applied sciences

MDPI

Article

Video Searching and Fingerprint Detection by Using the Image Query and PlaceNet-Based Shot Boundary Detection Method

DaYou Jiang [1] and Jongweon Kim [2,*]

[1] Department of Copyright Protection, Sangmyung University, Seoul 03016, Korea; dyjiang@cclabs.kr
[2] Department of Electronics Engineering, Sangmyung University, Seoul 03016, Korea
* Correspondence: jwkim@smu.ac.kr

Received: 31 July 2018; Accepted: 20 September 2018; Published: 26 September 2018

Abstract: This work presents a novel shot boundary detection (SBD) method based on the Place-centric deep network (PlaceNet), with the aim of using video shots and image queries for video searching (VS) and fingerprint detection. The SBD method has three stages. In the first stage, we employed Local Binary Pattern-Singular Value Decomposition (LBP-SVD) features for candidate shot boundaries selection. In the second stage, we used the PlaceNet to select the shot boundary by semantic labels. In the third stage, we used the Scale-Invariant Feature Transform (SIFT) descriptor to eliminate falsely detected boundaries. The experimental results show that our SBD method is effective on a series of SBD datasets. In addition, video searching experiments are conducted by using one query image instead of video sequences. The results under several image transitions by using shot fingerprints have shown good precision.

Keywords: video searching; video fingerprint; shot boundary detection; PlaceNet; image query; LBP-SVD; SIFT

1. Introduction

With videos becoming more popular, important, and pervasive, video tasks, such as searching, retrieving, tracking, summarization, object detection, and copy detection, are becoming more challenging. Video searching (VS) has been a challenging research topic since the mid-1990s, and video copy detection (VCD) also started at that time [1]. Video fingerprinting is widely employed in VS and VCD. The tendency of VCD has been focused on the extraction of robust fingerprints. The state-of-the-art VCD methods are mostly based on video sequences, and image-query-based VS/VCD technology is still imperfect. Therefore, developing robust fingerprints for VS/VCD by using image queries has great importance.

Video content analysis is the most basic process for different video applications. Table 1 lists some common video features. It describes the characteristics of those features and lists their robustness and frailty for video fingerprint detection.

Most of the features are visual content features, such as color-based, gradient-based, and transform-coefficients-based features, that are global features, and the common features of them are their low complexity extraction and that they are weak under local operations. Local features are local descriptors, such as SIFT (Scale-Invariant Feature Transform) [2], SURF (Speed Up Robust Features) [3], and ORB (Oriented FAST and Rotated BRIEF) [4]. Local features can search for abrupt changes in intensity values and their relationships from their neighboring pixels. Motion-based features represent the temporal relations in video sequences, but in videos that have a few camera activity changes, they cannot represent high-level semantic information better than other features.

These features can be used in many applications, such as recognition, retrieval, indexing, identification, searching, tracking, and filtering. For video, they can be applied for keyframe selection, shot boundary detection, video retrieval, identification, and so on.

Table 1. Examples of common features for image/video processing and their robustness and frailty to image/video transition.

Feature		Description	Robustness	Frailty
Color-based	Color histogram [5]	Color histogram for the intensity image in RGB (Red, Green, Blue) color space or in HSV (Hue, Saturation, Value) color space	Signal processing, Flip, Scaling	Color change, Post-production, or edition
	LBP [6]	Local Binary Pattern (LBP), Texture Spectrum model by computing neighborhood pixels	Signal processing	Color change, Post-production, or edition
Gradient-based	HOG [7]	Histogram of Oriented Gradients (HOG) counts the occurrences of gradient orientation in localized portions of an image	Signal processing	Geometrical transformations
	Edge [8]	Edge of oriented gradients	Scaling, compression	Color change, Post-production, or edition
GIST [9]		A set of spectral dimensions (naturalness, openness, roughness, expansion, ruggedness) that represent the spatial structure of a scene	Scaling, compression	Post-production or edition, cropping
Transform-coefficients-based	DWT [10]	Discrete Wavelet Transform (DWT) coefficients by using a mean value, an STD (Standard deviation) value, and SVD (Singular Value Decomposition)	Compression, Flip	Post-production or edition, blur
	DCT [11]	Discrete Cosine Transform (DCT) coefficients	Scaling	Post-production or edition
Motion-based [12]		Object motion and camera motion operated by a block matching consecutive frame blocks algorithm	Signal processing	Desynchronization
Local descriptors		Descriptors can search for abrupt changes in pixel intensity values	Most geometrical transformations	Luminance change

The state-of-the-art VS and VCD methods are mainly based on shot boundary detection (SBD) approaches because using SBD can process video more efficiently. The situation is that image-query-based VCD and deep-learning network-based SBD technologies have yet to be improved. Therefore, in this paper, we aim at developing a video searching/fingerprint detection system based on deep-learning networks and image queries. Figure 1 shows the overview of the image-query-based video searching and copy detection method. The first step is to build a new places-centric dataset and train a pre-trained PlaceNet for image classification. The places dataset should include both natural and computer-generated places-centric images because in many science fiction films the places are not common. The second step is to segment the videos into shots. The SBD method combines general, local, and deep-learning-based features. Then, shot-based fingerprints are built instead of using all frames or keyframes. Finally, several signal processing and geometric transitions are applied to build image queries for video searching and copy detection. Through these steps, the question of whether the image has been found in the video dataset is assessed and if it has, the location of it in the video is detected.

Our contributions include:

1. Introducing a new deep-learning-based SBD method. The method has three stages: candidate segment selection, places semantic-based segment, and segment verification. The Network is places-centric instead of object-centric.
2. Developing a novel image-query-based video searching/fingerprint detection system.

The paper is organized as follows. Section 2 overviews the previous VCD and SBD methods. Section 3 introduces our SBD method and image-query-based video fingerprint method. Section 4 introduces the evaluation method and presents the experimental results on some famous datasets. Section 5 offers our conclusions.

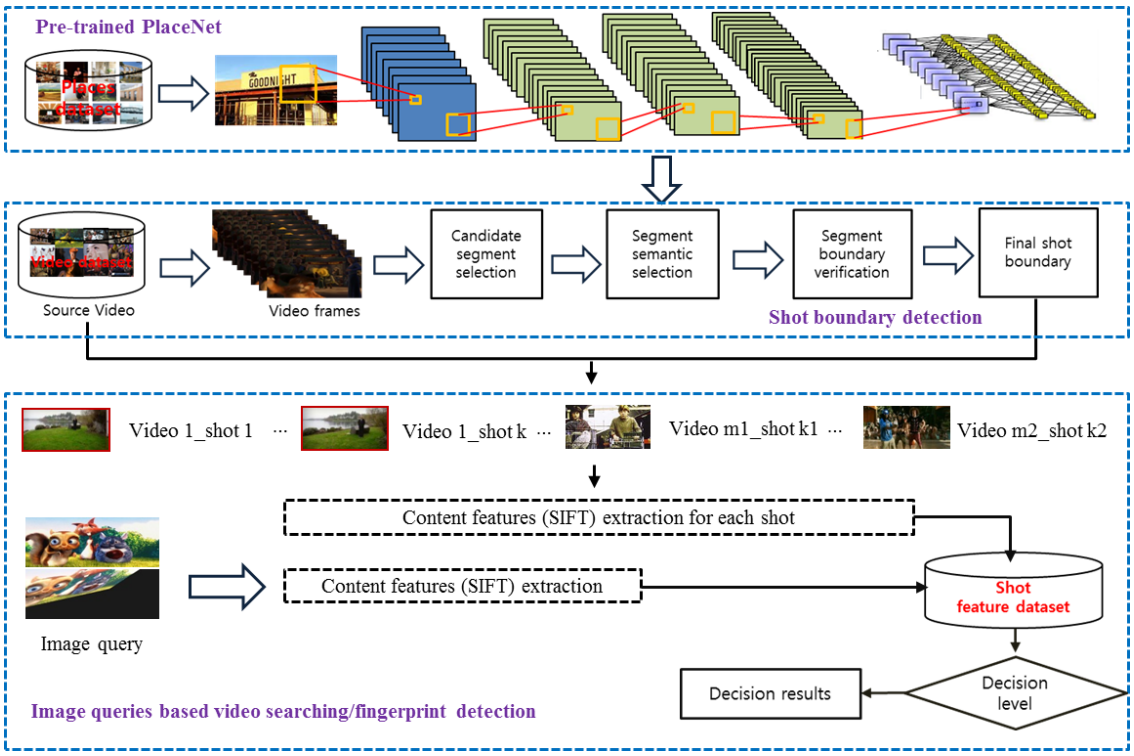


Figure 1. An illustration of the proposed image query based video searching/fingerprint detection process. SIFT, Scale-Invariant Feature Transform.

2. Overview of Related Works

2.1. VCD Approaches

The famous related work is the TRECVID [13] VCD track. TRECVID is the international conference on benchmarking technology for content-based video indexing and retrieval. From 2008 to 2011, the VCD task ran for four years. Before that, deep learning networks had not been proposed, so the methods are all based on low-level content features. As shown in Table 2, the methods can be divided into keyframe-based and shot-based methods. Among them, the keyframe-based methods are mainly used but use of the shot-based methods is on the increase. Furthermore, using a fusion of fingerprints continues to show an upward tendency. Some of those methods use a fusion of features of global content, local content, and deep-learning-based features, and some of those methods use a fusion of features of video and audio.

Table 2. Description of the video copy detection (VCD) approaches.

Presenters/Year	Methods	Characteristics	Based Types
INRIA-LEAR [14]/08	The method uses the SIFT features representing the uniform sampled query frames, uses K-means to generate the visual vocabulary, and uses hamming encoding to generate the candidate set of video segments.	Using keyframe classifiers for video processing to increase the performance. Computationally expensive.	Keyframes
AT&T team [15]/09	The method uses SBD to segment query videos and reference videos at the same time. The first frame of the shot is taken as a keyframe.	Using preprocessing to remove and reduce bad effects of transitions.	SBD
CRIM team [16]/09	The method segments the video files into shots, uses SIFT for feature representation, the L1 distance is used for quantization, and Latent Dirichlet Allocation (LDA) is applied to discrete discriminants over matches.	The Bag of Words (BOW) model is used to eliminate the features that are not representative enough.	SBD
PKU-IDM team [17]/10	The method uses four detectors: two visual local features (SIFT and SURF), one global feature DCT, and one audio feature: Mel-frequency cepstrum coefficients (MFCCs).	A fusion of features: audio and video.	Keyframes
Gupta [18]/11	The method maps each video frame of the test to the closest query video frame firstly, and then moves the query over the test to find the test segment with the highest number of matching frames.	A fusion model reduces the false alarm rate.	Keyframes
Wu [19]/12	The method uses two global features: a pyramid histogram of oriented gradients (PHOG) and GIST; their binary features are quantitated by using the pairwise Euclidean distance.	Uses a sparse random projection method to encode the features	Keyframes
Zhao [20]/13	The method F-SIFT starts by estimating the dominant curl of a local patch and then geometrically normalizes the patch by flipping before the computation of SIFT.	Flip-invariant and can save on computational cost.	Keyframes
Kim [21]/14	The method fuses models of spatial modalities and temporal modalities. The spatial fingerprint consists of DCT coefficients signs in local areas in a keyframe and the temporal fingerprint computes the temporal variances in local areas in consecutive keyframes.	A video shots features and adaptive modality fusion.	SBD
Lu [22]/15	The method is based on local non-negative matrix factorization (LNMF), which is used for shot detection. The VCD has two-stage processing and the Hausdorff distance is used.	The robustness needs to be enhanced.	SBD
Mao [23]/16	The method uses five scene frames from a video for video authenticity.	Can save storage space.	Keyframes
Guzman-Zavaleta [24]/17	The method uses a combination of features: ORB, R&F (Resize and Flip), and a Spectrogram Saliency Map (SSM). The feature of a keyframe requires approximately 3 KB, which has a low-cost extraction.	Low-cost extraction of features and the synergy of lightweight video fingerprints.	Keyframes
Araujo [25]/17	The method uses the frame fisher vector and scene fisher vector for video segments.	The query image is compared directly against video clips in the database.	SBD
Kordopatis-Zilos [26]/17	The method extracts the Convolutional Neural Network (CNN) features from AlexNet [27], VGGNet [28], and GoogleNet [29] first, and then uses the vector aggregation method to make codebooks.	Deep learning based features.	Keyframes

SBD, shot boundary detection.

2.2. SBD Approaches

A video contains a great amount of information at different levels in terms of scenes, shots, and frames. SBD is the first process to simplify applications, such as video indexing, retrieval, and fingerprinting. A shot is a set of continuous frames recorded by a single camera. SBD approaches can be divided into two categories: the compressed domain and the uncompressed domain. Some SBD algorithms are in the compressed domain and they are faster than those methods in the uncompressed domain. However, the uncompressed domain presents more challenges because of the vast amount of visual information in the video frames. Consequently, research on SBD is focused on the uncompressed domain rather than the compressed domain.

In recent years, the SBD approach has made rapid progress. Table 3 lists some representative types of SBD approaches. The approaches are mainly in the following fields: pixel-based [30–32], histogram-based [33–35], edge-based [36,37], transform-based [38–40], motion-based [41], and statistical-based [42,43] approaches. The histogram-based method can be regarded as invariant to local motion or small global motion compared with the pixel-based methods. The edge-based method is simple to conduct. The processing of transform-based approaches usually transforms a signal (frame) from the time (spatial) domain into the transform domain. For motion-based approaches, the motion vectors are computed by block matching consecutive frame blocks. Transitions and camera operations, such as zoom or pan, can be differentiated. For statistics-based approaches, properties such as the mean, median, and standard deviation are often used. There are some other SBD approaches, such as the temporal slice coherency [44], fuzzy rules [45], and two-phased [46] approaches. Due to deep learning having become a hot topic in research work, deep-learning-based approaches have been increasingly applied to SBD works [47–50]. Those methods have accordingly shown better performance than convolutional methods.

Table 3. Description of the SBD approaches.

Type	Presenters	Methods	Characteristics
Pixel-based	Kikukawa [30]	The method uses the sum of the absolute differences of the total pixel with a threshold to locate the cut shot.	Easy and fast, but cannot give a satisfactorily result.
	Zhang [31]	The method uses the preprocessing method of average filtering before detecting shots.	False detection is the main problem.
	Shahraray [32]	The method divides the frame into regions, matches the regions between the current frame and its next frame, then chooses the best matches.	Real-time processing.
Histogram-based	Küçüktunç [33]	The method uses fuzzy logic to generate a color histogram for SBD in the L*ab color space.	Robust to illumination changes and quantization errors, so it performs better than conventional color histogram methods.
	Janwe [34]	The method uses the just-noticeable difference (JND) to map the RGB color space into three orthogonal axes JR, JG, and JB, and the sliding window-based adaptive threshold is used.	Highly depends on the size of the sliding window and parameter values.
	Li [35]	The method uses a three-stage approach: the candidate shots are detected by two thresholds based on the sliding window at first; then, the local maximum difference of the color histogram is used to eliminate disturbances; finally, the HSV color space and Euclidean distance are employed.	Can deal with a gradual change and a cut change in the same way.

Table 3. *Cont.*

Type	Presenters	Methods	Characteristics
Edge-based	Zheng [36]	The method uses the Robert edge detector for gradual shot detecting; the fixed threshold is to determine the total number of edges that appear.	Fast but the performance for gradual transition detection is not good.
	Adjeroh [37]	The method uses locally adaptive edge maps for feature extraction and uses three-level adaptive thresholds for video sequencing and shot detection.	Fast, uses an adaptive threshold, and is slightly superior to the color-based and histogram-based methods.
Transform-based	Cooper [38]	The method computes the self-similarity between the features of each frame and uses the DCT to generate low-order coefficients of each frame color channel for a similarity matrix.	Competitive with seemingly simpler approaches, such as histogram differences
	Priya [39]	The method uses the Walsh–Hadamard operation to extract the edge strength frames.	Simple but the performance should be improved.
Motion-based	Porter [40]	The method uses camera and object motion to detect transitions and uses the average inter-frame correlation coefficient and block-based motion estimation to track image blocks.	High computational cost and has dissolve detection as a weakness.
	Bounthemy [41]	The method estimates the dominant motion in an image represented by a two-dimensional (2D) affine model.	Applicable to MPEG videos.
Statistical-based	Ribnick [42]	The method uses the mean, standard deviation, and skew of the color moments.	Simple but has dissolve detection as a weakness.
	Bendraou [43]	The method uses SVD updating and pattern matching for gradual transitions.	Reduces the process and has good performance in both cut and gradual transition detection
Temporal Slice Coherency	Ngo [44]	The method constructs a spatial-temporal slice of the video and analyzes its temporal coherency. Slice coherency is defined as the common rhythm shared by all frames within a shot.	Capable of detecting various wipe patterns.
Fuzzy-rule-based	Dadashi [45]	The method calculates a localized fuzzy color histogram for each frame and constructs a feature vector using fuzzy color histogram distances in a temporal window. Finally, it uses a fuzzy inference system to classify the transitions.	Threshold-independent and has a weakness against gradual transition types.
Two-phased	Bhaumik [46]	The first phase detects candidate dissolves by identifying parabolic patterns in the mean fuzzy entropy of the frames. The second phase uses a filter to eliminate candidates based on thresholds set for each of the four stages of filtration.	Has good performance for detecting dissolve transitions, uses lots of sub-stages, and is threshold-dependent.
Deep learning based	Xu [47]	The method implements three steps and uses CNN to extract the features of frames. The final decision is based on the cosine distance.	Suitable for the detection of both cut and gradual transitions boundaries; is threshold-dependent.
	Baraldi [48]	The method uses Siamese networks to exploit the visual and textual features from the transcript at first and then uses the clustering algorithm to segment the video.	Uses CNN-based features and can achieve good results.
	Hassanien [49]	The method uses a support vector machine (SVM) to merge the outputs of a three-dimensional (3D) CNN with the same labeling and uses a histogram-driven temporal differential measurement to reduce false alarms of gradual transitions.	Exploits big data to achieve high detection performance.
	Liang [50]	The method extracts the features using the AlexNet and ResNet-152 model. The method uses local frame similarity and dual-threshold sliding window similarity.	Threshold-dependent.

3. Materials and Methods

3.1. PlaceNet-Based SBD Method

3.1.1. Candidate Segment Selection

Generally, a video sequence has a lot of non-boundary frames and several boundary frames. To reduce the computational complexity, selecting the candidate segment is the first step in our scheme. Here, we adopt nine different features for candidate segment selection. We use a short video "Leon" as a test to find out the difference between them. "Leon" has 3920 frames with a frame rate of 24 fps. In the experiment, we let the first frame be the keyframe of each quarter frame rate distance, and then used those features to compute the difference in consecutive keyframes. The feature values in each keyframe are normalized to a total of 1. The dimensionality of each feature and the total computing time for "Leon" are listed in Table 4. The keyframe differences of each feature are represented by the stem plot. The red star marks in each figure label the reference shot location. Comparing the results of these methods with respect to the computing complexity, the Red, Green, Blue (RGB) histogram method is faster than the others and the GIST method is the slowest. The Singular Value Decomposition (SVD)-based HOG and LBP methods are faster than the Hue, Saturation, Value (HSV) histogram method. The Dual-Tree Complex Wavelet Transform (DTCWT) [51] is slower than DWT. Some of the larger differences are not marked by using the edge method, so a lower threshold is needed. HOG-SVD and GIST show better discriminability than the transform-coefficients-based method. Among them, color-based methods, such as C-RGB and Local Binary Pattern (LBP)-SVD, are superior to the others not only in time reduction but also in discriminating the differences. However, the C-RGB method has its discriminatory power reduced for black and white documentary programs. Therefore, LBP-SVD has been chosen for candidate segment selection.

Table 4. The results of candidate segment selection by using different features on "Leon".

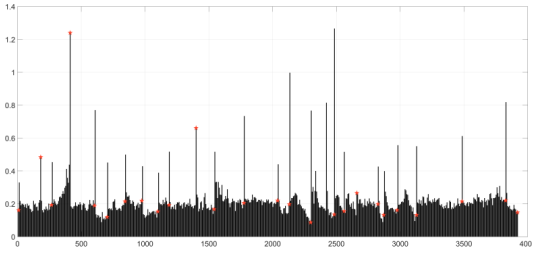
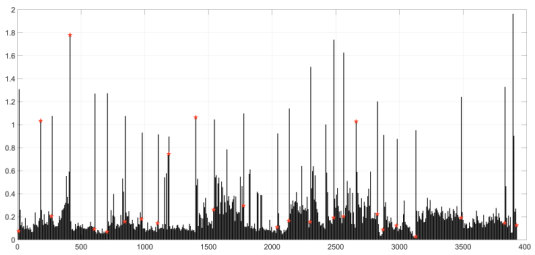
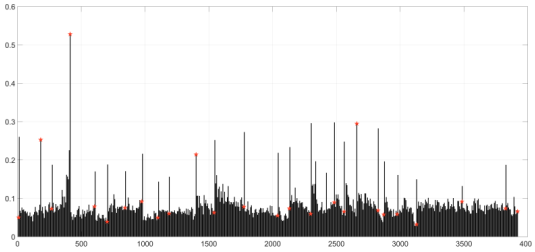
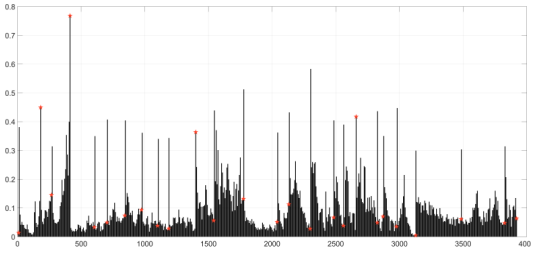
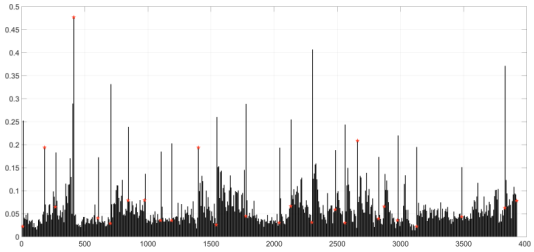
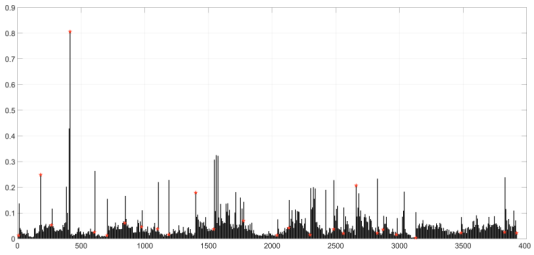
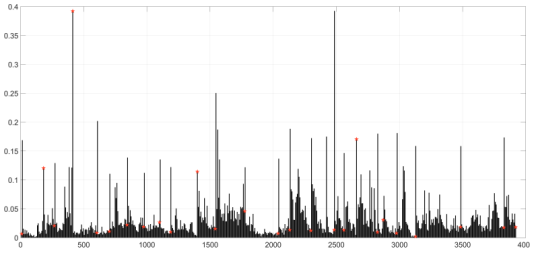
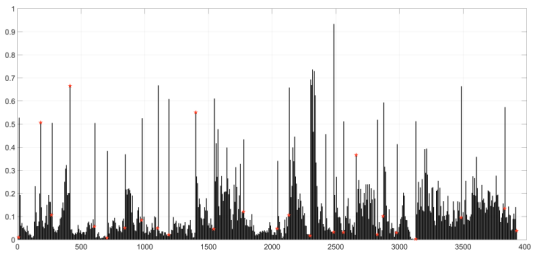
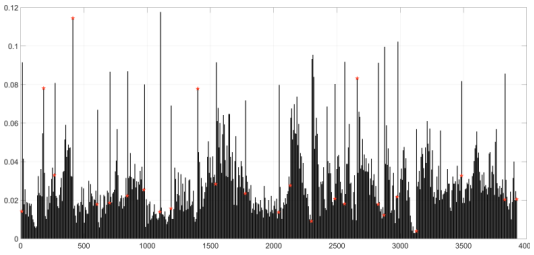
Method	Processing	Length/Time	Frame Difference Plot
C-RGB	Using MATLAB's IMHIST function	512/12.5 s	
C-HSV	Change to HSV space, each of them has eight bins	512/16.9 s	
LBP-SVD	Computing the Local Binary Pattern (LBP) then using singular value decomposition (SVD)	58/14.2 s	

Table 4. *Cont.*

Method	Processing	Length/Time	Frame Difference Plot
GIST	LMgist function	512/82.6 s	
HOG-SVD	Computing HOG then using SVD	31/13.6 s	
DWT	2D DWT, then using the mean, STD, and SV	32/13.2 s	
DTCWT	2D DTCWT, then using the mean, STD, and SVD values	52/17.5 s	
DCT	Block processing DCT using the DC value	64/15.4 s	
EDGE	Edge processing, then using the mean value of a block	64/16.6 s	

The threshold should be adaptive for candidate segment selection because the threshold levels are different for different videos. A lower threshold will lead to a higher recall value and a lower

precision value. Therefore, the threshold should be low enough to recall all shots. Here, we use some different types of videos with different cuts and the gradual transition ratio for training to identify suitable threshold. The training videos' information is listed in Table 5 and the sample video frames are shown in Figure 2.

Table 5. The training videos for threshold selections.

Video Segment	Types	Length	Cuts	Gradual Transitions	Ratio
Scent of a Woman (Tango dance)	Movie	03′53″	34	12	3:1
The Sound of Music	Movie	01′47″	12	1	12:1
Forrest Gump (Start)	Movie	01′44″	10	1	10:1
Run Devil Run	Music	03′28″	95	24	4:1
Donkey and Puss in Boots	Cartoon	02′06″	32	8	4:1
LANEIGE	Advertisement	32″	12	6	2:1
MISSHA	Advertisement	30″	10	2	5:1
Men's Basketball	News	04′09″	6	43	1:7
Edward Snowden	News	04′28″	54	13	4:1

Figure 2. The representative frames from the training videos.

"Scent of a Woman" is a clip of a dance scene and "Men's Basketball" is a basketball broadcast which contains a basketball game clip, so they have many gradual transitions. "LANEIGE" and "MISSHA" are advertisement videos. "Edward Snowden" includes an interview clip that has many camera switches between the interviewer and interviewee.

Here, we take three videos as examples and set three different thresholds. Let M_K be the mean value of the total keyframe differences and S_K be the standard value of the total keyframe differences. The thresholds are set as:

Threshold T1 = M_K, marked by the red line; Threshold T2 = M_K + 0.5 × S_K, marked by the green line; Threshold T3 = M_K + S_K, marked by the blue line.

As seen in Figure 3, a video that has a higher cut shot rate shows better discrimination. As shown in Figure 3c, the annotated shots values are higher than all three thresholds and much higher than the other keyframes. The video "LANEIGE" has a lower cut shot rate than the video "The Sound of Music", so, as shown in Figure 3a, most of the annotated shot values are lower than T2 and all of them are higher than T1. Figure 3b shows that some shot values are lower than T3 and most shot values are higher than T2. Therefore, the threshold should be much lower than T2 to ensure a higher recall rate. In the paper, to reduce the miss rate, we set the threshold to be lower than T1 to select the candidate shot boundaries. After the candidate selection, the total computing time will be greatly reduced.

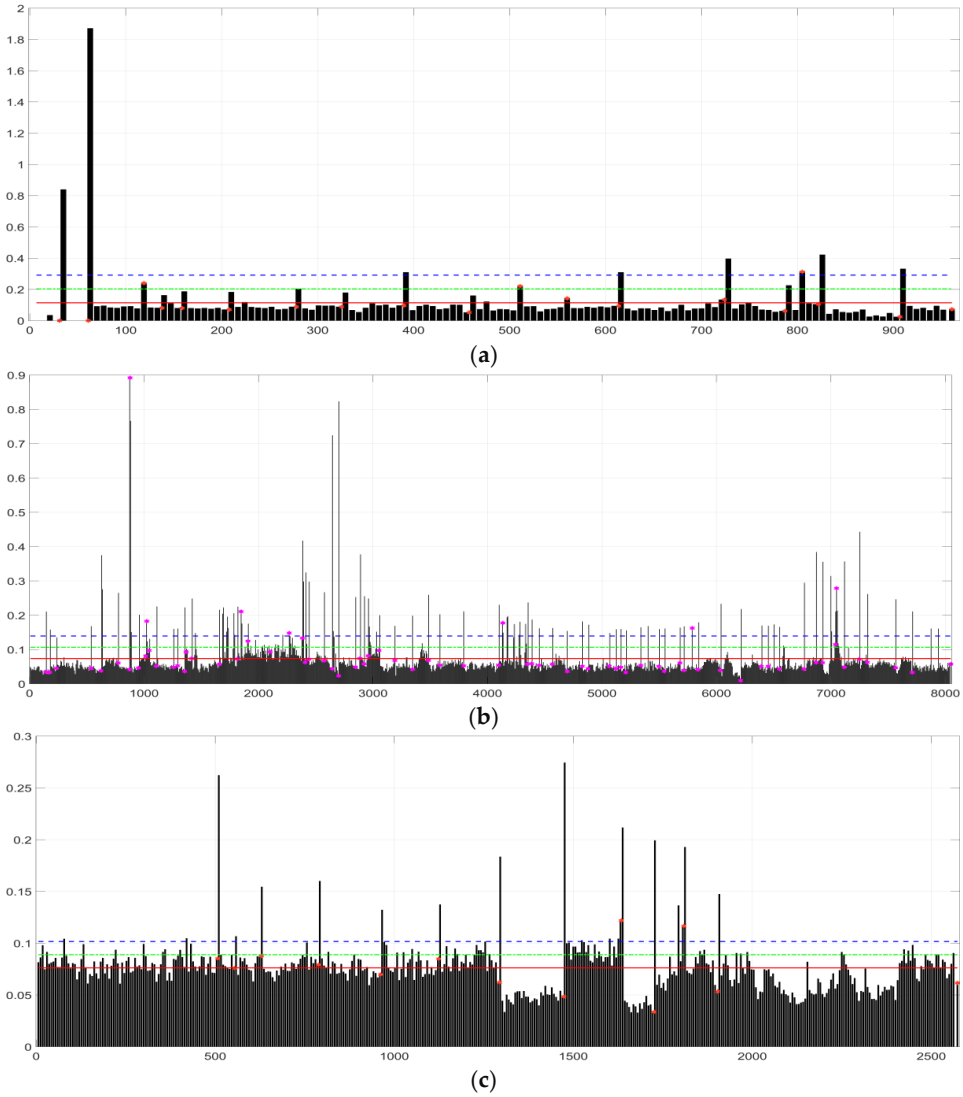


Figure 3. Comparison of the results using LBP-SVD for candidate segment selection. (**a**) LANEIGE, (**b**) Edward Snowden, (**c**) The Sound of Music.

3.1.2. Shot Detection Network

Most of the existing deep-learning-based SBD methods use deep neural networks to extract the features of each frame and then measure the similarity between two contiguous frames. Some researchers use video segments to train shot transition categories. However, the networks are targeted to recognize the object. Generally, the training object categories cannot represent most objects in real life and in videos. Therefore, place categories are used for training instead of object categories.

Dataset Construction

The datasets are mainly based on the Places database [52]. The website Places [53] provides the demo for testing the Place365 dataset. The training network is based on the PyTorch model. Here, we choose two different categories—"forest" and "fields"—for the test. The images are downloaded from the website by using the query word. The query word of each image, from left to right, from top to bottom, respectively, is "forest", "forest cartoon", "forest 3D model", "fields", "fields cartoon", and "fields 3D model". Their classified top-1 categories are shown in Figure 4.

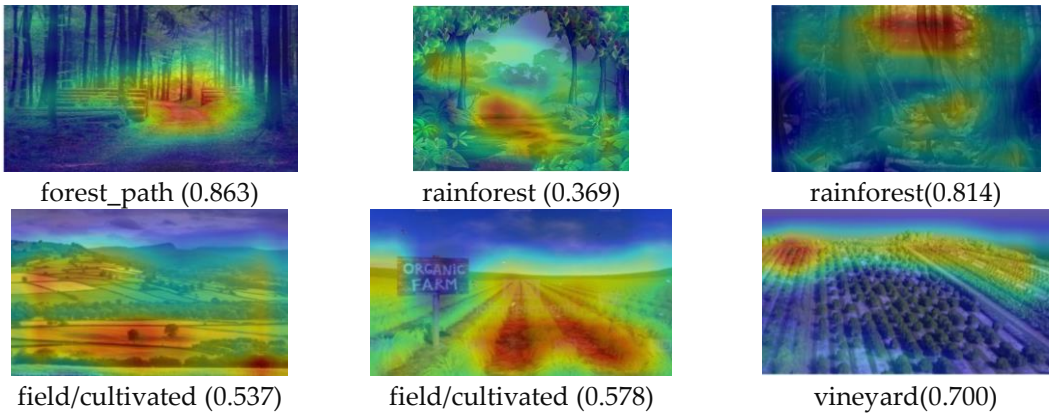


Figure 4. The top-1 classified categories of the test images by using the Places365 demo.

Figure 4 shows that the places classification by using the pre-trained Places365 demo still has some errors, especially for similar categories. What is more, the precision for computer-generated images, such as cartoon and three-dimensional (3D) models, is not as good as that for natural images. Due to the massive variation in videos, in order to get a good result for places classification, the training dataset also should include as many types of images as possible. Therefore, we make the following changes to the Place365 dataset.

1. We add computer-generated images, such as cartoons and 3D model images, to each category.
2. We merge the categories that have common features into one category. For example, the categories of "forest_broadleaf", "forest_needleleaf", "forest_path", and "forest_road" can be merged into the category "forest".

Network Architecture

The popular CNN architectures, such as Alexnet [27], ZFNet [54], GoogLenet [28], VGG [29], ResNet [55], and InceptionResNet-v2 [56], proposed in recent years have been widely used in image and video applications. Among them, InceptionResNet-v2 can perform with the top-1 error of 19.6% and the top-5 error of 4.7% in the ILSVRC 2012 image classification challenge dataset [57]. The ILSVRC dataset contains nearly all object classes, including rare ones, and is uniquely linked to all concrete nouns in WordNet. In Table 6, some deep neural networks are briefly described.

Table 6. The deep neural networks.

Proposed Year	Architecture	Default Input Size	Top-5 Error % ILSVRC 12	Work Contribution
2012	AlexNet	227	15.3	Use of rectified linear units (ReLU), the dropout technique, and overlapping max pooling
2013	ZF Net	224	14.8	Accurate tuning of the hyper-parameters
2014	VGG16/VGG19	224	6.67	Uses multiple 3 × 3 convolutional layers to represent complex features
2014	GoogleNet/Inception	224/299	7.3	Use of 1 × 1 convolutional blocks (NiN), use of a width increase
2015	ResNet50/101/152	224	3.6	Feeding the output of two successive convolutional layers AND also bypassing the input to the next layers
2016	InceptionResNet-v2	299	4.7	Training with residual connections accelerates the training of Inception networks

Considering the performance and the computing complexity, the GoogleNet model and the ResNet-50 model are more efficient. Therefore, the pre-trained GoogleNet and ResNet-50 support

packages from the MATLAB Neural Network Toolbox were selected for training. In the experiments, we took about 540 images (500 natural images and 40 computer-generated images) from each category for the training set, used 60 images (50 natural images and 10 computer-generated images) for the validation set, and used 100 images (80 natural images and 20 computer-generated images) for the test set. For these categories, the number of images is less than the preset number of images for training, validation, and testing. The numbers of these categories' images are distributed into three parts in percentages. The results of the classification accuracy on our used Places dataset are listed in Table 7.

Table 7. The classification accuracy on our validation and test datasets of pre-trained PlaceNet.

	Validation Set of Places Data		Test Set of Places Data	
	Top-1 acc	**Top-5 acc**	**Top-1 acc**	**Top-5 acc**
Places-GoogleNet	37.25%	65.81%	37.04%	65.63%
Places-ResNet-50	45.32%	74.28%	45.18%	74.21%

Table 7 shows that the accuracies are lower than those in the Places365 dataset. This is because the training dataset has less images than Places365 and the number of training iterations is also not large enough. However, the precision for places classification in the videos is not too bad.

3.1.3. Shot Boundary Verification

Only using the pre-trained PlaceNet for SBD is insufficient for common conditions. The classification places inevitably have some errors compared to the ground-truth data. In Figure 5, the classified categories are listed and they are actually in a shot.

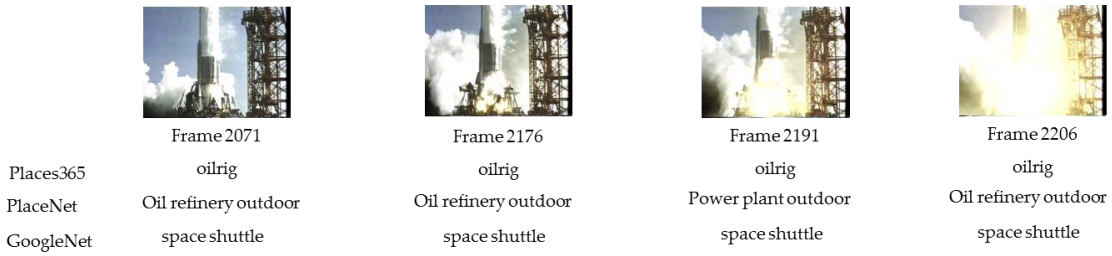


Figure 5. The classification results of PlaceNet, GoogleNet, and Places365 for the images in a shot.

Considering the consistency results, both Places365 and the Object-centric GoogleNet show better performance than PlaceNet, even though they have classified the wrong category. PlaceNet shows the right category in frame 2071, frame 2176, and frame 2206; however, as the middle frame, frame 2191 shows a different category. In this situation, the shot boundary should be verified.

At this stage, we use SIFT matching for verification to reduce false detections of shot boundaries. This process is done after using the pre-trained PlaceNet to extract the places category of the candidate segment boundaries. Here, the threshold of the shot boundary decision is transformed to the threshold of the matched numbers. If the matched number values are less than the threshold, the adjacent two candidate boundaries, which are assigned to different place categories, are truly different. If the matched number values are larger than the threshold, PlaceNet has wrongly classified the places and the current candidate shot boundary will change to the next candidate boundary.

Usually, matches for image content pairs that have little relation to each other are rare no matter how high the match threshold. As shown in Figure 6, the number of matches between them is greater than the threshold value. So, PlaceNet has made a false detection at the second stage, and SIFT matching can eliminate this falsity by feature matching.

Figure 6. The results of the shot boundary verification.

3.2. Image-Query-Based Video Searching

The related approaches have been described in Section 2.1. In this section, we study the method for video searching and fingerprint detection. Generally, the fingerprint should follow the main properties below:

1. Robustness. It should have invariability to common video distortions.
2. Discriminability. The features of different video contents should be distinctively different.
3. Compactness. The feature size should be large enough to retain the robustness.
4. Complexity. The computing complexity should be simple enough.

The local features are the first choice when generating video fingerprints since they can be used directly and can also be quantized by applying a quantization method. Here, we employ 12 different local features for image matching. Some of them are in VLfeat [58] and OpenCV. An image pair comprises the original image and its corresponding distorted image. The local descriptors are listed in Table 8 [59–67].

Table 8. The list of local features.

ID-Features	Feature Detector	Feature Descriptor	ID-Features	Feature Detector	Feature Descriptor
1-SIFT	VL_SIFT	VL_SIFT	7-DAISY [62]	OPENCV_SURF	OPENCV_DAISY
2-SURF	OPENCV_SURF	OPENCV_SURF	8-LATCH [63]	OPENCV_BRISK	OPENCV_LATCH
3-BRISK [59]	OPENCV_BRISK	OPENCV_BRISK	9-KAZE [64]	OPENCV_KAZE	OPENCV_KAZE
4-FREAK [60]	OPENCV_BRISK	OPENCV_FREAK	10-ASIFT [65]	VL_SIFT	VL_SIFT
5-MSER [61]	OPENCV_MSER	OPENCV_SURF	11-BF [66]	VL_SIFT	VL_SIFT
6-ORB	OPENCV_ORB	OPENCV_ORB	12-GMS [67]	OPENCV_ORB	OPENCV_ORB

The image matching results are shown in Figure 7. The numbers that are marked in the images are the ID of the descriptors. In the experiments, the image pairs are resized to 256 $\times$ 256 pixels and the threshold of the match is set as 2.0.

Figure 7 shows that the noise, contrast changes, cropping, rotation, and brightness changes transforms have less of an effect than horizontal flip transition. The BRISK, FREAK, MSER, and LATCH features under the flip and projection transitions show bad performance. In total, SIFT, SURF, DAISY, and KAZE show invariant characteristics to those transformations. Considering the computing time and storage, many researchers choose SURF as their first choice and some people choose DAISY and KAZE, but most people have chosen SIFT due to its invariant properties. Therefore, in this paper, we use the SIFT features to represent the fingerprint of the video shots and query images.

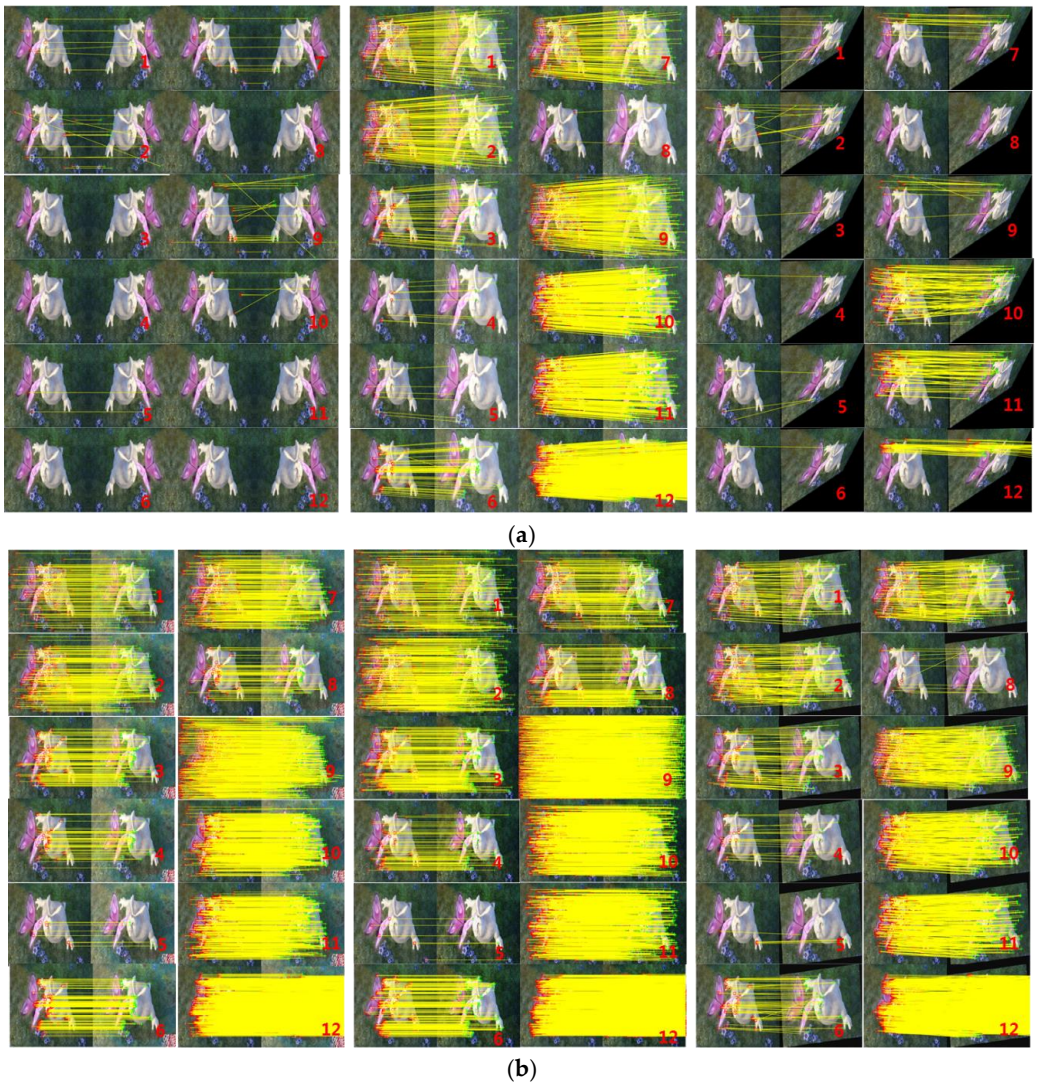


Figure 7. The results of the matching of image pairs under different feature descriptors. (**a**) The transforms from left to right: Horizontal flip; crop plus bright; Projection; (**b**) The transforms from left to right: picture add post; noise plus contrast; rotation plus crop plus noise.

4. Experimental Results and Analyses

4.1. Evaluation Methods

We use precision, recall, and the F_β score to evaluate our method's performance for shot boundary detection. Recall is the ratio of correctly identified shot boundaries to the number of ground-truth shot boundaries. Precision is the ratio of correctly identified shot boundaries to the total detected shot boundaries. The F_β score is used to balance the precision and recall and a higher β value will give more importance to high precision values. In the paper, the F1-score is used. The metrics are defined as follows:

$$\text{Recall} = (\text{Correctly\_identified}) / (\text{Correctly\_identified} + \text{Missed\_identified}) \quad (1)$$

$$\text{Precision} = (\text{Correctly\_identified}) / (\text{Total\_detected}) \quad (2)$$

$$\text{F}_\beta = ((\beta^2 + 1) * \text{Precision} * \text{Recall}) / (\beta^2 * \text{Precision} + \text{Recall}) \quad (3)$$

4.2. Experiment on Shot Boundary Detection

4.2.1. Open Video Scene Detection (OVSD) Dataset

Here, we use the videos in the OVSD dataset [49] to compare our proposed method with the Filmora software [68]. The OVSD dataset is presented for the evaluation of scene detection algorithms and its shot boundary annotations are also given. The ground-truth scene annotations are provided by using a movie script. It consists of five short videos and a full-length film. Recently, a dataset extension, including 15 new full-length videos, has also been uploaded, but it only provides the scene annotations. Information on the OVSD dataset is listed in Table 9. Among the data, "Bunny" has a more vivid color than the other animated movies.

Sample frames from the videos "Big Buck Bunny", "Cosmos Laundromat", and "Sintel" are shown in Figure 8.

In Table 8, the test video name, numbers of manual shots, shots numbers detected by using the proposed method, shots numbers detected by using the Filmora software, and total frame numbers are listed. A comparison of the results of the test videos is also displayed in Table 10.

Table 9. Annotations of the OVSD dataset.

Name	Duration (hh:mm:ss)	Number of Frames	Number of Shots	Video Size
Big Buck Bunny	00:08:08	11,726	130	1920 × 1080
Cosmos Laundromat	00:09:59	14,394	94	1920 × 804
Elephants Dream	00:09:22	13,490	130	1024 × 576
Tears of Steel	00:09:48	14,111	136	1920 × 800
Sintel	00:12:24	17,858	198	1024 × 436
Valkaama	01:33:05	139,133	713	1280 × 720

Figure 8. The representative frames from the three videos of the Open Video Scene Detection (OVSD) dataset.

Table 10. The examples of the SBD results.

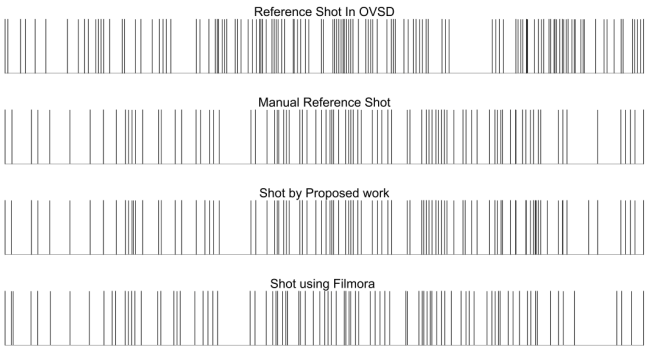
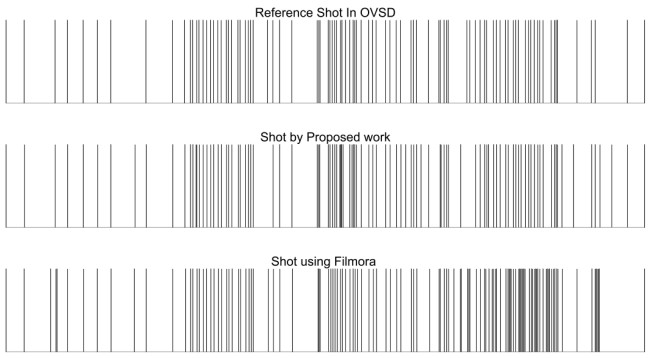
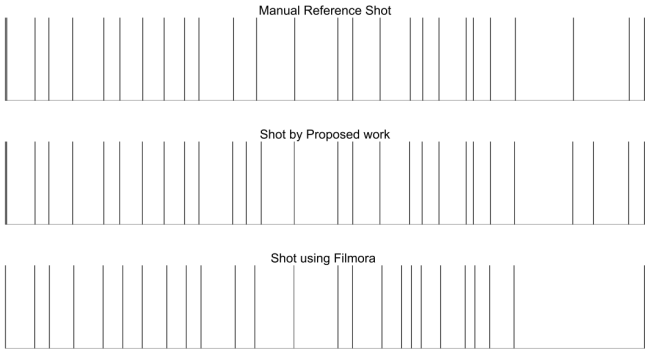
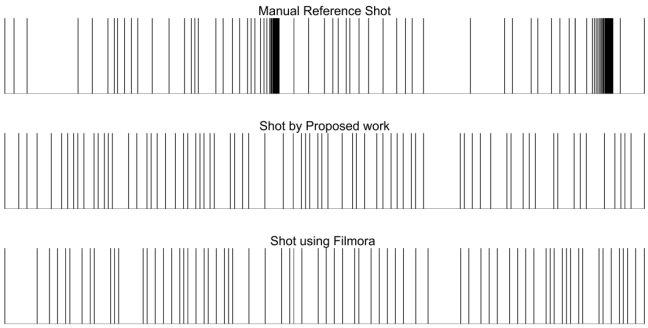
Video information	Shot Plot
Name: "Big Buck Bunny.mp4"; Manual OVSD: 130; Manual: 91; Proposed: 99; Filmora: 90; Frames: 11,726	

Table 10. *Cont.*

Video information	Shot Plot
Name: "Cosmos Laundromat.mp4"; Manual: 94; Proposed: 104; Filmora: 124; Frames: 14,394	
Name: "Leon.mp4"; Manual: 26; Proposed: 28; Filmora: 24; Frames: 3920	
Name: "Gangnam Style.mp4"; Manual: 108; Proposed: 74; Filmora: 70; Frames: 3771	

The open access video editor Filmora offers an advanced feature that can automatically split a film into its basic temporal segments by detecting the transitions between shots in a video. From Table 10, it can be seen that our proposed SBD method is similar to Filmora and has a slightly lower miss rate. The number of Filmora-detected boundaries is not less than the frame rate of the video; however, our proposed method uses a step that is a quarter of the frame rate. Since the shot annotations of the video "Big Buck Bunny" and "Cosmos Laundromat" are created by a script, the accuracy of the shot location has some difference to the real frames. Therefore, we use the manual annotation of "Big Buck Bunny" (marked as 91*) instead of the annotation in the OVSD dataset. The video "Leon" has many more cut shots than gradual shots and the distance between them is larger than the frame rate, so the shot boundary is clear. The video "Gangnam Style" is a music movie, and it has many cut shots with a distance that is less than half the frame rate, so our method and Filmora are unable to detect those shot changes.

Next, we use those videos' SBD to evaluate the performance of the proposed method. Due to the video "Valkaama" being too long, in the experiment, we only chose the second 10 min for the test. A boundary detected by the algorithm was said to be correct if it was within a quarter of the number of frame rate frames of a boundary listed in the baseline. Strictly speaking, the deviation for acceptance is

a little higher because the cut shots happen only during two neighboring frames. However, considering the longer duration of gradual shots, such as dissolves, fade-ins, fade-outs, and wipes, in the proposed method, the smallest step of the shots is not less than a quarter of the number of frame rate frames, so the deviation for acceptance should be higher than that value. The results of SBD for the OVSD dataset are listed in Table 11.

Table 11 shows that the shot boundaries of the videos "Bunny", "Cosmos", and "Valkaama" can be extracted because they have many cut shots. The video "Elephants Dream" is a 3D Computer-Generated Imagery (CGI) animated science fiction video, so most of the scenes in it are hard to match with a venue from a natural environment. This shows the weakness of our place network. The video "Sintel" is also a computer-animated film, but it has lots of action. Therefore, the large gradual shots that were brought on by the abundance of activities could make the shot boundary hard to detect.

Table 11. The results of SBD for the OVSD dataset of our method.

Name	Shots	Detected	Correct	False	Miss	Precision	Recall
Big Buck Bunny	91*	99	85	14	6	0.86	0.93
Cosmos Laundromat	94	104	85	19	9	0.82	0.90
Elephants Dream	130	146	113	33	17	0.77	0.87
Tears of Steel	136	156	125	31	11	0.80	0.92
Sintel	198	221	174	47	24	0.79	0.88
Valkaama	85	93	79	14	6	0.85	0.93
Total	734	819	661	158	73	0.81	0.90

4.2.2. BBC Planet Earth Dataset

The dataset for the BBC's educational TV series Planet Earth [48] has 11 videos. Sample images are shown in Figure 9. All of the videos are approximately 50 min in duration.

Figure 9. The representative frames of the BBC Planet Earth dataset.

Their information is listed in Table 12. Here, we select the first 10 min of each for the experiments. The results are also listed in Table 12.

Table 12. The results for the BBC Planet Earth Dataset.

Id-Name	Total Shots	Shots in 10 Min	Detected Shots	Correct Shots	False Shots	Miss Shots	Precision	Recall	F1-Score
01_From_Pole_to_Pole	445	71	86	66	20	5	0.767	0.929	0.8402
02_Mountains	383	111	117	103	14	8	0.963	0.928	0.9452
03_Ice Worlds	421	84	98	77	11	7	0.786	0.917	0.8464
04_Great Plains	472	86	94	78	16	8	0.830	0.907	0.8668
05_Jungles	460	63	77	57	20	6	0.740	0.905	0.8142
06_Seasonal_Forests	526	88	108	74	34	14	0.685	0.841	0.7550
07_Fresh_Water	531	99	104	90	14	9	0.865	0.909	0.8864
08_Ocean_Deep	410	65	76	57	19	8	0.750	0.877	0.8086
09_Shallow_Seas	366	65	68	58	10	7	0.853	0.892	0.8720
10_Caves	374	71	72	67	15	4	0.931	0.944	0.9374
11_Deserts	467	72	71	65	6	7	0.915	0.903	0.9090
Total	4855	875	971	792	179	83	0.816	0.905	0.8580

4.2.3. TRECVID 2001 Dataset

The TRECVID 2001 Dataset [69] is mostly used for shot boundary detection. The reference data of the transitions are assigned to four different categories: cut, dissolve, fade-in/out, and other. In the

test dataset, their percentages are respectively 65%, 30.7%, 1.7%, and 2.6% [70]. Here, we take some videos from the dataset for comparison experiments. Information on the videos and the results of SBD by using the proposed method are listed in Table 13.

Samples of the video frames are shown in Figure 10.

Table 13. Information on the test video and results of shot detection.

Id-Name	Total Frames	Total Shots	Cut Shots	Gradual Shots	Cut Precision	Cut Recall	Cut F1-Score	Gradual Precision	Gradual Recall	Gradual F1-Score	Total F1-Score
anni005	11,364	65	38	27	0.94	0.96	0.95	0.87	0.86	0.87	0.91
anni009	12,307	103	38	65	0.92	0.86	0.89	0.92	0.82	0.87	0.87
BOR03	48,451	242	231	11	0.87	0.95	0.91	0.80	0.82	0.81	0.91
BOR08	50,569	531	380	151	0.88	0.92	0.90	0.90	0.84	0.87	0.89
NAD53	25,783	159	83	76	0.86	0.93	0.89	0.88	0.86	0.84	0.88
NAD57	12,781	67	44	23	0.96	0.95	0.94	0.88	0.87	0.86	0.93
Total	161,255	1167	814	353	0.88	0.93	0.90	0.89	0.84	0.86	0.91

Figure 10. Samples of the frames from the test TRECVID 2001 dataset.

Here, to demonstrate the accuracy of our scheme, we also conduct comparison experiments. The results of the comparison are listed in Tables 14 and 15.

Table 14 shows that the proposed method has better performance than the correlation-based algorithm, the keener-correlation-based algorithm, and the edge-oriented-based algorithm. Table 15 shows that the proposed method is better than the methods that do not use deep-learning features. Additionally, our place-centric network-based SBD method has similar performance to the compared method that uses an object-centric network. For example, our method has a higher F1-score than the method that uses an object-centric network in cut shot detection in the anni005 video and in gradual shot detection in the anni009 video.

Table 14. The comparison results of the cut shot detection.

Id-Name	Correlation Based [71]		Kernel-Correlation [72]		Edge-Oriented [37]		Proposed Method	
	Cut Precision	Cut Recall	Cut Precision	Cut Recall	Cut Precision	Cut Recall	Cut Precision	Cut Recall
anni005	0.87	0.89	0.71	0.64	0.87	0.91	0.94	0.96
anni009	0.86	0.94	0.81	0.78	0.87	0.93	0.92	0.86
BOR08	0.85	0.88	0.60	0.83	0.86	0.91	0.88	0.92
NAD53	0.79	0.94	0.69	0.84	0.81	0.97	0.86	0.93

Table 15. The comparison results of shot detection methods.

Id-Name	Pre-Processing [73]				SVD [74]				CNN [75]				Proposed Method			
	Cut-Pr	Cut-Rc	Gra-Pr	Gra-Rc	Cut-Pr	Cut-Rc	Gra-Pr	Gra-Rc	Cut-Pr	Cut-Rc	Gra-Pr	Gra-Rc	Cut-Pr	Cut-Rc	Gra-Pr	Gra-Rc
anni005	0.95	0.95	0.75	0.96	0.88	0.97	0.67	0.80	1	0.90	0.89	0.86	0.94	0.96	0.87	0.86
anni009	0.88	0.74	0.92	0.69	0.88	0.74	0.68	0.80	1	0.82	0.94	0.73	0.92	0.86	0.92	0.82

4.3. Experiment on Image-Query-Based Video Searching

4.3.1. ReTRiEVED Dataset

The ReTRiEVED [76] Dataset was created to evaluate methods that require video quality assessment in transmissions. The ReTRiEVED dataset contains 176 test videos obtained from 8 source videos by applying the transmission parameters listed in Table 16.

Table 16. The video attacks simulated by the ReTRiEVED dataset.

Attacks	**Parameters**						
Delay (ms)	100	300	500	800	1000	/	/
Jitter (ms)	1	2	3	4	5	/	/
Packet Loss Rate (%)	0.1	0.4	1	3	5	8	10
Throughput (Mbps)	0.5	1	2	3	5	/	/

Samples of the source videos and test videos are shown in Figure 11. Here, we used this small dataset to assess the robustness of features we used to guard against possible video distortions.

Figure 11. Samples of the videos in the ReTRiEVED dataset. The upper row shows the original videos and the bottom shows the attacked videos.

Here, we compared the SIFT features against some related methods: CST−SURF [77], CC [78], and {Th; CC; ORB} [24] for video retrieval. In Table 17, the average detection F1-scores are presented.

Table 17. The average F1-scores for the different transmission attacks in the ReTRiEVED dataset.

Method	**Attacks**			
	Delay	**Jitter**	**Packet Loss Rate**	**Throughput**
SIFT	1	1	1	0.9610
CST–SURF	0.1740	0.3737	0.2889	0.2121
CC	0.8932	0.9777	0.9730	0.9335
{Th; CC; ORB}	0.9996	0.9930	0.9940	0.9495

The experimental results obtained from the methods CST-SURF, CC, and {Th; CC; ORB} are adopted from the paper [24]. The CST-SURF method uses the difference of the SURF key point numbers in stable frame pairs to generate normalized differences as features. The CC method uses color correlation in the divided non-overlapping blocks of each frame. In the {Th; CC; ORB} method, "Th" represents "Thumbnail", which is designed as a global feature to resist against flipping transformations.

In the experiment, since the test videos are short enough, we used the combined features of the selected frames to represent the features of each video. The videos' keyframes are selected at the step of half of the frame rate and they are downsized to 64 × 64 pixels. In the retrieval process, we used UBCMATCH to match the features of the tested videos and the source videos. The number of matches is regarded as the similarity value. Since the match numbers differ greatly under different thresholds, we use the 12-step threshold values to find better conditions for SIFT. Figure 12 shows that the Positive

Predicted Value (PPV) of ReTRiEVED using the SIFT descriptor under the throughput transmission is lower than that of other transitions.

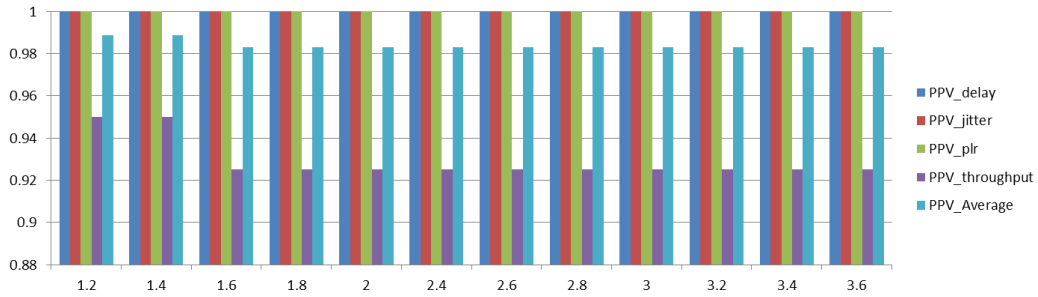


Figure 12. The Positive Predicted Value (PPV) of ReTRiEVED using the SIFT descriptor under different UBCMATCH thresholds.

4.3.2. CNN2h Dataset

The CNN2h dataset [79] is composed of 2 h of CNN video. The dataset provides 139 query photos with a ground-truth query. The photos with geometric and photometric distortions are taken with mobile phones and tablets from displays showing the video. For pair-matching experiments, 2951 true and 21,412 false ground-truth matching pairs of query images and database frames are also provided. In our experiments, we only consider image-to-video matching and ignore the pair-matching experiments. Sample query image and video frames are shown in Figure 13.

Figure 13. Examples of the query images and ground-truth database frames.

The test videos are down-sampled at 10 fps and have 72,000 frames in total. To search for the image in the video, we first use the proposed SBD to segment the videos into shots, and then we use the feature matching method to match the query photo features and the shot feature dataset. The retrieved numbers of video frames for each query image are shown in Figure 14.

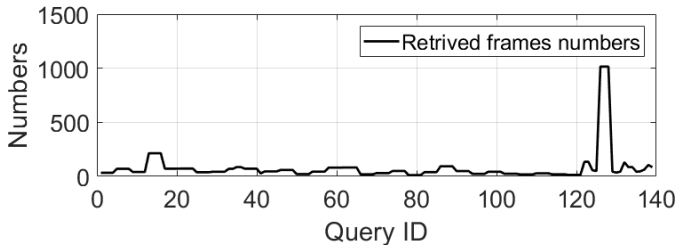


Figure 14. The number of retrieved video frames for each query image.

The retrieval accuracy depends on the accuracy of the shot boundaries. In order to reduce the miss rate, in the experiment, the skip sliding should be set to a lower value. We take five frames for a sling; after three-stage shot boundary detection, we obtain a total of 2864 shots and then use the shot features and query features for feature matching. About 2–3 frames will be missed due to gradual transitions.

4.3.3. Experiment on Our Video Searching and Fingerprint Detection Dataset

The source video dataset for video searching and fingerprint detection was composed of 110 videos, including 4 videos in the OSVD datasets; the other videos are mostly music clips. Information on the video data is shown in Figure 15.

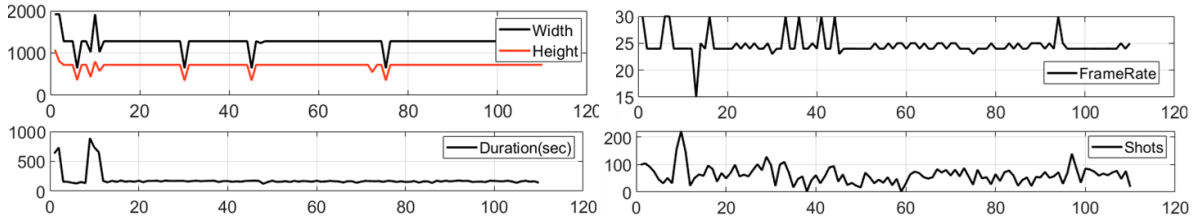


Figure 15. Information on the size, frame rate, duration, and detected shots in our test dataset.

Video distortions can be made by using signal processing, geometric transformations, and desynchronization. To search for and detect a fingerprint of the video in a higher accuracy by using the features of a single image or shot sequence, the proposed method must stand against most of those various distortions. Due to our method being based on a query image, the common attacks for an image are considered. Attacks on audio and video desynchronization are not included. Common transformations are listed in Table 18.

Table 18. Examples of common video transformations.

ID_VT	Video Transformations	Parameters
VT1	Flip	Horizontal flip
VT2	Contrast change	10%
VT3	Noise Addition	Gaussian: mean = 0; variance = 0.01
VT4	Brightness change	10%
VT5	Cropping	10% of the frame border
VT6	Rotation	+5°
VT7	Geometric projection	θ = 1°, projective 2d = ([cos(θ), −sin(θ), 0.001; sin(θ), cos(θ), 0.001; 0, 0, 1]);
VT8	Picture in picture	Original video resized to 90% at the front, background image randomly used
VT9	Picture fusion	original video and background images are added, background alpha value = 0.4
VT10	patterns insertion	Patterns are random images and occupy 2.25% of the area of the Original video

To test the performance of our image-to-video method, a combination of transformations is used. The processing orders and samples of the attacked images of each attack type are also shown in Table 19. The processing parameters of each attack are the same as shown in Table 19.

For the query images, we took 200 random frames for each video from the OVSD dataset. For other videos, because their durations are less than 3 min, we take 20 random frames from each of those videos; otherwise, we take 250 poster images from the website as false examples. So, there are a total of 3170 original query images. Then, we apply the transforms listed in Table 18 to make the source video search task difficult. Finally, 34,870 query images are generated. We used two thresholds in the feature matching and decision stage. The first threshold is the UBCMATCH threshold; the other one is the matched numbers threshold for fingerprint detecting decisions. The parameters in the experiment are listed in Table 20.

Table 19. The performed attacks on the test images for video searching.

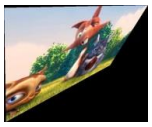
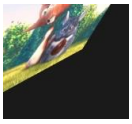
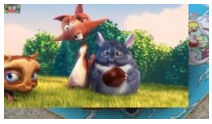
ID_AT	Attack Types	Sample Attacked Image	ID_AT	Attack Types	Sample Attacked Image
AT1	VT1		AT6	VT5→VT6→VT3	
AT2	VT2→VT3		AT7	VT7	
AT3	VT1→VT2→VT3		AT8	VT7→VT4→VT5	
AT4	VT4→VT5		AT9	V10→VT8	
AT5	VT1→VT4→VT5		AT10	VT9→V10	

Table 20. The experimental parameters.

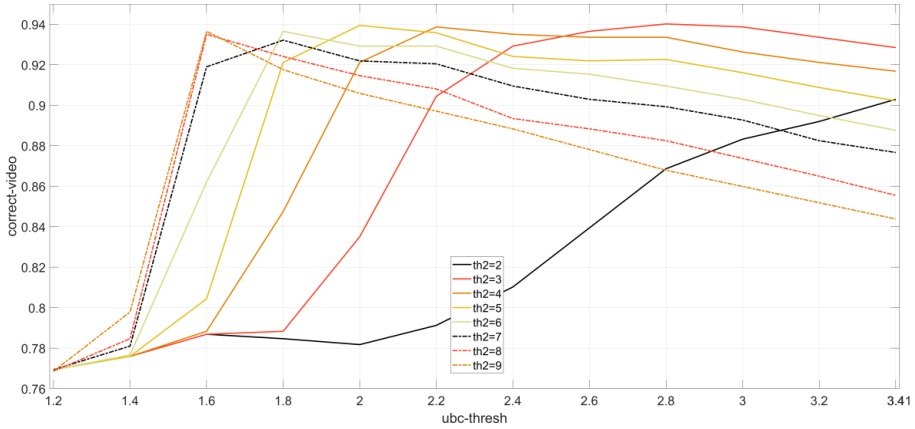
Parameters	Value
Image resize for feature extraction	128 × 128; 256 × 256
Select frames in a shot	three frames (first frame, middle frame, last frame); step skip frames (step skip by a quarter of the frame rate)
Using different features	SIFT; Speed Up Robust Features (SURF)
UBCMATCH threshold1	[1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4]
Matched numbers threshold2	[2, 3, 4, 5, 6, 7, 8, 9]

Results of Image-Query-Based Video Searching Experiments without Transformations

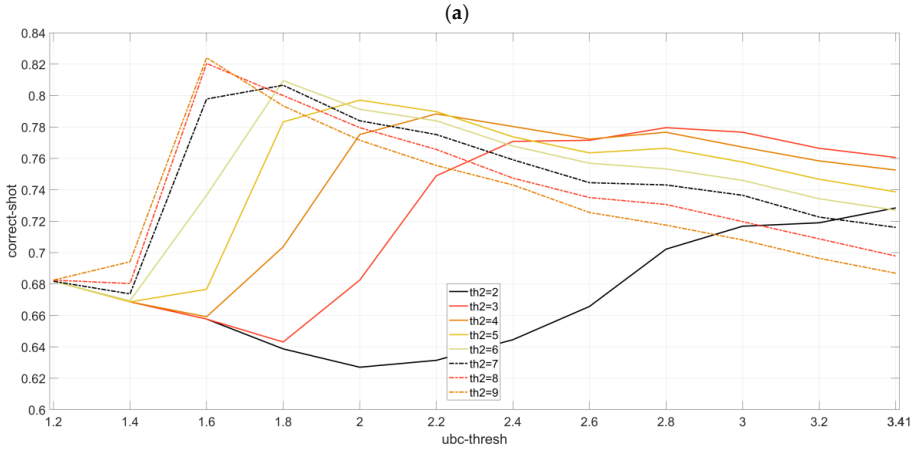
Here, the experiments are conducted under the parameters listed in Table 20. Additionally, image queries from the source videos without transformations are taken for the experiments. The goal is to study the effects of those factors on image-query-based video searching. To simplify the comparison experiments, a control group with the parameters of a 128-pixel image size and application of the step skip frames method and the SIFT descriptor is used to compare the experimental results for different factors. The results of the video searching and shot searching are shown in Figure 16. Shot searching detects a shot's location in a particular video, so the accuracies of it are lower than video searching. Compared to video searching, the shot searching accuracies have decreased by at least 0.1. One reason is that a video may include many similar shots, especially for interview videos. When the UBCMATCH threshold is lower than 1.6, there will be more false alarms; as a result, the searching results are much lower. When the UBCMATCH threshold is higher than 3, there will be lower recall; therefore, the searching results show a declined tendency. The highest video searching accuracy is 0.94 and the shot searching accuracy is 0.82. However, under most conditions, the shot searching accuracies are less than 0.8.

The first comparison experiment was conducted by using a bigger image size: 256 × 256. A larger image size conveys more content information; in theory, the video searching results from using

features extracted from larger-size images will be larger. The only difference in the control group is the image size. The results of video searching and shot searching are shown in Figure 17. Compared to Figure 16, the video searching accuracies and shot searching accuracies are both increased. The shot searching accuracies rise to a greater extent than the video searching accuracies. According to the results, using a larger-size image for feature extraction improves the performance by 0.01~0.03 on average compared to using a smaller-size image. However, the computing time under a larger-size image has been increased a lot not only for feature extraction but also for video searching.

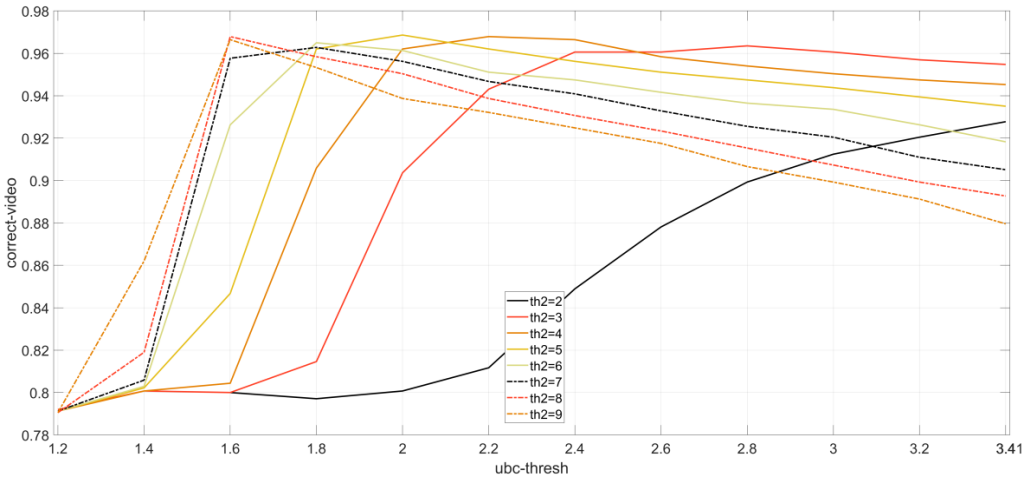


(**a**)



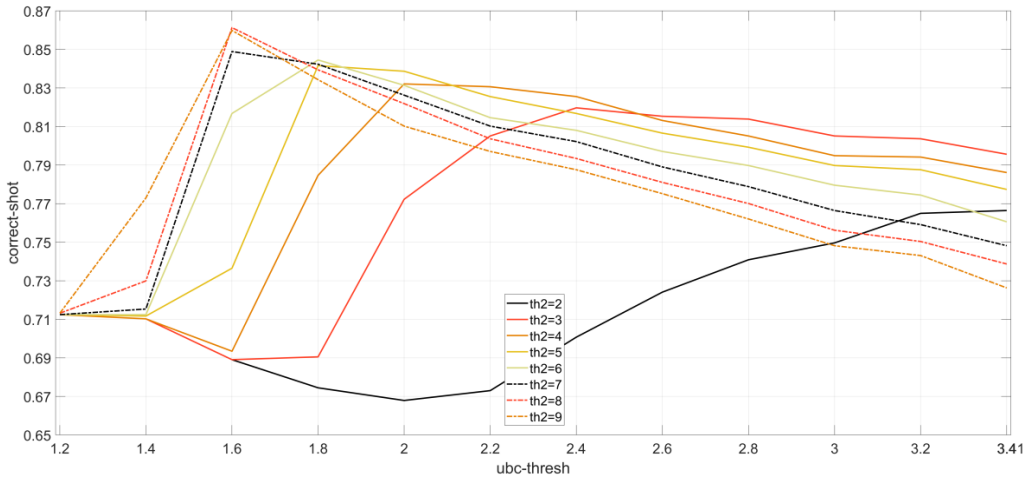
(**b**)

Figure 16. The experimental results with the control group. (**a**) Video searching; (**b**) Shot searching.



(**a**)

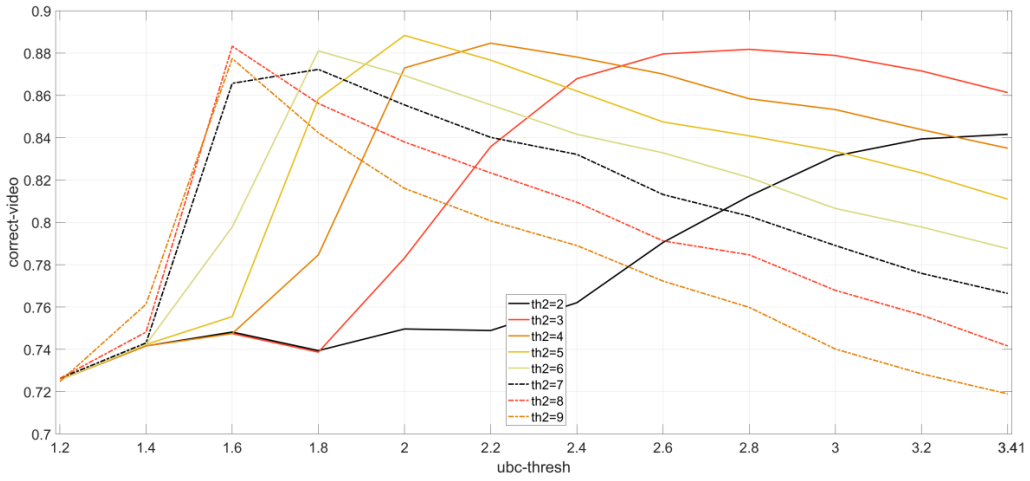
Figure 17. *Cont.*



(**b**)

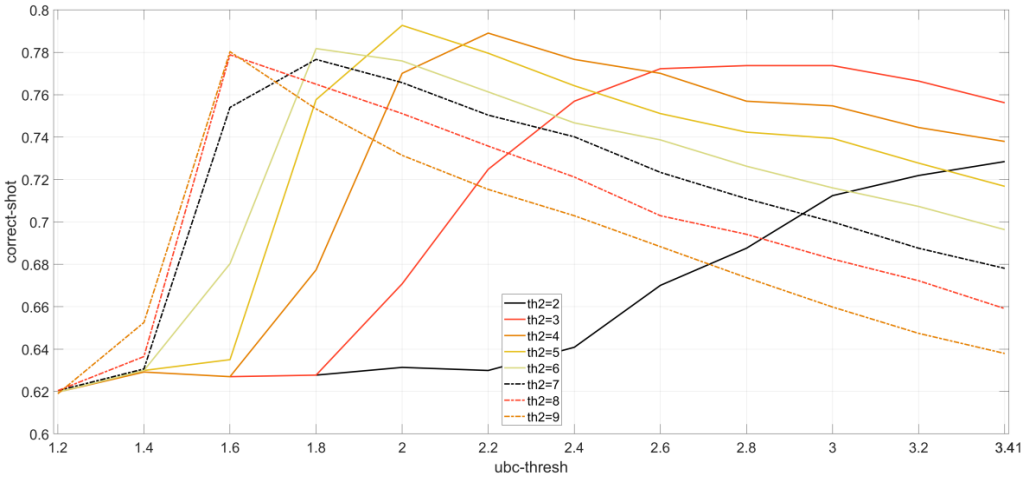
Figure 17. The experimental results by using different size images. (**a**) Video searching; (**b**) Shot searching.

The second comparison experiment adopted different frame selection methods to select the frame numbers in a shot for feature extraction. The only difference in the control group is the use of the three frames method instead of the step skip frames method. The results of video searching and shot searching are shown in Figure 18. Compared to Figure 16, the results show that three-frames-based feature selection for a shot is much worse than step skip frames-based feature selection. The highest video searching accuracy of the three frames method is lower than 0.9 and its downward trends fall faster. The video searching accuracies have decreased by at least 0.05 and the highest value has dropped by 0.07. The shot searching accuracies have also gone down. The highest shot searching accuracy is less than 0.8. Although the three frames method is simpler and has a shorter computation time, the results of it are not satisfactory, especially on video searching. Consequently, the step skip frames method should be considered instead of the three frames method in further experiments.



(**a**)

Figure 18. *Cont.*



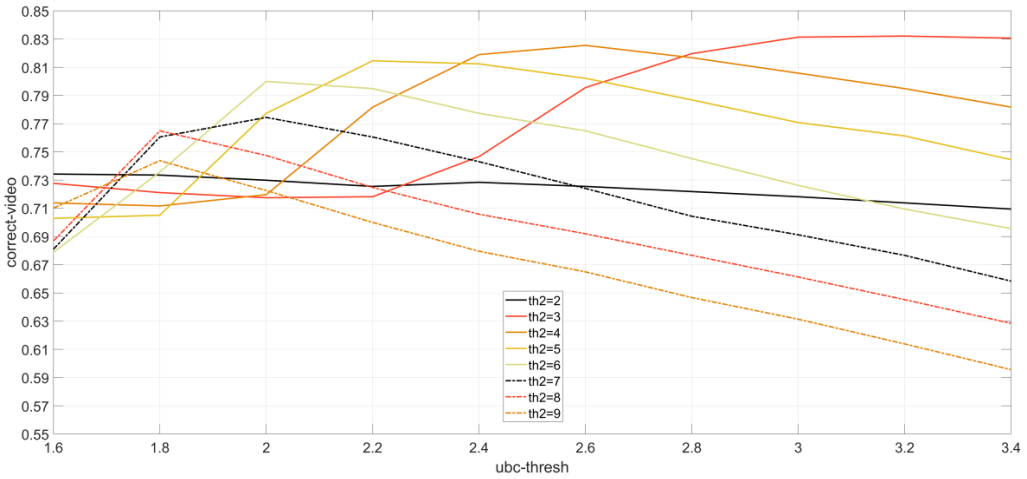
(**b**)

Figure 18. The experimental results using the three frames method. (**a**) Video searching; (**b**) Shot searching.

The third comparison experiment used different feature descriptors. Since the SURF descriptor is faster and also works well in many applications, it can be used instead of SIFT. The results of video searching and shot searching are shown in Figure 19. To show the result more clearly, the results under lower UBCMATCH thresholds (1.2 and 1.4) are not shown. Compared to Figure 16, the results show that the SIFT method has far better performance than SURF. Both the video searching accuracies and shot searching accuracies of SURF have dropped at least 10%. The highest video searching accuracy is less than 0.85 and the highest shot searching accuracy is less than 0.7. Consequently, although the SURF descriptor is faster, the SIFT descriptor should be adopted to achieve better performance.

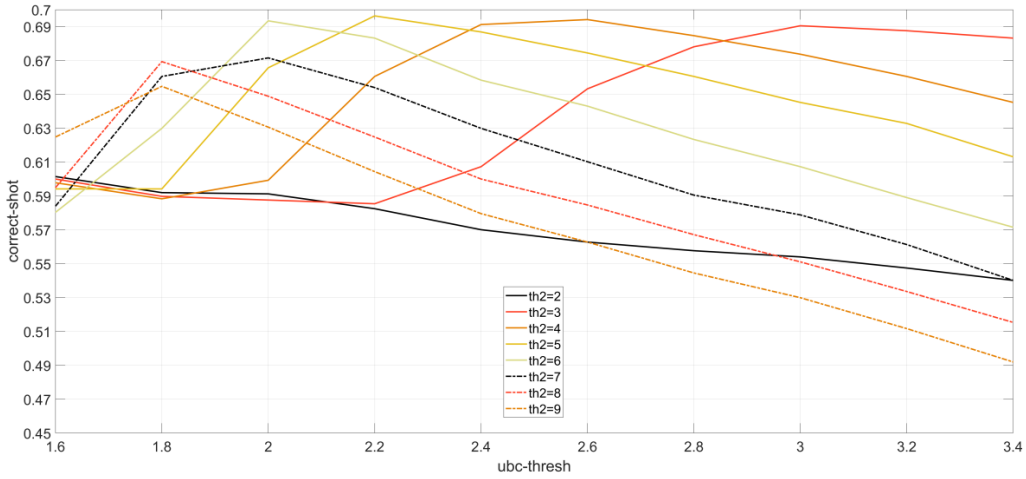
In conclusion, considering the accuracy and computing time, the control group is the better one among them for video searching and shot searching.

The two thresholds also have a big effect on the final performance. When the threshold1 value is lower, the threshold2 value should be higher accordingly. As shown in Figures 16–19, the best thresholds for shot searching are different from the best thresholds for video searching. Additionally, the best thresholds also depend on the feature descriptors and video types. When using the SIFT descriptor, under the threshold1 value of 1.6 and the threshold2 value of 8 or 9, the video searching and shot searching accuracies are higher than under most other conditions.



(**a**)

Figure 19. *Cont.*

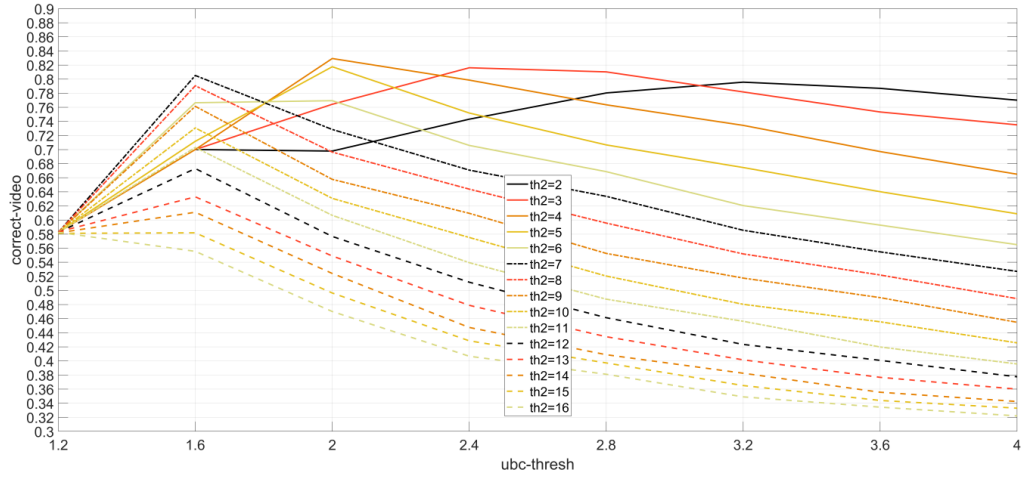


(**b**)

Figure 19. The experimental results by using the SURF method. (**a**) Video searching; (**b**) Shot searching.

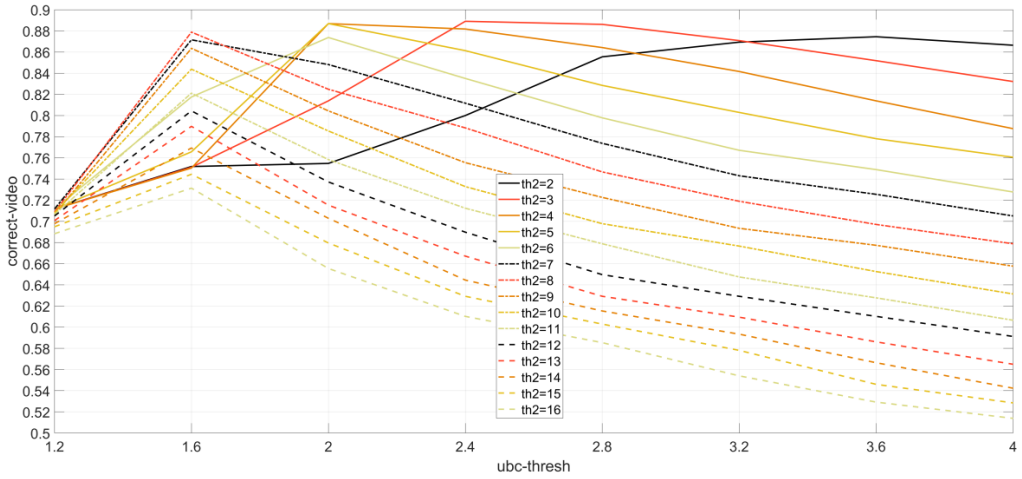
Results of Image-Query-Based Video Searching Experiments with Transformations

From the above experiments, the threshold1s less than 1.6 have bad performance. Additionally, when the threshold1 value is larger than 3.0, the plot lines begin to go down. Considering image distortion situations, higher thresholds may not return any matches. However, in the experiments, the threshold range should be large enough. Therefore, we set threshold1 with a step size of 0.4 and decide to enlarge threshold2 from 2 to 16. During the shot feature generation stage, the frames are resized to 128 × 128 and the step skip frames method is used. Transformations, such as flip, contrast change, noise addition, brightness change, cropping, rotation, and geometric projection, are commonly used for video copy detection. In addition, a picture in a picture, a fusion of pictures, and a pattern insertion are also used. To show the experimental results under transformations, we select five representative transformations: AT1, AT2, AT4, AT7, and AT9. AT1 employs flip. AT4 applies contrast change and noise addition. AT4 applies brightness change and cropping. AT7 uses geometric projection. AT9 was transformed using a picture in a picture and a pattern insertion. Figure 20 shows the results of video searching by using one query image under these transformations.

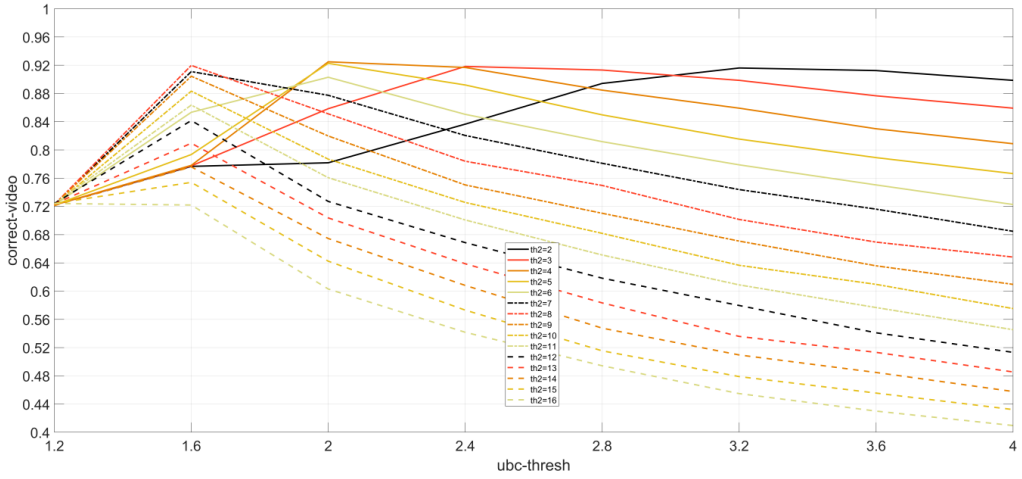


(**a**)

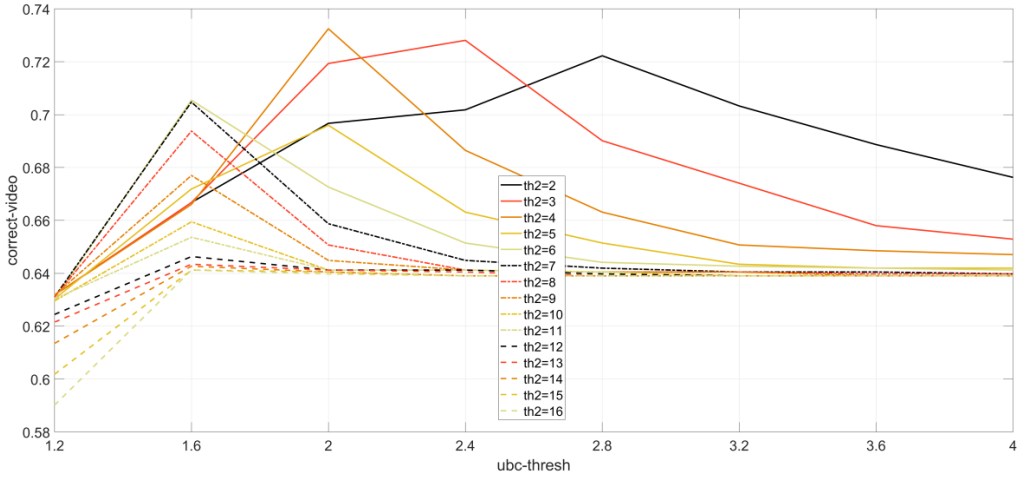
Figure 20. *Cont.*



(**b**)

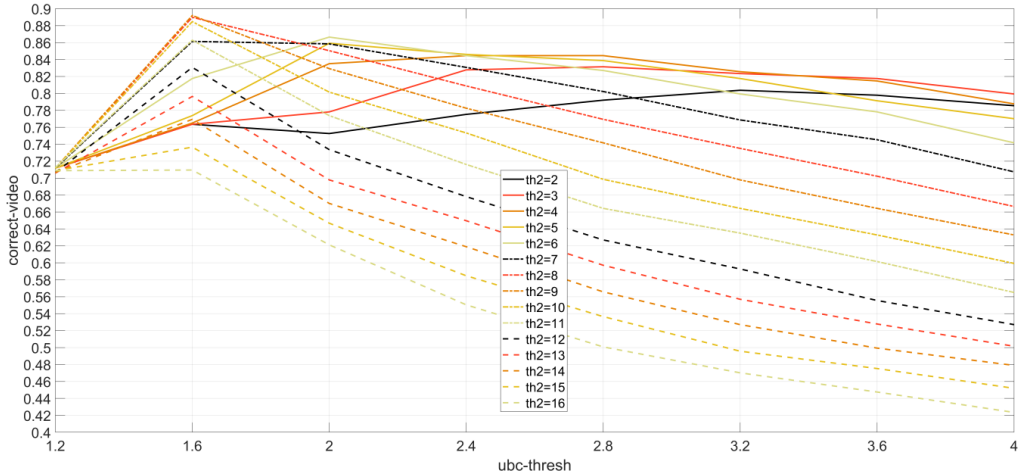


(**c**)



(**d**)

Figure 20. *Cont.*

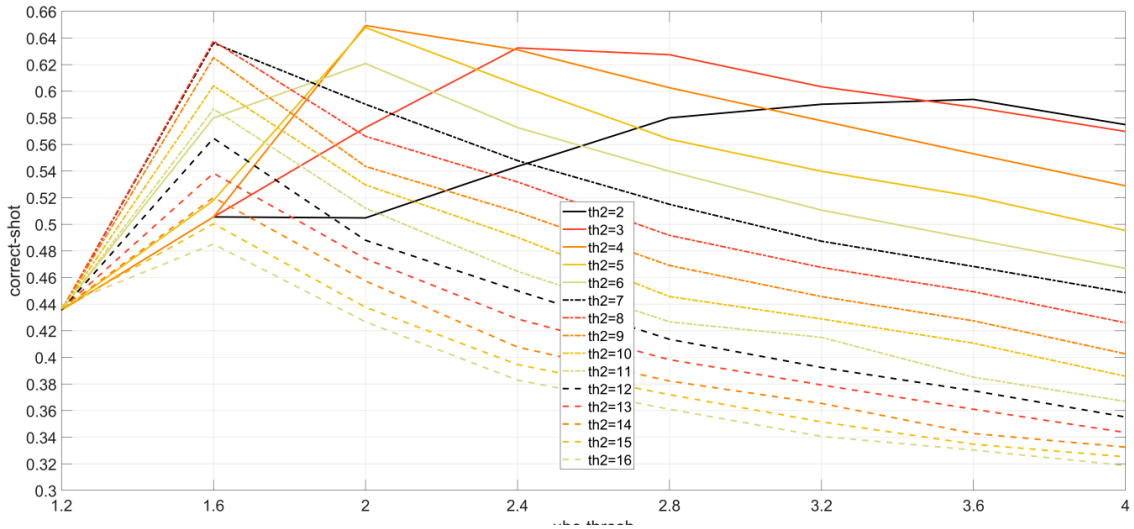


(**e**)

Figure 20. Examples of video searching by using one image query under the transformations. (**a**) Transformation AT1; (**b**) Transformation AT2; (**c**) Transformation AT4; (**d**) Transformation AT7; (**e**) Transformation AT9.

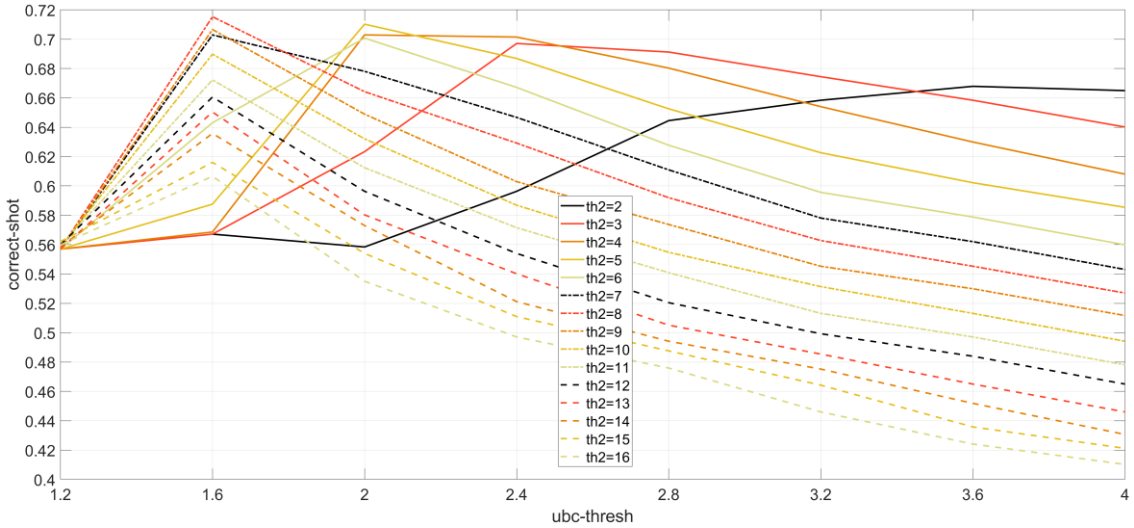
As seen in Figure 20, the different kinds of attacks lead to different results. Compared to Figure 16a, the accuracies of the flip transform and the geometric projection transform have dropped severely. The results of AT1 and AT7 are decreased by nearly 0.1 and 0.2, respectively. The attacks noise addition, contrast change, picture in a picture, and pattern insertion have not changed the image seriously as the video searching accuracies have dropped by no more than 0.06. In addition, the performances of the cropping and bright change transformations have decreased slightly.

However, for query-image-based shot location detection, the accuracies have all dropped very seriously. Without transformations, the shot searching accuracy can reach 0.82. Under the transformations, the accuracies have dropped by at least 0.1. As shown in Figure 21, the results of shot searching are much lower than corresponding results for video searching. Compared to Figure 16b, the results of AT1 and AT4 have dropped by 0.17, and AT7 has decreased by 0.3. Even AT2 has dropped by 0.1. Although the shot searching results are lower, the video searching results under several transformations can be accepted for video searching and video copy detection.

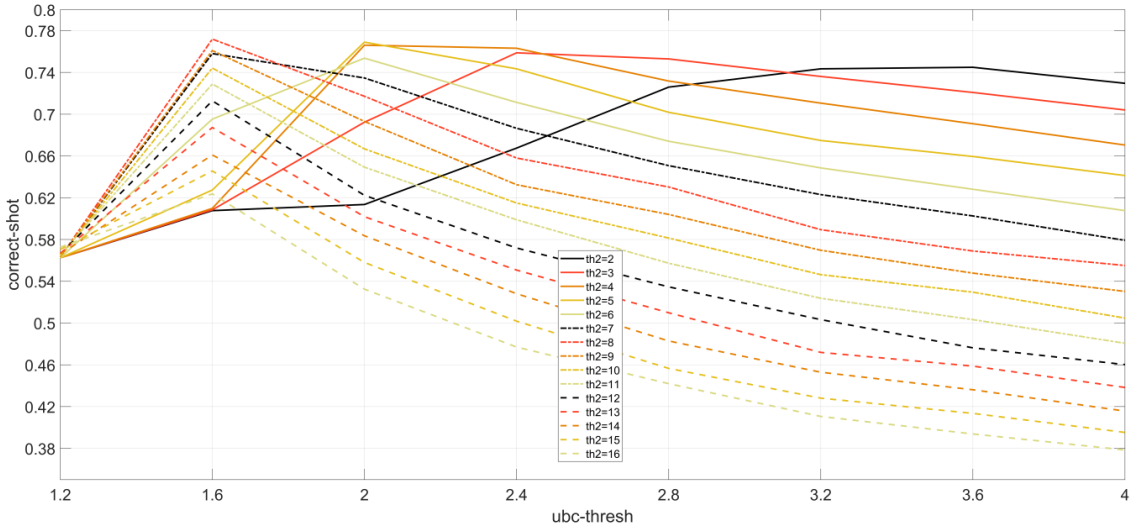


(**a**)

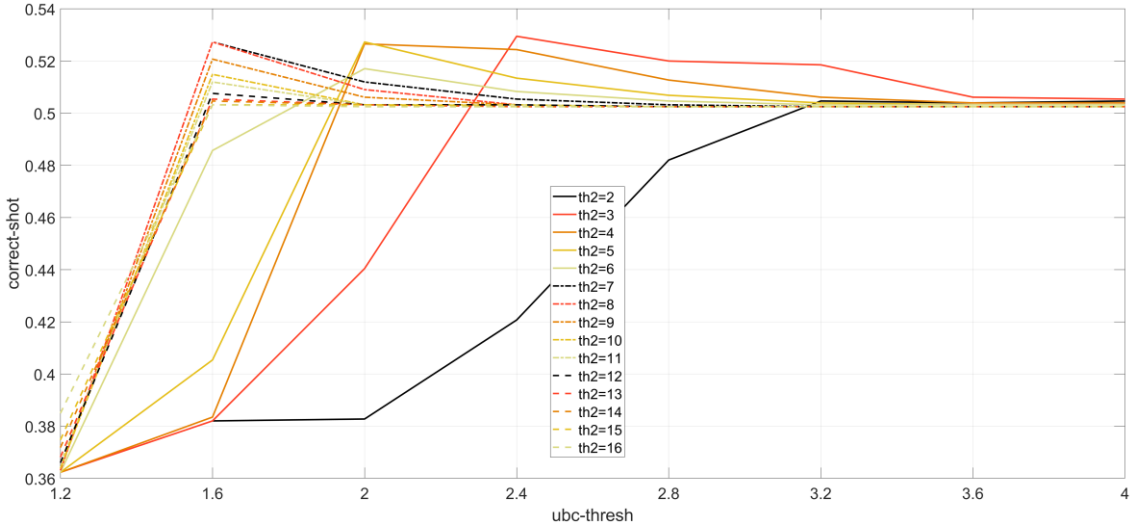
Figure 21. *Cont.*



(**b**)

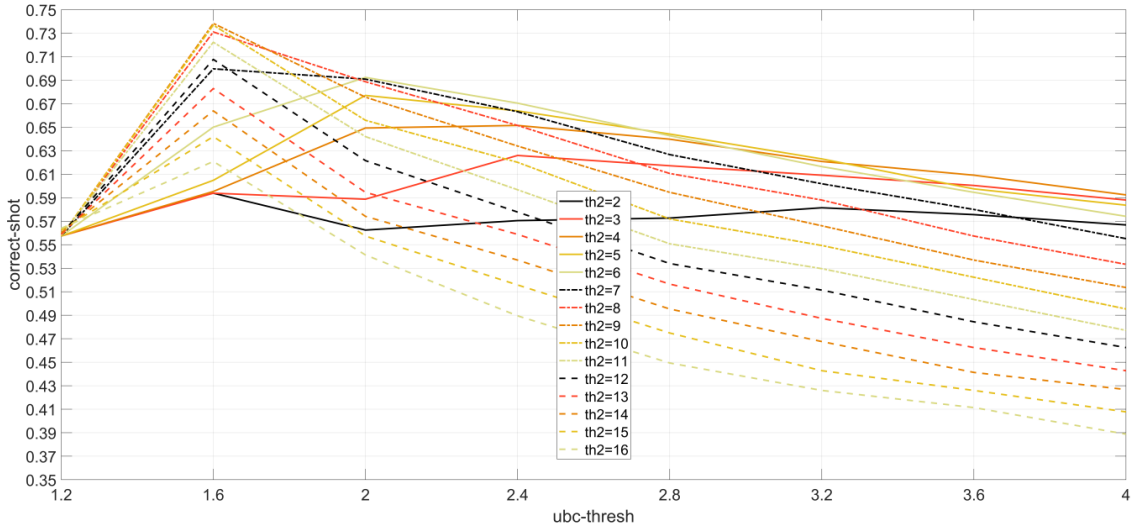


(**c**)



(**d**)

Figure 21. *Cont.*



(**e**)

Figure 21. Examples of shot searching by using one image query under the transformations. (**a**) Transformation AT1; (**b**) Transformation AT2; (**c**) Transformation AT4; (**d**) Transformation AT7; (**e**) Transformation AT9.

As seen from Figures 20 and 21, threshold1 shows better performance under the values of 1.6, 2, and 2.4. When threshold1 is 1.6, the better choices of threshold2 are 7, 8, and 9. When threshold1 is 2.0, the better choices of threshold2 are 4, 5, and 6. When threshold1 is 2.4, the better choices of threshold2 are 3 and 4.

5. Conclusions

In this paper, we have proposed a new video searching and fingerprint detection method by using an image query and PlaceNet-based SBD method. We used a places-centric dataset for PlaceNet training and combined it with an object-centric network for shot boundary detection. We presented a three-stage SBD method. We used several visual content features for candidate segment selection and used SIFT for shot boundary verification. For video searching and fingerprint detection, we tested several features and studied the thresholds. We compared our proposed method to several datasets, and the results showed the effectiveness of the SBD method. However, the feature storage required is larger than the other studied methods and the computing complexity still needs to be improved. For future research, we will evaluate our proposed method against larger datasets and simplify the processing complexity.

Author Contributions: D.Y.J. and J.K. designed the experiments; J.K. supervised the work; D.Y.J. analyzed the data; D.Y.J. wrote the paper; J.K. reviewed the paper.

Acknowledgments: This work was supported by the Technological Innovation R&D Program (S2380813) funded by the Small and Medium Business Administration (SMBA, Korea).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Awad, G.; Over, P.; Kraaij, W. Content-based video copy detection benchmarking at TRECVID. *ACM Trans. Inf. Syst.* **2014**, *32*, 14. [CrossRef]
2. Ng, P.C.; Henikoff, S. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Res.* **2003**, *31*, 3812–3814. [CrossRef] [PubMed]
3. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In *Computer Vision—ECCV 2006, Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.

4. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
5. Huang, J.; Kumar, S.R.; Mitra, M.; Zhu, W.J.; Zabih, R. Image indexing using color correlograms. In Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 7–19 June 1997; pp. 762–768.
6. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
7. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
8. Canny, J. A computational approach to edge detection. In *Readings in Computer Vision*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1987; pp. 184–203.
9. Hays, J.; Efros, A.A. August. Scene completion using millions of photographs. *ACM Trans. Graph.* **2007**, *26*, 4. [CrossRef]
10. Shensa, M.J. The discrete wavelet transform: Wedding the trous and Mallat algorithms. *IEEE Trans. Signal Process.* **1992**, *40*, 2464–2482. [CrossRef]
11. Ahmed, N.; Natarajan, T.; Rao, K.R. Discrete cosine transform. *IEEE Trans. Comput.* **1974**, *100*, 90–93. [CrossRef]
12. Tekalp, A.M. *Digital Video Processing*; Prentice Hall Press: Upper Saddle River, NJ, USA, 2015.
13. TREC Video Retrieval Evaluation: TRECVID. Available online: https://trecvid.nist.gov/ (accessed on 15 September 2018).
14. Matthijs, D.; Adrien, G.; Herve, J.; Marcin, M.; Cordelia, S. INRIA-IMEDIA TRECVID 2008: Video Copy Detection. 2008. Available online: http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/inria-lear.pdf (accessed on 15 June 2018).
15. Liu, Z.; Liu, T.; Shahraray, B. ATT Research at TRECVID 2009 Content-Based Copy Detection. 2009. Available online: http://www-nlpir.nist.gov/projects/tvpubs/tv9.papers/att.pdf (accessed on 15 September 2018).
16. Maguelonne, H.; Vishwa, G.; Langis, G.; Gilles, B.; Samuel, F.; Patrick, C. CRIMs Content-Based Copy Detection System for TRECVID. Available online: http://www-nlpir.nist.gov/projects/tvpubs/tv9.papers/crim.pdf (accessed on 22 September 2018).
17. Li, Y.N.; Mou, L.T.; Jiang, M.L.; Su, C.; Fang, X.Y.; Qian, M.R.; Tian, Y.; Wang, Y.; Huang, T.; Gao, W. PKU-INM @ TRECVid 2010: Copy Detection with Visual-Audio Feature Fusion and Sequential Pyramid Matching. 2010. Available online: http://www-nlpir.nist.gov/projects/tvpubs/tv10.papers/pku-idm-ccd.pdf (accessed on 15 September 2018).
18. Gupta, V.; Varcheie, P.D.Z.; Gagnon, L.; Boulianne, G. CRIM AT TRECVID 2011: CONTENT-BASED COPY DETECTION USING NEAREST NEIGHBOR MAPPING. 2011. Available online: http://www-nlpir.nist.gov/projects/tvpubs/tv11.papers/crim.ccd.pdf (accessed on 22 September 2018).
19. Wu, C.; Zhu, J.; Zhang, J. A content-based video copy detection method with randomly projected binary features. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 21–26.
20. Zhao, W.L.; Ngo, C.W. Flip-invariant SIFT for copy and object detection. *IEEE Trans. Image Process.* **2013**, *22*, 980–991. [CrossRef] [PubMed]
21. Kim, S.; Choi, J.Y.; Han, S.; Ro, Y.M. Adaptive weighted fusion with new spatial and temporal fingerprints for improved video copy detection. *Signal Process. Image Commun.* **2014**, *297*, 788–806. [CrossRef]
22. Lu, Z.M.; Li, B.; Ji, Q.G.; Tan, Z.F.; Zhang, Y. Robust video identification approach based on local non-negative matrix factorization. *AEU Int. J. Electron. Commun.* **2015**, *69*, 82–89. [CrossRef]
23. Mao, J.; Xiao, G.; Sheng, W.; Hu, Y.; Qu, Z. A method for video authenticity based on the fingerprint of scene frame. *Neurocomputing* **2016**, *173*, 2022–2032. [CrossRef]
24. Guzman-Zavaleta, Z.J.; Feregrino-Uribe, C.; Morales-Sandoval, M.; Menendez-Ortiz, A. A robust and low-cost video fingerprint extraction method for copy detection. *Multimed. Tools Appl.* **2017**, *76*, 24143–24163. [CrossRef]

25. Araujo, A.; Girod, B. Large-scale video retrieval using image queries. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 1406–1420. [CrossRef]
26. Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; Kompatsiaris, Y. Near-duplicate video retrieval by aggregating intermediate CNN layers. In *MMM 2017: MultiMedia Modeling, Proceedings of the International Conference on Multimedia Modeling, Reykjavík, Iceland, 4–6 January2017*; Springer: Cham, Switzerland, 2017; pp. 251–263.
27. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
28. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv*, **2014**; arXiv:1409.1556.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th Annual Conference on Neural Information Processing Systems 25 (NIPS2012), Lake Tahoe, Nevada, 3–6 December 2012; pp. 1097–1105.
30. Kikukawa, T.; Kawafuchi, S. Development of an automatic summary editing system for the audio-visual resources. *Trans. Inst. Electron. Inf. Commun. Eng.* **1992**, *75*, 204–212.
31. Zhang, H.; Kankanhalli, A.; Smoliar, S.W. Automatic partitioning of full-motion video. *Multimed. Syst.* **1993**, *1*, 10–28. [CrossRef]
32. Shahraray, B. Scene change detection and content-based sampling of video sequences. In *IST/SPIE's Symposium on Electronic Imaging: Science Technology*; International Society for Optics and Photonics: San Jose, CA, USA, 1995; pp. 2–13.
33. Küçüktunç, O.; Güdükbay, U.; Ulusoy, Ö. Fuzzy color histogram-based video segmentation. *Comput. Vis. Image Underst.* **2010**, *114*, 125–134. [CrossRef]
34. Janwe, N.J.; Bhoyar, K.K. Video shot boundary detection based on JND color histogram. In Proceedings of the 2013 IEEE Second International Conference on Image Information Processing (ICIIP), Shimla, India, 9–11 December 2013; pp. 476–480.
35. Li, Z.; Liu, X.; Zhang, S. Shot Boundary Detection based on Multilevel Difference of Color Histograms. In Proceedings of the 2016 First International Conference on Multimedia and Image Processing (ICMIP), Bandar Seri Begawan, Brunei, 1–3 June 2016; pp. 15–22.
36. Zheng, J.; Zou, F.; Shi, M. An efficient algorithm for video shot boundary detection. In Proceedings of the 2004 IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, China, 20–22 October 2004; pp. 266–269.
37. Adjeroh, D.; Lee, M.C.; Banda, N.; Kandaswamy, U. Adaptive edge-oriented shot boundary detection. *EURASIP J. Image Video Process.* **2009**, *2009*, 859371. [CrossRef]
38. Cooper, M.; Foote, J.; Adcock, J.; Casi, S. Shot boundary detection via similarity analysis. In Proceedings of the National Institute of Standards and Technology (NIST) TREC Video Retrieval Evaluation (TRECVID) Workshop, Palo Alto, CA, USA, 31 October 2003; pp. 79–84.
39. Priya, G.L.; Domnic, S. Edge Strength Extraction using Orthogonal Vectors for Shot Boundary Detection. *Procedia Technol.* **2012**, *6*, 247–254. [CrossRef]
40. Porter, S.; Mirmehdi, M.; Thomas, B. Temporal video segmentation and classification of edit effects. *Image Vis. Comput.* **2003**, *21*, 1097–1106. [CrossRef]
41. Bouthemy, P.; Gelgon, M.; Ganansia, F. A unified approach to shot change detection and camera motion characterization. *IEEE Trans. Circuits Syst. Video Technol.* **1999**, *9*, 1030–1044. [CrossRef]
42. Miadowicz, J.Z. Story Tracking in Video News Broadcasts. Ph.D. Thesis, University of Kansas, Lawrence, KS, USA, 2004.
43. Bendraou, Y. *Video Shot Boundary Detection and Key-Frame Extraction Using Mathematical Models*; Image Processing; Université du Littoral Côte d'Opale: Dunkirk, France, 2017.
44. Ngo, C.W.; Pong, T.C.; Chin, R.T. Video partitioning by temporal slice coherency. *IEEE Trans. Circuits Syst. Video Technol.* **2001**, *11*, 941–953.
45. Dadashi, R.; Kanan, H.R. AVCD-FRA: A novel solution to automatic video cut detection using fuzzy-rulebased approach. *Comput. Vis. Image Underst.* **2013**, *117*, 807–817. [CrossRef]

46. Bhaumik, H.; Chakraborty, M.; Bhattacharyya, S.; Chakraborty, S. Detection of Gradual Transition in Videos: Approaches and Applications. In *Intelligent Analysis of Multimedia Information*; IGI Global: Hershey, PA, USA, 2017; pp. 282–318.
47. Xu, J.; Song, L.; Xie, R. Shot boundary detection using convolutional neural networks. In Proceedings of the 2016 Visual Communications and Image Processing (VCIP), Chengdu, China, 27–30 November 2016; pp. 1–4.
48. Baraldi, L.; Grana, C.; Cucchiara, R. A deep siamese network for scene detection in broadcast videos. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 1199–1202.
49. Hassanien, A.; Elgharib, M.; Selim, A.; Hefeeda, M.; Matusik, W. Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. *arXiv*. 2017. Available online: http://research.ibm.com/haifa/projects/imt/video/Video_DataSetTable (accessed on 22 September 2018).
50. Liang, R.; Zhu, Q.; Wei, H.; Liao, S. A Video Shot Boundary Detection Approach Based on CNN Feature. In Proceedings of the 2017 IEEE International Symposium on Multimedia (ISM), Taichung, Taiwan, 11–13 December 2017; pp. 489–494.
51. Selesnick, I.W.; Baraniuk, R.G.; Kingsbury, N.C. The dual-tree complex wavelet transforms. *IEEE Signal Process. Mag.* **2005**, *22*, 123–151. [CrossRef]
52. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1452–1464. [CrossRef] [PubMed]
53. Places. Available online: http://places2.csail.mit.edu/demo.html (accessed on 22 September 2018).
54. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision—ECCV 2014, Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; pp. 818–833.
55. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
56. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI), San Francisco, CA, USA, 4–9 February 2017; Volume 4, p. 12.
57. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Berg, A.C. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
58. Vedaldi, A.; Fulkerson, B. VLFeat: An open and portable library of computer vision algorithms. In Proceedings of the 18th ACM International Conference on Multimedia, Firenze, Italy, 25–29 October 2010; pp. 1469–1472.
59. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary robust invariant scalable keypoints. In Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
60. Alahi, A.; Ortiz, R.; Vandergheynst, P. Freak: Fast retina keypoint. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 510–517.
61. Tola, E.; Lepetit, V.; Fua, P. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 815–830. [CrossRef] [PubMed]
62. Matas, J.; Chum, O.; Urban, M.; Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* **2004**, *22*, 761–767. [CrossRef]
63. Levi, G.; Hassner, T. LATCH: Learned arrangements of three patch codes. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–10 March 2016; pp. 1–9.
64. Alcantarilla, P.F.; Bartoli, A.; Davison, A.J. KAZE features. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 214–227.
65. Morel, J.M.; Yu, G. ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Imaging Sci.* **2009**, *2*, 438–469. [CrossRef]
66. Lin, W.Y.; Cheng, M.-M.; Lu, J.; Yang, H.; Do, M.N.; Torr, P. Bilateral functions for global motion modeling. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 341–356.

67. Bian, J.; Lin, W.Y.; Matsushita, Y.; Yeung, S.K.; Nguyen, T.D.; Cheng, M.M. GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2828–2837.
68. Filmora. Available online: https://filmora.wondershare.com/ (accessed on 22 September 2018).
69. Ground Truth Download. Available online: https://www-nlpir.nist.gov/projects/trecvid/trecvid.data.html#tv01 (accessed on 22 July 2018).
70. Smeaton, A.F.; Over, P.; Taban, R. The TREC-2001 Video Track Report. In Proceedings of the Tenth Text REtrieval Conference (TREC), Gaithersburg, MD, USA, 13–16 November 2001.
71. Li, S.; Lee, M.C. Effective detection of various wipe transitions. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 663–673. [CrossRef]
72. Cooper, M.; Liu, T.; Rieffel, E. Video segmentation via temporal pattern classification. *IEEE Trans. Multimed.* **2007**, *9*, 610–618. [CrossRef]
73. Li, Y.; Lu, Z.; Niu, X. Fast video shot boundary detection framework employing pre-processing techniques. *IET Image Process.* **2009**, *3*, 121–134. [CrossRef]
74. Lu, Z.; Shi, Y. Fast video shot boundary detection based on SVD and pattern matching. *IEEE Trans. Image Process.* **2013**, *22*, 5136–5145. [CrossRef] [PubMed]
75. Tong, W.; Song, L.; Yang, X.; Qu, H.; Xie, R. CNN-based shot boundary detection and video annotation. In Proceedings of the 2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Ghent, Belgium, 17–19 June 2015; pp. 1–5.
76. Paudyal, P.; Battisti, F.; Carli, M. A study on the effects of quality of service parameters on perceived video quality. In Proceedings of the 5th European Workshop on Visual Information Processing, EUVIP 2014, Paris, France, 10–12 December 2014; Available online: http://vqa.como.polimi.it/sequences.htm (accessed on 22 September 2018).
77. Roopalakshmi, R.; Reddy, G.R.M. A framework for estimating geometric distortions in video copies based on visual-audio fingerprints. *Signal Image Video Process.* **2015**, *9*, 201–210. [CrossRef]
78. Lei, Y.; Luo, W.; Wang, Y.; Huang, J. Video sequence matching based on the invariance of color correlation. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1332–1343. [CrossRef]
79. Dataset: CNN2h—Video Search Using Image Queries. Available online: http://purl.stanford.edu/pj408hq3574 (accessed on 22 September 2018).