CNN-Based Shot Boundary Detection and Video Annotation

Wenjing Tong[1], Li Song[1,2], Xiaokang Yang[1,2], Hui Qu[1], Rong Xie[1,2]
[1]Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University
[2]Cooperative Medianet Innovation Center, Shanghai China
{tongwenjing, song_li, xkyang, xierong}@sjtu.edu.cn, quhui2005@gmail.com

***Abstract*—With the explosive growth of video data, content-based video analysis and management technologies such as indexing, browsing and retrieval have drawn much attention. Video shot boundary detection (SBD) is usually the first and important step for those technologies. Great efforts have been made to improve the accuracy of SBD algorithms. However, most works are based on signal rather than interpretable features of frames. In this paper, we propose a novel video shot boundary detection framework based on interpretable TAGs learned by Convolutional Neural Networks (CNNs). Firstly, we adopt a candidate segment selection to predict the positions of shot boundaries and discard most non-boundary frames. This pre-processing method can help to improve both accuracy and speed of the SBD algorithm. Then, cut transition and gradual transition detections which are based on the interpretable TAGs are conducted to identify the shot boundaries in the candidate segments. Afterwards, we synthesize the features of frames in a shot and get semantic labels for the shot. Experiments on TRECVID 2001 test data show that the proposed scheme can achieve a better performance compared with the state-of-the-art schemes. Besides, the semantic labels obtained by the framework can be used to depict the content of a shot.**

***Index Terms*—Retrieval and indexing, shot boundary detection, deep learning, convolutional neural networks, video coding and processing**

I. INTRODUCTION

With the development of multimedia technology, digital video has a rapid growth in both quality and quantity. Consequently, video indexing, browsing, retrieval, representation and other video analysis technologies have drawn much attention. Video shot boundary detection is the first and fundamental step for those technologies. A video shot is defined to be a sequence of images which are captured by a single camera in an uninterrupted run[1]. There are two kinds of transitions between two adjacent shots: cut transition(CT) and gradual transition(GT). CT consists of the last frame of the previous shot and the first frame of the next shot. While GT usually consists of several frames and the transition between adjacent shots is in a milder manner. Thus video shot boundary detection is a process of identifying the transition between every two adjacent shots[2].

Great efforts have been made on video shot boundary detection, and most of them are about cut detection. They all take advantage of the abrupt change between the adjacent frames in different shots. In [3], a pixel-wise difference and adaptive threshold-based CT detection is adopted. Z. Lu et al. [4] proposed the frame-feature matrix extracted from the color histogram in the hue-saturation-value space. The similarity between frames is defined by the Just Noticeable Difference (JND) color histogram in [5]. Compared with CT detection, GT detection is a much harder task. Besides the preprocessing of the input video, a sequence of frame distances are computed and the triangle pattern[3] is also searched and used for GT boundaries detection. In [4], the SVD is used to describe the frame pixel histogram, then an empirical triangle pattern is observed to detect GT boundaries.

So far most works ignore the frame content. But considering the definition of video shots, it is a natural thought of using the frame content to do shot detection. In this paper, we propose to use CNN to extract high-level interpretable features from the frames, thus promoting the accuracy of SBD. Recently, CNNs[6] have been demonstrated as an effective class of models for understanding contents of videos and images. In CNNs, low-level features, such as edges and angles, can be learned by the first several layers. And high-level features can be extracted from low-level ones as the networks go deeper[7].

Another task related to SBD is video annotation, which is the allocation of video shots to different predefined semantic concepts[8]. Existing methods for video annotation follow these two steps: First, extracting low-level features, and then training classifiers to classify shots to concepts. A supervised learning method based on mid-level features[9] is proposed for annotation in sports video. A cross-training strategy[10] is used to stack concept detectors into a single discriminative classifier. Since we already have high-level semantic features learned by CNNs, we directly synthesize these features and perform annotation for shots.

II. APPROACH

A. Feature Extraction Using CNN

We use a CNN model which is similar to the ImageNet challenge winning model[11]. The main architecture of the CNN model is shown in Fig. 1. The network contains eight layers with weights: the first five ones are convolutional layers and the remaining layers are fully-connected layers.

The first convolutional layer has a total of 96 convolutional kernels of size $11 \times 11 \times 3$ with a stride of 4 pixels. The second convolutional layer has 256 kernels of size $5 \times 5 \times 48$. The third, fourth and fifth convolutional layers have 384 kernels of size $3 \times 3 \times 192$. The first and second convolutional layers

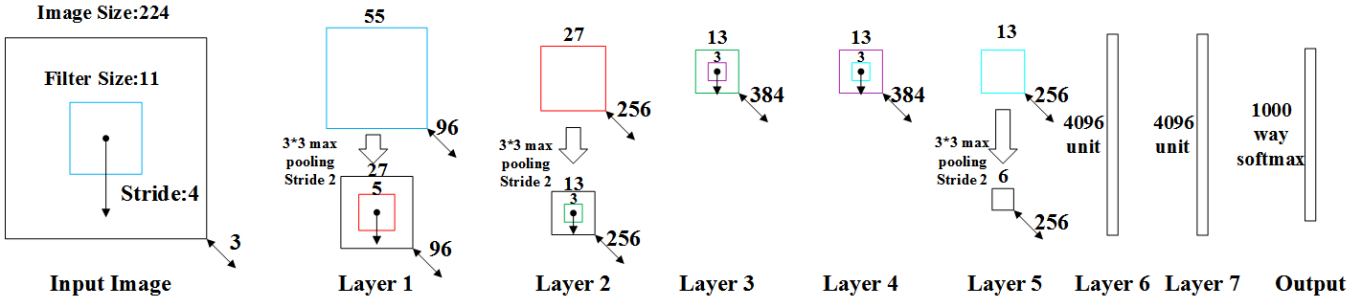

Fig. 1. The main architecture of the CNN model with 1000-way softmax layer.

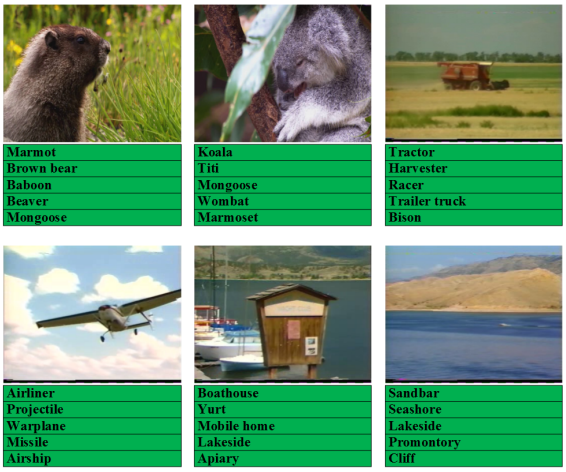

Fig. 2. TAGs of several example frames extracted by the CNN model.

have local response normalization layers right behind them. After the local response normalization layers are max-pooling layers. And the fifth convolutional layer directly connect with a max-pooling layer.

We train the network on ImageNet dataset with 50000 iterations. Taken one frame as input, the output of the network is a probability distribution among 1000 classes. The five classes with the highest probabilities are selected as the high-level features of the frame and called as the TAGs of the frame for simplicity. As shown in Fig. 2, these TAGs can describe the contents of the frame well. Therefore, they can be used for better shot detection. Because frames in the same shot tend to share similar TAGs while frames in different shots do not.

B. Candidate Segmentation Selection

Generally, most frames in a video sequence with more than one shot inside are non-boundary frames. So we take candidate segment selection as our first step to eliminate those non-boundary frames. This step can increase both accuracy and speed of the further steps. Candidate segment selection is based on the fact that consecutive frames within one shot always have high correlations[3]. Thus if the first and last frames of one segment share great similarities, there are no boundaries within this segment.

The candidate segment selection process is similar to that in [3], but we employ a new adaptive threshold to make it more precise and suitable for the next CNNs-based SBD steps. The process is illustrated as follows:

Step 1: Calculating distance of segment. Firstly, cut video sequence into segments of 21 consecutive frames. For example, the n th segment ranges from the $20n$ th frame to the $20(n+1)$ th frame. Then we calculate the luminance distance between the first and last frames of one segment as equation (1):

$$d(20n, 20(n+1)) = \sum_x \sum_y |F(x, y; 20n) - F(x, y; 20(n+1))| \tag{1}$$

where $F(x, y; k)$ denotes the luminance component of the pixel in the position (x, y) at frame k. In the remainder of this paper, we use $d(n)$ to represent $d(20n, 20(n+1))$ for simplicity. The luminance distance is a common feature and is easy for calculation.

Step 2: Calculating local threshold. Every 100 consecutive segments are grouped into one unit and the mean value of all $d(n)$ in the unit is calculated, i.e., the unit mean value. And for the n th segment, a sliding window of length 5 centered at the current segment is used to calculate the local mean and local standard deviation. Then we can obtain the local adaptive threshold of the n th segment in equation (2):

$$T_L^n = \mu_L + 0.7(1 + ln(\frac{\mu_{unit}}{\mu_L}))\sigma_L \tag{2}$$

where μ_{unit} denotes the unit mean value, μ_L denotes the local mean value and σ_L is the local standard deviation. During experiment, we find the original T_L^n in [3], i.e., $T_L^n = 1.1\mu_L + 0.6(1 + ln(\frac{\mu_G}{\mu_L}))\sigma_L$ is not a so good threshold that some segments with CT are falsely ejected. Thus we change it to a more local adaptive value.

Step 3: Classifying candidate segment. Compare $d(n)$ of the n th segment with T_L^n, if it is greater than T_L^n, that segment may contain shot boundary and is classified as a candidate segment. When $d(n)$ of a segment is smaller than T_L^n and is much larger than those of the neighboring segments, it is also considered as a candidate segment. Thus we classify those segments whose distances satisfy equation (3) or (4) as candidate segments.

$$d(n) > T_L^n \tag{3}$$

$$(d(n) > 3d(n-1) \ or \ d(n) > 3d(n+1)) \\ and \ d(n) > 0.8\mu_{unit} \quad (4)$$

Step 4: First round bisection-based comparisons. This step divides candidate segment into two parts, discards the non-boundary parts and preserves the suspect shot change parts. The concrete steps are listed as follows:

Firstly, calculate the forward distance $d_F(n)$ and backward distance $d_B(n)$ of the n th segment according to equation (5) and (6).

$$d_F(n) = \sum_x \sum_y |F(x,y;20n+10) - F(x,y;20(n))| \quad (5)$$

$$d_B(n) = \sum_x \sum_y |F(x,y;20n+10) - F(x,y;20(n+1))| \quad (6)$$

Next, each candidate segment will be classified as one of the four types according to Table I.

TABLE I
CANDIDATE SEGMENT TYPES

Type	Condition the distances
Type 1	$\frac{d_F}{d_B} > 1.5$ and $\frac{d_F}{d} > 0.7$
Type 2	$\frac{d_B}{d_F} > 1.5$ and $\frac{d_B}{d} > 0.7$
Type 3	$\frac{d_F}{d} < 0.3$ and $\frac{d_B}{d} < 0.3$
Type 4	Else

In **Type 1**, the first half of the segment is kept as candidate segment while the second half is discarded. Similarly, in **Type 2**, the second half is kept while the first half is discarded. In **Type 3**, the whole segment is discarded. In **Type 4**, the segment is regarded as having a GT boundary in it.

Step 5: Second round bisection-based comparisons. The same bisection-based comparison is performed in candidate segments of length 11 obtained in **step 4**. After this step, candidate segments of length 6 are suspected to have a CT within them. While the other candidate segments with length 11 or 21 are considered to have a GT within them. Thus we can employ different shot boundary detection algorithms to the two types of candidate segments.

C. CT Detection

CT detection is applied to those candidate segments of length 6. Considering the consistency of our method and the computational complexity, we adopt the pixel-wise difference and the adaptive threshold-based method in [3]. Assuming the n th segment starts at the s th frame and ends at the e th frame.

TABLE II
TEST VIDEOS ADN THE DESCRIPTIONS

Videos	Frames	Transitions			Sources
		Total	CT	GT	
anni001	914	8	0	8	7 documents from TRECVID 2001 test data[12]
anni005	11358	65	38	27	
anni007	1590	11	5	6	
anni008	2775	14	2	12	
anni009	12304	103	38	65	
BOR10_001	1815	11	0	11	
BOR10_002	1795	10	0	10	
Total	32551	222	83	139	

In [3], a CT occurs in the segment if the the expressions (7) and (8) are satisfied:

$$t_m = argmax_{s \le t \le e} d(t, t+1) \quad (7)$$

$$\frac{d(t_m, t_m+1)}{d(t_{sm}, d_{sm}+1) + C} \ge 3 \quad (8)$$

where $d(t_{sm}, t_{sm}+1)$ is the second maximum distance value, C is a small constant for avoiding divide-by-zero error.

Besides, one more restriction is added for better CT detection performance. Let $T(t_m)$ denote the set of TAGs of the t_m th frame. The restriction is defined as follows:

$$|(T(t_m-3) \cap T(t_m-1)) \cap (T(t_m+2) \cap T(t_m+4))| \le 1 \quad (9)$$

As shown in expression (9), the two previous frames of the suspected CT boundary are used to get semantic information which can express contents of the previous shot section. Similarly, the two afterwards frames are used to get contents information of the next shot section. There shouldn't be too many intersections of the semantic information from the two parts if a CT occurs. If expression (9) is also satisfied, the CT starts at the t_m th frame and ends at the t_m+1 th frame. The segments which do not satisfy one of expressions (7), (8) and (9) are treated as gradual-transition segments.

D. GT Detection

GT detection is employed in this step to detect segments with gradual transitions in them. The gradual transition detection is always a difficult task, since it often occurs through a serial of frames, and the difference of each frame pairs is relatively lower. Here we use CNNs to get TAGs of the previous and next frames of one candidate segment and analyse the relationships between those TAGs to judge if the segment has gradual transition in it. Let the n th segment starts at the s th frame and ends at the e th frame. Let $K_F(n)$ denote the combined TAGs of the first half of the n th candidate segment and $K_B(n)$ denote the combined TAGs of the second half. The $K_F(n)$ and $K_B(n)$ are defined as follows:

$$K_F(n) = T(s-5) \cap T(s-3) \cap T(s-1) \quad (10)$$

$$K_B(n) = T(e+5) \cap T(e+3) \cap T(e+1) \quad (11)$$

The main reason to combine the neighbour frames rather than the internal frames of the candidate segments is that the

TABLE III
RECALL(R),PRECISION(P)AND F_1 VALUES OF CT DETECTION

Videos	Recall			Precision			F_1		
	[3]	[4]	**Proposed**	[3]	[4]	**Proposed**	[3]	[4]	**Proposed**
anni001	—	—	—	—	—	—	—	—	—
anni005	0.947	**0.974**	0.895	0.947	0.881	**1**	**0.947**	0.925	0.932
anni007	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
anni008	**1**	**1**	**1**	**1**	0.667	**1**	**1**	0.800	**1**
anni009	0.737	0.737	**0.821**	0.875	0.875	**1**	0.800	0.800	**0.901**
BOR10_001	—	—	—	—	—	—	—	—	—
BOR10_002	—	—	—	—	—	—	—	—	—
Average	0.852	0.867	**0.869**	0.958	0.878	**0.986**	0.902	0.892	**0.924**

TABLE IV
RECALL(R),PRECISION(P)AND F_1 VALUES OF GT DETECTION

Videos	Recall			Precision			F_1		
	[3]	[4]	**Proposed**	[3]	[4]	**Proposed**	[3]	[4]	**Proposed**
anni001	0.375	0.875	**1**	0.750	0.700	**1**	0.500	0.778	**1**
anni005	**0.963**	**0.963**	0.889	0.839	0.426	**0.857**	**0.897**	0.591	0.857
anni007	**1**	**1**	**1**	0.833	0.417	**1**	0.909	0.589	**1**
anni008	**0.917**	0.833	**0.917**	0.846	0.500	**0.917**	0.880	0.625	**0.917**
anni009	0.692	**0.800**	0.734	0.918	0.675	**0.940**	0.789	0.732	**0.825**
BOR10_001	0.636	**1**	0.909	0.875	0.647	**0.909**	0.737	0.786	**0.909**
BOR10_002	**0.800**	**0.800**	**0.800**	**1**	0.571	0.842	**0.889**	0.666	0.842
Average	0.746	**0.875**	0.826	**0.887**	0.564	0.867	0.810	0.686	**0.867**

semantic information of frames in a gradual-transition candidate segment is not stable. But the contents of non-boundary segments are always continuous. Then a gradual transition occurs in the s th segment if the following expression is satisfied:

$$K_F(n) \cap K_B(n) = \emptyset \tag{12}$$

E. Annotation For Shots

After segmenting the input video into shots, we extract the first, the middle and the last frames of each shot. Then TAGs from CNNs are attached to those three frames. After that, the TAGs of frames of a shot are integrated to get semantic labels for the shot. Let f_k^{first}, f_k^{middle} and f_k^{last} denote the first, middle and last frame of the k th shot, respectively. Let s_k denote the k th shot and $L(s_k)$ denote the semantic labels for the k th shot.

$$L(s_k) = (T(f_k^{first}) \cap T(f_k^{middle})) \cup (T(f_k^{last}) \cap T(f_k^{middle})) \tag{13}$$

ImageNet is an image database organized according to the WordNet hierarchy(in tree structures)[13]. Thus TAGs are leaf nodes of the WordNet tree. For each TAG in $L(s_k)$, the content of grandfather node of the TAG node is selected as the semantic labels for the k th shot. The main purpose of this step is that the categories of the ImageNet may be too concrete for video annotation, e.g. synsets dalmatian, alsatian and husky are all ImageNet classes but are less clear than synset dog for annotation purpose.

III. EXPERIMENT

In this section we show the segmentation and annotation results of our framework. We also compare our segmentation algorithm with state-of-the-art methods, namely [3] and [4]. The test videos and their descriptions are listed in Table II.

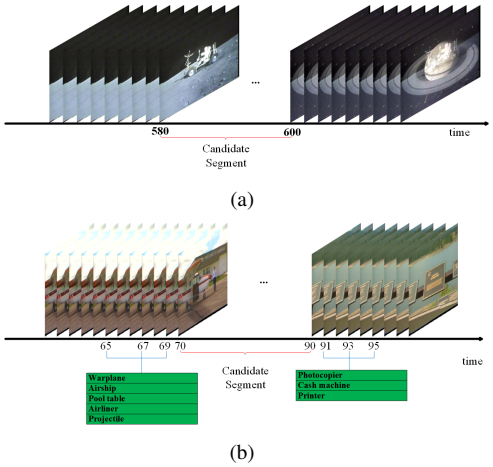

Fig. 3. Examples of the candidate segment detection and gradual transition detection. (a) A candidate segment is detected by the candidate segment selection with the new threshold computation method. (b) An example result of gradual transition detection by our method.

A. Evaluation Standards

Most of the papers about SBD take *recall, precision* and F_1 as evaluation standards. They are calculated as:

$$recall = \frac{N_c}{N_C + N_m} \tag{14}$$

$$precision = \frac{N_c}{N_c + N_f} \tag{15}$$

$$F_1 = \frac{2 \times recall \times precision}{recall + precision} \tag{16}$$

where N_c stands for the number of correctly detected shot boundaries, N_m stands for the missed shot boundaries. The

Fig. 4. An example result of video annotation by our method. Each figure is the middle frame of one shot which has been annotated.

value of F_1 is often used for ranking the performance of different SBD methods, since this value takes both precision and recall into account.

B. Segmentation Results Comparison

The *recall, precision* and F_1 values for CT detection of [3], [4] and the proposed framework are listed in Table III. While the GT detection results are shown in Table IV. The mean values of F1 in CT and GT detection of our proposed method are 0.924 and 0.867, respectively, illustrating that our SBD algorithm outperforms the state-of-the-art methods.

A concrete example is given in Fig. 3. For the gradual transition candidate segment in Fig. 3(a), both [3] and [4] fail to detect it, but the proposed method detects it successfully due to a more adaptive threshold. And for the segment with gradual transition in Fig. 3(b), it is falsely rejected by [3] and [4], but it is detected successfully by our method since the frames' TAGs of the two adjacent shots have little in common(i.e., equation (12) is satisfied).

C. Annotation Results

In addition, our experiment results prove that the proposed framework can do a well performance annotation for video shots. Fig. 4 shows one example of our video annotation results. Each figure is the middle frame of one shot. We can see that the semantic labels added by the program can depict the contents of the shots well. More results are available at the website http://medialab.sjtu.edu.cn/research/ShotDetection.html.

IV. CONCLUSIONS

In this paper, we present a novel video shot boundary detection approach based on frames' TAGs, which are generated by a CNN model. It is capable of detecting both CT and GT boundaries and is proven to outperform the state-of-the-art methods by the experiment results. We also merge TAGs of one shot to perform video annotation on that shot. Experiment results show that the semantic labels allocated to the shots can depict the contents of the shots well.

V. ACKNOWLEDGMENT

This work was supported by NSFC (61221001), the 111 Project (B07022) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

REFERENCES

[1] C. Cotsaces, N. Nikiolaidis, and I. Pitas, Video shot detection and condensed reprsentation. A review, *IEEE Signal Process. Mag*, Vol. 23, no. 2, pp. 28-37, Mar. 2006.

[2] J. H. Yuan, H. Y. Wang, L. Xiao, W. J. Zheng, J. M. Li, F. Z. Lin, and B. Zhang, A formal study of shot boundary detection, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, pp. 168-186, Feb. 2002.

[3] Y. Li, Z. Lu, and X. Niu, Fast video shot boundary detection framework employing pre-processing techniques, *IET Image Process.*, vol. 3, no. 3, pp. 121-134, Jun. 2009.

[4] Z. Lu and Y. Shi, Fast video shot boundary detection based on SVD and pattern matching, *IEEE Trans. Image Processing*, vol. 22, no. 12, pp. 5136-5145, Dec. 2013.

[5] N. Janwe and K. Bhoyar, Video shot boundary detection based on JND color histogram, in *Proc. Int. Conf. ICIIP.*, Shimla, Dec. 2013, pp. 476-480.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278-2324, 1998.

[7] M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:*1311.2901, 2013.

[8] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank, A Survey on Visual Content-Based Video Indexing and Retrieval, *IEEE Transactions on Systems*, Man, and Cybernetics, Part C, vol. 41, no. 6, pp. 797-819, 2011.

[9] L. Y. Duan, M. Xu, Q. Tian, and C.Xu, An unified framework for semantic shot classification in sports video, *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1066-1083, Dec. 2005.

[10] X. Shen, M. Boutell, J. Luo, and C. Brown, Multi-label machine learning and its application to semantic scene classification, in *Proc. Int. Symp. Electron. Imag.*, pp. 188-199, Jan. 2004.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, Imagenet classification with deep convolutional neural networks. In *NIPS.*, 2012.

[12] TREC video retrieval test collection. [Online]. Available:http://www.open-video.org/ , 2001.

[13] J Deng, A Berg, S Satheesh, H Su, A Khosla, and L Fei-Fei. Large scale visual recognition challenge. www.image-net.org/ ,2012