE SEQUENCE MODEL

Settings	AP	M_{iou}	Acc
Random	0.19	0.30	50.49
Histogram, Boreczky *et al* [51]	0.50	0.71	84.63
RegionHistogram, Boreczky *et al* [51]	0.59	0.70	90.71
Block, Hanjalic *et al* [52]	0.60	0.68	92.00
PerframeContent, Brandon *et al* [9]	0.57	0.71	85.29
Graph, Yuan *et al* [26]	0.17	0.35	78.33
SVD, Lu *et al* [10]	0.27	0.16	87.68
2D-CNN, Zhao *et al* [53]	0.65	0.64	93.86
3D-CNN, Wu *et al* [11]	0.64	0.66	93.38
2D-CNN (+ mf), Zhao *et al* [53]	0.66	0.67	94.11
3D-CNN (+ mf), Wu *et al* [11]	0.68	0.71	94.27
SCTSNet (base)	0.60	0.72	89.67
SCTSNet (+ mf)	0.69	0.77	91.89
SCTSNet (+ mt)	0.75	0.73	93.82
SCTSNet (full, mf + mt)	**0.77**	**0.78**	94.11

TABLE IV
RESULTS ON TRECVID 2007

Settings	AP	M_{iou}	Acc
Histogram, Boreczky *et al* [51]	0.14	0.11	94.12
RegionHistogram, Boreczky *et al* [51]	0.31	0.54	92.83
Block, Hanjalic *et al* [52]	0.42	0.73	**94.57**
PerframeContent, Brandon *et al* [9]	0.39	0.55	87.56
SVD, Lu *et al* [10]	0.18	0.25	94.00
SCTSNet (full, mf + mt)	**0.57**	**0.79**	94.43

TABLE V
THE EFFECTS OF DIFFERENT ATTRIBUTE LEARNING

Move.	Scale	Angle	AP	M_{iou}	Acc
		✓	0.30	0.46	69.61
	✓		0.44	0.58	84.26
✓			0.60	0.72	89.67
	✓	✓	0.45	0.61	83.96
✓		✓	0.60	0.71	89.61
✓	✓		0.62	0.74	90.69
✓	✓	✓	**0.69**	**0.77**	**91.89**

TABLE VI
THE EFFECTS OF DIFFERENT TEMPORAL RELATIONSHIP

#	Temporal reception field	10	20	40	AP	M_{iou}
1	LSTM (base)	✓			0.60	0.72
2	Transformer [54]	✓			0.53	0.66
3	Bi-LSTM [55]	✓			0.59	0.72
4	LSTM (base)		✓		0.66	0.73
5	LSTM (base)			✓	0.70	**0.74**
6	LSTM (mt)	✓	✓		**0.76**	**0.74**
7	LSTM (mt)		✓	✓	0.75	0.73

ing, 2) different temporal relationship, 3) different extractor backbone, and 4) joint learning.

1) Different Attributes Learning: We testify the performance of SCTSNet with different shot attributes, as shown in Table V. When we only use one of the shot attributes (the first three rows in the table), we can find out that the movement attribute information is the most useful attribute for shot boundary detection, followed by the scale type, and then the angle type. We conjecture that this is due to the reason that movement has more granular categories than the others. It has eight types, while scale has five types, and angle has only three types. Attributes with high granularity categories guide the network to have a better ability to distinguish the differences between the two adjacent shots. Overall, with the help of more shot attribute categories, the performance is gradually improved. From SCTSNet (scale) to SCTSNet (movement + scale), the AP improves from 0.44 to 0.62 (relatively 41%) and M_{iou} from 0.58 to 0.74 (relatively 28%). From SCTSNet (movement + scale) to the full model, SCTSNet (movement + scale + angle), the AP raises relatively 11% and M_{iou} improves relatively 4%.

2) Different Temporal Relationship: To show the effectiveness of our video temporal segmentation network $\mathcal{T}$, we conduct study on $\mathcal{T}$ from two perspectives: network structure and temporal reception field. Results are shown in Table VI. In the first three settings, we study different temporal sequential models, including our LSTM, Transformer [54] and Bi-LSTM [55], with the same temporal reception field, *i.e.*, take 10 consecutive shots as input and predict the 9 boundaries among them. The baseline LSTM (base) is a single LSTM with 512 hidden dimensions using the movement attribute only. In Bi-LSTM setting, the hidden dimension is set to be 256. In Transformer setting, the input dimension is set to 1024, the number of heads in the multi-head-attention models is set to 8 and the number of sub-encoder-layers in the encoder is set to 6. From the experiment results, we can find out that the baseline single LSTM shares similar results with Bi-LSTM. However, Transformer achieves worse results as it may suffer from overfitting. Therefore, we choose the simple and efficient structure LSTM as our base model.

Furthermore, in the settings 1, 4-7 in Table VI, we study the performance of $\mathcal{T}$ under different time-range reception field. Specifically, in time range 10, 20, 40 settings, we test on predictions on the central 9 boundaries. We observe that when the time-range reception field increases from 10 to 40, the AP improves from 0.60 to 0.70 (relatively 16%), which shows that a larger temporal reception field helps to improve the performance. When we use multi-temporal-scale sequential model LSTM (mt), we find out that by adding a shorter temporal reception field's model, the performance improves about 10%, *e.g.*, AP increases from 0.66 to 0.76 comparing with the 4-th row and 6-th row, from 0.70 to 0.75 comparing with the 5-th row and 7-th row. The combination of temporal reception field $10 + 20$ is also much better than a single 40 time range, comparing with the 5-th row and 6-th row. These prove the effectiveness of our multi-temporal-scale design.

3) Different Extractor Backbone: We compare the effects of different extractor backbone and show the results in Table VII. To illustrate the impact of the effects of different backbone more clearly, we conduct an ablation study on the SCTSNet (+

TABLE VII
THE EFFECTS OF DIFFERENT EXTRACTOR BACKBONE

Settings			Boundary detection		Type classification		
Move.	Scale	Angle	AP	M_{iou}	Move.	Scale	Angle
Res50	Res50	Res50	0.37	0.55	*43.35*	68.86	72.94
I3D	*I3D*	Res50	0.56	0.69	71.50	*60.55*	72.94
I3D	Res50	*I3D*	0.65	0.76	71.50	68.86	*52.01*
I3D	Res50	Res50	**0.69**	**0.77**	**71.50**	**68.86**	**72.94**

TABLE VIII
THE EFFECTS OF JOINT TRAINING

Settings	Boundary detection			Type classification
	AP	M_{iou}	Acc	Acc
separate	0.26	0.34	72.00	60.55
joint	**0.60**	**0.72**	**89.67**	**71.50**

mf) with a single LSTM model that sets its temporal reception field as 10. Two backbone ResNet50 [48] and I3D [37] are tested.

SCTSNet (+ mf) is shown in the fourth row, where we apply I3D as the movement backbone, ResNet50 as the scale and angle backbones. When we compare the first and the last rows of Table VII, we can find out that the 3D convolution network brings significant improvement on the boundary detection (AP improves from 0.37 to 0.69) and the movement type classification (Acc improves from 43.35 to 71.50). This can be ascribed to better temporal representation ability of the 3D convolution network, which is exactly what the movement prediction needs. Comparing the second and fourth rows of the performance on scale attribute, ResNet50 achieves better results on boundary detection (relatively 23%) and scale type classification (relatively 13%) than I3D. Similar observations can be found from the experiments on the angle attribute. This may due to the reason that in the scale and angle feature learning, the spatial representation is more important than the temporal representation, which makes 2D networks to be more suitable here.

4) Joint Learning: Our SCTSNet has three parts, attributes feature extractor $\mathcal{E}$, shot types classification network $\mathcal{F}$ and video temporal segmentation network $\mathcal{T}$. These three parts are jointly trained together in our framework. To study the effects of different training processes, we compare the performance of separate training and joint training in Table VIII. We take the movement attribute extractor along with a 10 time-range LSTM as the base model. In SCTSNet-separate, we first train the $\mathcal{E}$ and $\mathcal{F}$ with the shot type classification loss. And then we fix $\mathcal{E}$ and train the $\mathcal{T}$ with boundary annotation. In SCTSNet-joint, $\mathcal{E}$, $\mathcal{F}$ and $\mathcal{T}$ are jointly trained together. The joint training brings significant improvements on boundary detection (AP increases from 0.26 to 0.60 and M_{iou} increases from 0.34 to 0.72), and shot type classification (Acc increases from 60.55 to 71.50, relatively 18%). The results indicate that positive synergy exists between these two tasks, and proves the effectiveness of joint training on boundary detection.

TABLE IX
COMPARISON ON TIME AND SPACE COST

Settings	FPS	Model size	AP
2D-CNN, Zhao *et al* [53]	555	112.09 MB	0.65
3D-CNN, Wu *et al* [11]	121	249.65 MB	0.64
SCTSNet (base)	489	256.06 MB	0.60
SCTSNet (full, mf + mt)	405	576.06 MB	0.77

TABLE X
THE EFFECTS OF DIFFERENT SAMPLING INTERVALS

Interval(s)	1	3/4	1/2	1/4	1/8	1/24
AP	0.51	0.55	0.60	0.60	0.61	0.62
M_{iou}	0.43	0.65	0.72	0.71	0.72	0.72

5) Time and Space Cost: We conduct experiments on a 90-minute movie with $129,600$ frames using one Tesla K80 GPU. All the methods use the 0.5 seconds sampling strategy for a fair comparison. The time and space cost comparison is shown in Table IX. The reported frames per second (FPS) is the quotient of the total processing time divided by the total frames. Although our full SCTSNet achieves better performance at the cost of larger time and space complexity, SCTSNet stays at a practical usage level with a speed above 400 FPS and the model size less than 1 GB. It takes about 5 minutes to test on a 90-minute long movie using all the modules.

D. Different Hyperparameters

1) Sampling Intervals: To study the effects of different sampling in splitting videos, we differ the sampling intervals on our base model and report the results in Table X, considering the FPS of the videos is 24. We find that $1/2$ second sampling interval outperforms 1 second sampling interval relatively 18% on AP and relatively 67% on M_{iou}. Starting from $1/2$ second sampling interval, the performance doesn't improve too much as the sampling becomes denser, even if we use every frame with the $1/24$ second sampling. It shows that $1/2$ (0.5) second sampling interval servers as a good balance between the performance and the computation cost.

2) Ensemble Weights: We differ the weights λ_i and μ_k in Equation (4) of the full model SCTSNet (full, mf + mt). λ_i and μ_k are the weights to ensemble the results of different shot attributes and temporal scale models respectively. The results are shown in Table XI and Figure 7. It is observed that the larger λ_m uses, the better the performance is. The best performance achieves when the weights are set to $\lambda_m : \lambda_s : \lambda_a = 0.7 : 0.2 : 0.1$. As for the choice of μ_k, recall that $w = 5$, we will use two temporal reception field 10 and 20 when $K = 2$. The sum of μ_k is set to be one. When $\mu_1 = 0.00$ and $\mu_2 = 1.00$, the model degrades to SCTSNet (+ mf) with $w = 10$ and $K = 1$. As μ_1 increases, the performance first rises and then drops. It achieves the best when the weights μ_1 and μ_2 are set to be 0.10 and 0.90.

TABLE XI
THE EFFECTS OF λ_i, $\{m, s, a\}$ REFER TO MOVEMENT, SCALE AND ANGLE

λ_m	λ_s	λ_a	AP	M_{iou}	Acc
0.20	0.40	0.40	0.64	0.51	89.11
0.40	0.30	0.30	0.74	0.68	92.94
0.60	0.20	0.20	0.76	0.72	93.43
0.70	0.15	0.15	0.76	0.72	93.44
0.60	0.30	0.10	0.76	0.72	93.55
0.70	0.20	0.10	0.76	**0.73**	**93.49**
0.80	0.10	0.10	0.75	0.73	93.48

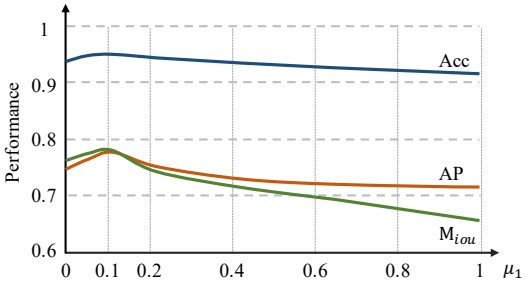

Fig. 7. The effects of μ_k on performance where $K = 2$ and $\mu_1 + \mu_2 = 1$.

Fig. 8. Qualitative results of different shot attributes in shot transformation from movie *Kill Bill (2003)*, *Iron Man (2008)* and *X-Man: Days of Future Past (2014)*. Each shot is represented with two frames. Three colors correspond to three attributes, where blue is camera movement, red is scale and green is angle. The rectangle length of each attribute is its softmax score to represent its effects in determining a shot boundary.

Fig. 9. (Top) The shots have complex camera movement and actors performance with long duration from movie *X-Man: Days of Future Past (2014)*. (Bottom) There are moving objects that occupy most of the space in the frame from movie *Kill Bill (2003)*.

E. Qualitative Results

In this section, we present the qualitative results to show the effectiveness of multi-attribute feature learning. We also analyze the error cases and imply some future directions.

1) Multi-attribute Feature Learning: In Figure 8, we visualize the effects of different attributes to illustrate how they contribute to the prediction of the shot boundary. In the first two cases Figure 8, we can see that different shot attributes contribute differently. While a single attribute can not convey most situations, due to the variety of shot transitions, multi-attribute can better help the network to identify the boundary. In the last two cases in Figure 8, the appearance difference among adjacent shots is subtle. With the recognition of shot attributes changes, these two cases could be successfully handled. Compared to the traditional methods using hand-crafted features and empirical thresholds, the multi-attribute features used in our SCTSNet are complementary to each other and help the shot boundary detection.

2) Error Case Analysis: Although our framework can work very well in most of the cases, there are still some hard cases that existing methods and our method cannot deal. In the two cases shown in Figure 9, although the appearance/brightness of the frames changes significantly, they are single shots. But all the methods will falsely cut the shot into several pieces. The reason is that the first case in Figure 9 is a long take, where both appearance and shot attributes change significantly. The foreground in the second case in Figure 9 is too big and results in the false prediction on shot attributes. How to solve these shots remains our future work. We conjecture that visual information only may not be enough to make the right decision. Multi-modal features, *e.g.*, audio or text, might help with these cases.

VI. CONCLUSION

This paper proposes a joint learning framework to detect shot boundaries in videos. Inspired by the professional filming theory in cinematographic art, instead of only focusing on the boundary visual difference, we resort to the analysis of the shot attributes and contents. A joint classification and temporal segmentation network (SCTSNet) is proposed to segment shot by learning multiple shot attributes and utilizing multi-temporal-scale information. To support the study on it, we collect a large-scale video shot boundary and attribute dataset *MovieShots2*, which contains 15091 shots coming from 282 movie clips. With detailed experiments, the proposed framework achieves better results than existing methods and proves the effectiveness of every design.

ACKNOWLEDGMENT

This work is partially supported by the SenseTime Collaborative Grant on Large-scale Multi-modality Analysis (CUHK Agreement No. TS1610626 & No. TS1712093), the General Research Fund (GRF) of Hong Kong (No. 14203518 & No. 14205719), Innovation and Technology Support Program (ITSP) Tier 2, ITS/431/18F, and National Key R&D Program of China 2017YFB1402203-2.

REFERENCES

[1] I. Cherif, V. Solachidis, and I. Pitas, "Shot type identification of movie content," in *2007 9th International Symposium on Signal Processing and Its Applications*. IEEE, 2007, pp. 1–4. 1, 2

[2] G. L. Priya and S. Domnic, "Edge strength extraction using orthogonal vectors for shot boundary detection," *Procedia Technology*, vol. 6, pp. 247–254, 2012. 1, 2

[3] H. Wang, A. Divakaran, A. Vetro, S.-F. Chang, and H. Sun, "Survey of compressed-domain features used in audio-visual indexing and analysis," *Journal of Visual Communication and Image Representation*, vol. 14, no. 2, pp. 150–183, 2003. 1, 2

[4] N. J. Janwe and K. K. Bhoyar, "Video shot boundary detection based on jnd color histogram," in *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*. IEEE, 2013, pp. 476–480. 1, 2

[5] L. Canini, S. Benini, and R. Leonardi, "Classifying cinematographic shot types," *Multimedia tools and applications*, vol. 62, no. 1, pp. 51–73, 2013. 1, 2

[6] I. Karakostas, I. Mademlis, N. Nikolaidis, and I. Pitas, "Shot type constraints in uav cinematography for autonomous target tracking," *Information Sciences*, vol. 506, pp. 273–294, 2020. 1

[7] L. D. Giannetti and J. Leach, *Understanding movies*. Prentice Hall Upper Saddle River, New Jersey, 1999, vol. 1, no. 1. 1, 2, 3

[8] A. Rao, J. Wang, L. Xu, X. Jiang, Q. Huang, B. Zhou, and D. Lin, "A unified framework for shot type classification based on subject centric lens," in *2020 European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 4

[9] B. Castellano, "Pyscenedetect: Intelligent scene cut detection and video splitting tool," https://pyscenedetect.readthedocs.io/en/latest/, 2018. 2, 3, 6, 7

[10] Z.-M. Lu and Y. Shi, "Fast video shot boundary detection based on svd and pattern matching," *IEEE Transactions on Image processing*, vol. 22, no. 12, pp. 5136–5145, 2013. 2, 6, 7

[11] L. Wu, S. Zhang, M. Jian, Z. Lu, and D. Wang, "Two stage shot boundary detection via feature fusion and spatial-temporal convolutional neural networks," *IEEE Access*, vol. 7, pp. 77 268–77 276, 2019. 2, 6, 7, 8

[12] J. Xu, L. Song, and R. Xie, "Shot boundary detection using convolutional neural networks," in *2016 Visual Communications and Image Processing (VCIP)*. IEEE, 2016, pp. 1–4. 2, 6

[13] W. Tong, L. Song, X. Yang, H. Qu, and R. Xie, "Cnn-based shot boundary detection and video annotation," in *2015 IEEE international symposium on broadband multimedia systems and broadcasting*. IEEE, 2015, pp. 1–5. 2

[14] M. Xu, J. Wang, M. A. Hasan, X. He, C. Xu, H. Lu, and J. S. Jin, "Using context saliency for movie shot classification," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 3653–3656. 2, 4

[15] M. Svanera, S. Benini, N. Adami, R. Leonardi, and A. B. Kovács, "Over-the-shoulder shot detection in art films," in *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2015, pp. 1–6. 2

[16] H. Jiang and M. Zhang, "Tennis video shot classification based on support vector machine," in *2011 IEEE International Conference on Computer Science and Automation Engineering*, vol. 2. IEEE, 2011, pp. 757–761. 2

[17] H. L. Wang and L.-F. Cheong, "Taxonomy of directing semantics for film shot classification," *IEEE transactions on circuits and systems for video technology*, vol. 19, no. 10, pp. 1529–1542, 2009. 2, 4

[18] S. Bhattacharya, R. Mehran, R. Sukthankar, and M. Shah, "Classification of cinematographic shots using lie algebra and its application to complex event recognition," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 686–696, 2014. 2, 4

[19] J.-C. Lin, W.-L. Wei, T.-L. Liu, Y.-H. Yang, H.-M. Wang, H.-R. Tyan, and H.-Y. M. Liao, "Coherent deep-net fusion to classify shots in concert videos," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3123–3136, 2018. 2

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 2

[21] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *Signal processing: Image communication*, vol. 16, no. 5, pp. 477–500, 2001. 2

[22] C. Yasira Beevi and S. Natarajan, "An efficient video segmentation algorithm with real time adaptive threshold technique," 2009. 2

[23] T. Lindeberg, "Scale invariant feature transform," 2012. 2

[24] J. Baber, N. Afzulpurkar, and S. Satoh, "A framework for video segmentation using global and local features," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 05, p. 1355007, 2013. 2

[25] S. Tippaya, S. Sitjongsataporn, T. Tan, M. M. Khan, and K. Chamnongthai, "Multi-modal visual features-based video shot boundary detection," *IEEE Access*, vol. 5, pp. 12 563–12 575, 2017. 2

[26] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," *IEEE transactions on circuits and systems for video technology*, vol. 17, no. 2, pp. 168–186, 2007. 2, 6, 7

[27] B. Luo, H. Li, T. Song, and C. Huang, "Object segmentation from long video sequences," in *Proceedings of the 23rd ACM international conference on multimedia*, 2015, pp. 1187–1190. 2

[28] D. Ghadiyaram, D. Tran, and D. Mahajan, "Large-scale weakly-supervised pre-training for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 046–12 055. 2

[29] Q. Huang, Y. Xiong, A. Rao, J. Wang, and D. Lin, "Movienet: A holistic dataset for movie understanding," in *The European Conference on Computer Vision (ECCV)*, 2020. 2

[30] J. Xia, A. Rao, L. Xu, Q. Huang, J. Wen, and D. Lin, "Online multi-modal person search in videos," in *The European Conference on Computer Vision (ECCV)*, 2020. 2

[31] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 146–10 155. 2

[32] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based lstm and semantic consistency," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2045–2055, 2017. 2

[33] J. Dong, X. Li, and C. G. Snoek, "Predicting visual features from text for image and video caption retrieval," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3377–3388, 2018. 2

[34] J. Chen, Y. Pan, Y. Li, T. Yao, H. Chao, and T. Mei, "Temporal deformable convolutional encoder-decoder networks for video captioning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8167–8174. 2

[35] M. Gygli, Y. Song, and L. Cao, "Video2gif: Automatic generation of animated gifs from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1001–1009. 2

[36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497. 2

[37] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308. 2, 5, 8

[38] C. Zhang and Y. Tian, "Automatic video description generation via lstm with joint two-stream encoding," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2924–2929. 2

[39] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (t-cnn) for action detection in videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5822–5831. 2

[40] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5552–5561. 2

[41] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542. 2

[42] Y. Lu, C. Lu, and C.-K. Tang, "Online video object detection using association lstm," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2344–2352. 2

[43] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634. 3

[44] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2740–2755, 2018. 3

[45] L.-Y. Duan, M. Xu, Q. Tian, C.-S. Xu, and J. S. Jin, "A unified framework for semantic shot classification in sports video," *IEEE Transactions on multimedia*, vol. 7, no. 6, pp. 1066–1083, 2005. 4

[46] NIST, "Trec video retrieval evaluation home page," https://trecvid.nist.gov/. 4, 6

[47] L. Li, X. Zhang, W. Hu, W. Li, and P. Zhu, "Soccer video shot classification based on color characterization using dominant sets clustering," in *Pacific-Rim Conference on Multimedia*. Springer, 2009, pp. 923–929. 4

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 5, 8

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 5

[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015. 6

[51] J. S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques," *Journal of Electronic Imaging*, vol. 5, no. 2, pp. 122–129, 1996. 6, 7

[52] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?" *IEEE transactions on circuits and systems for video technology*, vol. 12, no. 2, pp. 90–105, 2002. 6, 7

[53] B. Zhao, X. Li, and X. Lu, "Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7405–7414. 6, 7, 8

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. 7

[55] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005. 7

Xuekun Jiang received the B.S. degree from the School of Computer Science, Communication University of China in 2017. He is currently pursuing the Ph.D. degree in Information and Communication Engineering School. Communication University of China. His research interests include video analysis and deep learning.

Libiao Jin received the B.S. degree in electronic information engineering from Minzu University of China in 2000, and the M.S. and Ph.D. degrees in communication and information systems from Communication University of China in 2003 and 2008, respectively. He is currently a professor with the School of Information and Communication Engineering, Communication University of China, Beijing, China. He is the author of three books and more than 80 articles. His research interests mainly include artificial intelligence, information network, multimedia communication and wireless communication.

Anyi Rao received the B.S. degree from Nanjing University in 2018. He is a Ph.D. candidate at the Multimedia Laboratory, the Chinese University of Hong Kong. He visited the Department of Computer Science, the University of Toronto in 2020, the Department of Computer Science, the University of Hong Kong in 2017. He is interested in computer vision, multimodality, video semantic understanding, cinematic style analysis and editing.

Linning Xu received the B.S. degree from the Chinese University of Hong Kong, Shenzhen (CUHKSZ) in 2020. She is a Ph.D. student at the Multimedia Laboratory, the Chinese University of Hong Kong. She has broad interest in machine learning, with special focus on computer vision now.

Dahua Lin Dahua Lin is an Associate Professor at the department of Information Engineering, the Chinese University of Hong Kong, and the Director of CUHK-SenseTime Joint Laboratory. He received the B.Eng. degree from the University of Science and Technology of China (USTC) in 2004, the M. Phil. degree from the Chinese University of Hong Kong (CUHK) in 2006, and the Ph.D. degree from Massachusetts Institute of Technology (MIT) in 2012. Prior to joining CUHK, he served as a Research Assistant Professor at Toyota Technological Institute at Chicago, from 2012 to 2014. His research interest covers computer vision and machine learning. He serves on the editorial board of the International Journel of Computer Vision (IJCV). He also serves as an area chair for multiple conferences, including ECCV 2018, ACM Multimedia 2018, BMVC 2018, CVPR 2019, BMVC 2019, AAAI 2020, and CVPR 2021.