Research Article

Video Scene Detection Using Compact Bag of Visual Word Models

Muhammad Haroon¹,¹ Junaid Baber,¹ Ihsan Ullah,¹ Sher Muhammad Daudpota,² Maheen Bakhtyar,¹ and Varsha Devi³

¹Department of Computer Science & IT, University of Balochistan, Pakistan
²Department of Computer Science, Sukkur IBA University, Pakistan
³Department of Computer Science, Sardar Bahadur Khan Women's University, Pakistan

Correspondence should be addressed to Muhammad Haroon; haroonsdba@gmail.com

Received 18 May 2018; Revised 14 August 2018; Accepted 3 October 2018; Published 8 November 2018

Academic Editor: Deepu Rajan

Copyright © 2018 Muhammad Haroon et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Video segmentation into shots is the first step for video indexing and searching. Videos shots are mostly very small in duration and do not give meaningful insight of the visual contents. However, grouping of shots based on similar visual contents gives a better understanding of the video scene; grouping of similar shots is known as scene boundary detection or video segmentation into scenes. In this paper, we propose a model for video segmentation into visual scenes using bag of visual word (BoVW) model. Initially, the video is divided into the shots which are later represented by a set of keyframes. Keyframes are further represented by BoVW feature vectors which are quite short and compact compared to classical BoVW model implementations. Two variations of BoVW model are used: (1) classical BoVW model and (2) Vector of Linearly Aggregated Descriptors (VLAD) which is an extension of classical BoVW model. The similarity of the shots is computed by the distances between their keyframe feature vectors within the sliding window of length \( L \), rather comparing each shot with very long lists of shots which has been previously practiced, and the value of \( L \) is 4. Experiments on cinematic and drama videos show the effectiveness of our proposed framework. The BoVW is 25000-dimensional vector and VLAD is only 2048-dimensional vector in the proposed model. The BoVW achieves 0.90 segmentation accuracy, whereas VLAD achieves 0.83.

1. Introduction

The size of video databases is increasing exponentially due to the emergence of cheap and fast Internet. The indexing and retrieval of the videos are getting more difficult. The expectation of users are high due to advancement in technologies. The giant video portals, such as YouTube, Dailymotion, and Google, are investing huge amount on efficient and smart indexing and retrieval so that their portals remain attractive and addictive to the users.

To process videos for indexing and searching, the first task is to segment the videos into shots and extract representative frames, known as keyframes, from each shot. These keyframes are later used for searching, efficient indexing, scene generation, and video classification. Main idea to select key frame is to reduce the computational cost as video is the collection of frames which are stored in temporal order; i.e., every video uploaded on Youtube is 30 frames per second or higher. The more the frames per second, the better the visual effect. Despite being very sophisticated hardware, all the frames cannot be processed in real time applications such as event detection from CCTV streaming. To process one frame for the detection of possible objects, it takes 0.5 to 1.5 seconds to identify objects in the frame (cascade object detector is used to identify possible text boards in the frame using Matlab).

In video scene segmentation, the video is divided into shots and similar shots are combined together to make the scenes. Shots are uninterrupted and unceasing sequences of video frames where there is no change in theme and camera [1]. Generally, the video shots can be categorized into two types: abrupt shots and gradual shots. An abrupt shot
boundary is the sudden change in the scene such as change of speaker during TV interviews, whereas gradual shots take several frames to change the shot such as fades and dissolves. In videos, many shots repeat in very short interval of time; if those shots are combined then these collections of shots are called scenes. For example, if two actors are talking then the camera keeps switching to both actors with very little change in background and in two-minute conversation of video clip there are sometimes 25-30 shots. Scene detection, a.k.a. scene boundary detection or video scene segmentation, is the study to merge similar or repeating shots into one clip, or dividing the videos into clips which are semantically or visually related or similar.

Manual segmentation of videos for websites and DVDs is very time consuming and not feasible when dealing with large datasets. Recently, automatic video segmentation into shots and scenes have gained wide attraction among industry and researchers [2–5].

In the proposed methodology, videos are segmented into abrupt shot boundaries which are further grouped on the basis of the similarity to construct the scenes. The proposed methodology is inspired by the BoVW model for scene detection [2]; the abstract flow diagram is shown in Figure 1. In bag of visual word model, local key point descriptors which are extracted from the key frame of the shots are represented by the histograms of visual words. These key frames are matched based on bag of visual word histograms in sliding window of length $L$ [3]. It has been shown that shots matched in sliding windows are more efficient [2, 3]. Classical BoVW model and VLAD are used with compact vocabularies without compromising in accuracies.

Rest of the paper is organized as follows. Section 2 presents the related work. It is divided into three subsections: (1) shot boundary detection, (2) key frame extraction, and (3) scene boundary detection. Section 3 presents the proposed methodology along with experimental protocols, and finally Section 4 concludes the papers and discusses the future work.

2. Related Work

In this section, a brief literature review and state-of-the-art methodologies are presented for all the main steps of the video segmentation, which include shot boundary detection, key frame extraction, and scene boundary detection.

2.1. Shot Boundary Detection. In the problem of video indexing and searching, the first and foremost step is shot boundary detection. Shot boundary has two types as mentioned earlier: abrupt and gradual shot boundaries. Abrupt shot boundary is the sudden change in the stream; if the dissimilarity difference between the two consecutive frames is very large, then either of the adjacent frames is considered the boundary, whereas gradual shot boundary is the gradual change in the video such as the effects like fade-in, fade-out, and dissolves.

Let the $F = \{f_1, f_2, \ldots, f_n\}$ be frames of a video, and $f_i$ is said to be an abrupt shot boundary if and only if the difference between $f_i$ and $f_{i+1}$ is greater than a threshold $\tau$. In our experiments we are not taking the gradual shot detection into consideration as the ratio of gradual shots in any cinematic video is too small, more than 90% of shot boundaries are abrupt boundaries [2, 3].

There is long list of methodologies first on pixel to pixel difference between the consecutive video frames; i.e., $f_i$ and $f_{i+1}$ are used for segmentation of the video [6]. In this technique, if the sum of the pixels difference is greater than some threshold, then it was considered to be an abrupt shot boundary.

Later on, many other scientists worked on this problem and proposed a new technique in which pixel intensity histograms were used in successive frames instead of the pixel to pixel difference to detect the abrupt shot boundaries [7, 8]. These techniques are good except that they are sensitive to motion of objects and camera [9].

Moreover, a latter approach [10] detects the shot boundary based on the mutual information and joint entropy between the consecutive frames. A sports dataset has been used to detect the shot boundary. This technique of joint entropy is useful if used for faded or gradual boundaries. The entropy is high for an extended time period during fade-in because the visual intensity gradually increases, and the entropy is low during fade-out as the intensity slowly decreases.

Videos fragmentation by pixel variance in frames and pixel strength in histogram calculations has been presented in [11, 12]. The frame indexing was used with rapid boundary, when the amount of shot pixels between two frames is overdone by some threshold.

Chavez et al. [13] proposed a different technique in which they used supervised learning with support vector machine (SVM) in order to separate the abrupt boundaries from the gradual boundaries. In this technique, authors calculated the dissimilarity vector which assimilate set of different features, including Fourier-Mellin moments, Zernike moments, and color histograms (RGB and HSV) to capture the information like illumination changes and rapid motion. After then, this vector is used in SVM for detection of shot boundaries. The authors also used illumination changes for detecting the gradual shot boundaries.
Furthermore, [14] proposed a new technique of learning algorithm which has three main steps.

(1) Firstly, frames which have smooth changes are removed.

(2) Secondly, three types of feature differences are extracted. The intensity difference, difference in vertical and horizontal edge histograms, and difference between the HSV color histograms are calculated from the shot boundaries.

(3) Lastly, the authors detect the gradual boundaries from the video using a technique named as temporal multisolution analysis.

Several other works use various kinds of methodologies for different kind of shots. Such an example work [15] used different techniques for abrupt shots and gradual shots by utilizing SIFT and SVM. Their methodology also comprises few main steps which are given below.

(1) In the first step, they select the shot boundary frames from video using the difference between color histograms of two consecutive frames.

(2) Then in the second step, they extract the SIFT features from the frames selected as a shot boundary.

(3) Lastly, they use different approach for abrupt and gradual shot boundaries by using SIFT and SVM. SIFT is considered to be the most efficient, effective, and massively used in state-of-the-art techniques.

Although SIFT is considered to be the most widely used feature extraction technique, it still has some downsides as compared to SURF. The SIFT feature has high dimensional feature vector, i.e., 128-D, whereas the SURF only has 64-D vector. SIFT is slow as compared to SURF due to complex and comprehensive images. Moreover Baber et al. [3] proposed a new technique for shot boundary detection using two different feature extraction methods that are SURF and Entropy. Their research consists of different steps in which they detect shot boundaries (abrupt and gradual) and differentiate gradual boundaries from abrupt shot boundaries. The steps are as follows.

(1) In the first step, the fade boundaries are detected by the analysis of entropy pattern during fade effects.

(2) After the detection of fade shot boundaries, the other kind of shot boundary that is abrupt shot boundary is detected by using the entropy difference between two consecutive frames. If the difference between two consecutive frames is higher than the threshold $\tau$, then it is considered to be the abrupt shot boundary.

(3) SURF is used for removing the false negative boundaries.

2.2. Key Frame Extraction. Most of the researchers use the shot boundary detection as the important step to extract the representative key frames from the videos. Representative key frames in the videos are the particular frames which describe the whole content of the particular scene in the video. Each video may consist of one or more key frames based on the scenes or content in the video.

Shot boundary detection is one of the most crucial steps in our problem for finding representative key frames from the videos, as scene detection is completely based on these representative key frames. Baber et al. [5] used the entropy differences between two consecutive frames for finding the shot boundaries. If the contents of the two consecutive frames say $f_i$ and $f_{i+1}$ are different and their entropy difference is greater than the specified threshold, then $f_i$ is said to be a shot boundary and considered as a representative key frame.

In our methodology we have first calculated the entropy of each video frame and then the difference between two consecutive frames is recorded. The difference is greater than the the threshold $\tau$ is considered to be the representative key frame. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. Mathematically entropy is defined as

$$E_x = -\sum_y H_y \log(H_y)$$

where $H$ is the normalized histogram of the gray scale image's pixels intensity.

2.3. Scene Detection. At the first stage, the video is segmented in shots and semantically similar shots are merged to form scenes. Scenes are categorized into various classes such as conversation, indoor, and outdoor scenes. Many important researches have been published related to video segmentation into scenes using different type of videos, for example, cinematic, drama serials (indoor and outdoor), video lectures, and documentaries. Although a lot of work has been reported for segmentation of video into scenes, there is still a gap to address the challenge in cinematic videos. Commonly there are two types of features being extracted from the videos for segmentation, i.e., audio and visual. We have focused on the visual features in our research.

Yeung et al. [16] proposed a technique in which the authors used the scene transition graph (STG) to segment the video. The nodes in the graph are considered as a shot, which is based on the temporal relationship and visual similarity edges are described. Then the graph is divided into subgraphs and these subgraphs are considered as scenes which are based on the color similarity of the shots.

Rasheed et al. [17] proposed an effective technique for scene detection in Hollywood and TV shows. For features they have used the motion, color, and length of the shots. In the initial step, they first cluster the shots by using Backward Shot Coherence (BSC). Next, by calculating the color similarity between shots, they first detect potential scene boundaries and after that they remove the false negative from potential scene boundaries by scene dynamics which is based on motion and length of the shot.

Many recent authors worked on video scene segmentation and proposed new technique for this problem in research. Some researchers used multimodal fusion technique of optimally grouped features using the dynamic programming scheme [18–20]. Their methodology includes few steps in which the first step was to divide the video into
shots and then using clustering technique they cluster the shots. The authors in their paper [19] proposed a technique known as intermediate fusion, which uses all the information from different modalities. They considered this problem an optimization problem and used it via dynamic programming [19]. The authors have some previous research [18] in which they proposed a technique of dividing the video into scenes using the sequential structure. In this technique they decided a location for video segmentation and only inspected the partitioning possibilities. In this technique the video is represented by set of features and each set is given by a distance metric between them. The segmentation purely depends on input features and distance metric [18].

Furthermore, a different technique was proposed in which they used spectral clustering technique with an automatic selection on number of clusters and extracted the normalized histogram of each shot. Further they used Bhattacharyya distance and temporal distance as a distance metric. Authors in this paper said that clustering is not consistent and adjacent shots belong to different clusters [20].

Sakarya et al. [21] used a new technique of graph construction for the segmentation of video into scenes. They construct a graph, weighting the temporal and spatial function of similarity. From this the dominant shots are detected and for temporal consistency constraint they used the edges of the scene via mean and standard deviation of the shot position. This process kept on going until all the video is allocated to scenes. Lin et al. [22] used the approach of color histograms for the shot boundary detection and then formed the scene by merging the similar shots via identifying the local minima and maxima to determine the scene transitions.

Baraldi et al. [4] used another approach for shots and scene detection from the videos using the color histograms and clustering technique, respectively. The authors first detect the shots using the color histogram; then the authors clustered the shots using the hierarchical K-means clustering technique and created N clusters for N number of shots. Each shot is assigned a particular cluster and they find the least dissimilar shots using the distance metric formula and merged the two clusters with the least distance. This process continues until and unless all the scenes are detected and video is completed.

Chen et al. [23] proposed a new approach for scene detection from the H.264 video sequences. They define a scene change factor which is used to reserve bits for each frame. Their methodology has reduced rate error and was found better when compared with JVT-G012 algorithm. The work of [24] proposed a novel technique for scene change detection especially for H.264/AVC encoded video sequences and they take into consideration the design and performance evaluation of the system. They further worked with a dynamic threshold which adapts and tracks different descriptors and increased the accuracy of system by locating true scenes in the videos.

3. Proposed Methodology for Scene Detection

The proposed framework comprises shot boundary detection, key frame extraction, local key point descriptors extraction from key frames, feature quantization, and scene boundary detection.

3.1. Shot Boundary Detection. Shot boundary detection is the primary step for any kind of video operations. There are number of frameworks for shot boundary detection. We have used the technique for shot boundary detection based on entropy differences [5, 26]. The entropy is computed for each frame and differences between the adjacent frames are computed. The frame \( f_i \) is considered to be a shot boundary, particularly abrupt shot boundary, if the entropy difference between \( f_i \) and \( f_{i+1} \) is greater than the predefined threshold \( \tau_A \) [2, 3, 5]. It can be returned as

\[
\mathcal{B}(f_i) = \begin{cases} 
\text{True} & \text{if } D(f_i, f_{i+1}) > \tau_A \\
\text{False} & \text{otherwise}
\end{cases}
\]

\( \mathcal{B}(f) \) decides either the given frame \( f_i \) is shot boundary or not, and \( D(\cdot) \) computes the dissimilarity or difference between adjacent frames. The value \( \tau_A \) gives better precision with poor recall if it is high and better recall with poor precision if it is low, as shown in Figure 2. During experiment, the value of \( \tau_A \) is set experimentally which gives high F-score.

3.2. Key Frame and Local Key Point Descriptors Extraction. Let \( \mathcal{S} = \{s_1, s_2, \ldots, s_n\} \) be the set of all shot boundaries. One or set of key frame(s) from each shot are selected. There are a number of possibilities to select representative frames, a.k.a. key frames, from each shot. Since the entropies are already computed in shot boundary process, so entropy based key frame selection criteria are used [3].

For any given shot, \( s_j \in \mathcal{S} \), the frame with maximum entropy is selected as key frame. It has been shown experimentally that if the entropy is larger, the contents in the frame are dense which represents the shots precisely. The
shots are now represented by key frames and denoted by \( S = f_1, f_2, \ldots, f_n \), where \( f_i \) denotes the key frame of shot \( s_i \).

Two images can be matched if they are similar based on some similarity criteria. Similarity is computed between the features of the images. SIFT [27] is widely used as an image feature for various applications of computer vision and video processing. For any given image, key points are detected and those key points are represented by some descriptors such as SIFT. On average, there are 2-3 thousand key points on single image which makes matching very expensive and exhaustive, as single image is represented by 2-3 thousand feature vectors. To match two images of size \( 800 \times 600 \) each, it takes 2 seconds on commodity hardware on average. If one image has to be matched with several hundreds or thousand images then it is not practical to use SIFT or any raw descriptors. Quantization is used to reduce the feature space.

### 3.3. Quantization: BoVW Model

Bag of visual word model is widely used for feature quantization. Every key point descriptor, \( x_i \in \mathbb{R}^d \), is quantized into a finite number of centroids from 1 to \( k \), where \( k \) denotes the total number of centroids, a.k.a. visual words, denoted by \( V = \{ v_1, v_2, \ldots, v_k \} \) and each \( v_i \in \mathbb{R}^d \). Let, say, a frame \( f \) be represented by some local key point descriptors \( X = \{ x_1, x_2, \ldots, x_m \} \), where \( x_i \in \mathbb{R}^d \). In BoVW model, a function \( \mathcal{G} \) is defined as

\[
\mathcal{G} : \mathbb{R}^d \rightarrow [1, k], \quad x_i \rightarrow \mathcal{G} (x_i) \tag{3}
\]

\( \mathcal{G} \) maps descriptor \( x_i \in \mathbb{R}^d \) to an integer index. For given frame, \( f \), bag of visual word, \( \mathcal{F} = \{ \mu_1, \mu_2, \ldots, \mu_k \} \), is computed, \( \mu_i \) indicates the number of times \( v_i \) appeared in frame \( f \), and \( \mathcal{F} \) is unit normalized at the end. Mostly, \( k \)-mean or hierarchical \( k \)-mean clustering is applied, and centroids (visual words), \( \mathcal{V} \), are obtained. The value of \( k \) is kept very large for image matching or retrieval applications; the suggested value of \( k \) is 1 million. The accuracy of quantization mainly depends on the value of \( k \); if the value is small then two different key point descriptors will be quantized to same visual words which will decrease the distinctiveness, or if the value is very large, then two similar key point descriptors which are slightly distorted can be assigned different visual words which will decrease the robustness [28].

In the case of the video segmentation, the scenario is different than the searching or matching one image with set of very large database which have severe image transformations such as illumination, scale, viewpoint, and scene capture at different time. In video segmentation, image is matched with few other images, 4 to 7, in sliding window which contain slightly different contents. The each image in sliding window is a key frame which represents the shot; an example of sliding window matching is shown in Figure 3.

In proposed framework, the value of \( k \) is kept far smaller than the value suggested in the literature [2] without compromising on the segmentation accuracy. During experiment, the value of \( k = 25000 \) gives approximately same accuracy as the value 500000 which is used in our previous work [2]. For the above-mentioned experiment, the value of \( k \) was gradually increased from 5000 to 30000 by the factor of 1000, and it was found that the value \( k = 25000 \) gives approximately same accuracy as of our previous work [2].

### 3.4. Quantization: VLAD Model

VLAD is emerging quantization framework for local key point descriptors [29]. Instead, computing the histogram of visual words, it computes the sum of the differences of residual descriptors with visual words and concatenates into single vector of \( d \times k \). Let \( \mathcal{G}^V \) be VLAD quantization function [30]

\[
\mathcal{G}^V : \mathbb{R}^d \rightarrow v_j \in \mathcal{V}.
\]

\[
x_i \rightarrow \mathcal{G}^V (x_i) = \arg \min_{v \in \mathcal{V}} \|x_i - v\|^2 \tag{4}
\]

The VLAD is computed in three steps:

1. offline visual words are obtained \( \mathcal{V} \).
2. all the key point descriptors obtained from given frame, \( f^X \), are quantized using (4),
3. VLAD is computed for given frame, \( \mathcal{J}_f = \{ j_1, j_2, \ldots, j_k \} \), where each \( j_q \) is \( d \)-dimensional vector obtained as follows:

\[
j_q = \sum_{X : \mathcal{G}^V(X) = v_m} X - v_m \tag{5}
\]

\( \mathcal{J}_f \) is \( d \times k \) dimensional feature. In case of SIFT, \( d = 128 \) and recommended value of \( k \in \{64, 128, 256\} \) [29]. As stated above, video segmentation does not require very large values of \( k \). During experiments, the value of \( k \) for VLAD is 16 and \( \mathcal{J} \) using SIFT is \( 128 \times 16 = 2048 \) dimensional. \( \mathcal{J} \) is unit normalized at the end. The vector is very compact without the loss of accuracy as shown in experiments.

### 3.5. Scene Boundary Detection

Algorithm 1 is used to find the scene boundaries [2]. \( \mathcal{H} \) denotes feature vectors for key frames; the feature vectors are either VLAD or BoVW vectors explained in the previous section. The similarity between two key frames is decided by dissimilarity function \( \mathcal{D} \) which can be computed as follows:

\[
\mathcal{D}(H_i, H_j) = \sum_{q=1}^{k} \min(h_{i,q}^j, h_{j,q}^i) \tag{6}
\]

Two key frames are treated as similar if their \( \mathcal{D}(\cdot) > \tau_s \). The value of \( \tau_s \) is the average of the minimum and the maximum similarities of the similar shots on a subset of the videos used in the experiments. The average of similarity score is widely used as the value of \( \tau_s \). In our experiment the average of similarity scores gives low segmentation accuracy, i.e., 0.713.

### 4. Experiments and Results

Cinematic and drama videos are used for scene boundary detection; list of movies and dramas is given in Table 1. F-score is used as performance metrics for scene boundary detection. There is no benchmark dataset. Two strategies have been used to obtain the ground-truth, first party and third
party ground-truth. First party ground-truth is generated by the authors and third party ground-truth is collected from the experts who have adequate knowledge of shots and scene boundaries [2, 3]. To make ground-truth hinased, third party approach is used in our experiments [3, 5, 26].

The accuracy of proposed system can be seen in Table 1. Our dataset has two different groups with completely different videos. One group consists of cinematic movies with entirely different environment and challenging effects with complex motion of scenes. On the other hand, the second group of data consists of indoor drama serials which are easy to segment compared to cinematic movies because of their simple scene with no challenging effects, that is why then length of the sliding window \( L \) is different for both groups of dataset. The sensitivity of \( L \) can be seen in Figure 4 [2]. In cinematic videos, the scenes are longer and shots are shorter. In just few seconds, there are sometimes more than 20 shots due to different effects and actions. The value of \( L \) is marginally bigger compared on drama types of videos. Though, single value can also be used for all types.

Since the values of \( k \) for VLAD and BoVW are shorter in proposed experiments compared to recommended values which increase the efficiency for similarity computation, the similarity computation by (6) or any other distance is at least \( O(n) \), where \( n \) denotes the dimensionality of the feature. The computation of similarity is faster if the value of \( n \) is shorter, as shown in Figure 5. It can be seen that VLAD is faster than BoVW because VLAD has shorter dimensions compared to BoVW. The recommended value of \( k \) for BoVW is 1000000, as discussed in previous section, whereas in our experiments the value of \( k \) is 25000.

5. Conclusion

Video segmentation is a primary step for video indexing and searching. Shot boundary detection divides the videos into small units. These small units do not give meaningful insight of the video story or theme. However, grouping of similar shots give better insight of the video and this grouping can be treated as video scene, and grouping of similar shots is called scenes.
Table 1: Performance of BoVW and VLAD on cinematic and drama videos.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The Pink Panther (2006)</td>
<td>24</td>
<td>01:32:01</td>
<td>0.91</td>
<td>0.86</td>
<td>0.79</td>
<td>0.88</td>
<td></td>
</tr>
<tr>
<td>Grown Ups (2010)</td>
<td>25</td>
<td>01:40:55</td>
<td>0.90</td>
<td>0.86</td>
<td>0.83</td>
<td>0.88</td>
<td></td>
</tr>
<tr>
<td>The Expendables (2010)</td>
<td>24</td>
<td>01:43:29</td>
<td>0.85</td>
<td>0.82</td>
<td>0.72</td>
<td>0.81</td>
<td></td>
</tr>
</tbody>
</table>

F -score

---

In this paper, we propose framework which uses state-of-the-art searching techniques such as BoVW and VLAD, which is widely used for image and video retrieval, for scene boundary detection. Images, or video frames, are represented by BoVW and VLAD which are very high dimensional feature vectors. We experimentally show that, in the field of scene boundary detection, competitive accuracy can be achieved by keeping the dimensions of BoVW and VLAD to very small. The recommended dimensions for BoVW are 1 million; in our experiments, we just tuned it to be 25000. The recommended dimensions of VLAD are 32768; in our experiments it is tuned to 2048. We exploit the sliding window for shot boundary detection. In very small sliding window, the contents of the video shots do not change drastically, which helps to represent shots by reduced dimensions of BoVW and VLAD.
Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We are thankful to Shin’ichi Satoh from National Institute of Informatics, Japan, Nitin Afzulpurkar from Asian Institute of Technology, Thailand, and Chadaporn Keatmanee from Thai-Nichi Institute of Technology, Thailand, for their expertise that greatly assisted this research.

References
