IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 March 2023, accepted 23 March 2023, date of publication 29 March 2023, date of current version 5 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3262796

RESEARCH ARTICLE

Wipe Scene Change Detection in Object-Camera Motion Based on Linear Regression and an Inflated Spatial-Motion Neural Network

DIPANITA CHAKRABORTY[1], WERAPON CHIRACHARIT[1], (Member, IEEE), KOSIN CHAMNONGTHAI[1], (Senior Member, IEEE), AND THEEKAPUN CHAROENPONG[2]

[1]Department of Electronic and Telecommunication Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand
[2]Department of Biomedical Engineering, Srinakharinwirot University, Nakhon Nayok 26120, Thailand

Corresponding author: Kosin Chamnongthai (kosin.cha@kmutt.ac.th)

This work was supported by Petchra Pra Jom Klao Ph.D. Research Scholarship from the King Mongkut's University of Technology Thonburi under Grant 62/2563.

ABSTRACT To facilitate content-based video analysis, automatic scene change detection (SCD) with large-scale motion activity is an essential fundamental step for locating a transition from one video scene to another. With the exponential increase in digital media usage, SCD has become more challenging in processing large motion content with minimal information loss and maximum perseverance. Wipe SCD in object-camera motion is noticeable evidence of this issue. Wipe transitions, which are a type of gradual transition, have diverse motion pattern changes when influenced by object-camera motion (camera pan, large-object, and zoom-in/out), creating a velocity imbalance in the same frame. Furthermore, this motion imbalance leads to false detection. Due to the loss of motion information and longer processing time of existing frameworks, we propose a novel method of wipe scene change detection (WSCD) based on deep spatial-motion feature analysis. First, large input videos are segmented into shots using dimensionality reduction and adaptive threshold. Secondly, linear regression is used to compute slope angle changes in shots for candidate selection and wipe localization. Finally, only selected candidates are processed to extract features using a two-stream inflated 3D-convolutional neural network for RGB stream and optical flow velocity for motion stream network (I3DCNN) and then classified into wipe in-motion and no-motion clips. The experimental results are obtained by classifying wipe patterns using a detection reviewing and merging strategy on corresponding wipe frames. The average improvement in wipe scene change detection accuracy evaluated on the benchmark TRECVID dataset is 11.9%, demonstrating the efficacy of our proposed method.

INDEX TERMS Cut transition, gradual transition, I3DCNN, linear regression, object-camera motion, scene change detection (SCD), shot boundary detection (SBD), wipe scene change detection (WSCD).

I. INTRODUCTION

After the COVID-19 pandemic, the number of digital media platform users has continued to grow at an extraordinary level, which has created a challenge for data management systems to handle this massive amount of data [1]. In addition, the use of object-camera motion activities in multimedia video production has increased, significantly [2]. Object-camera motion can be mainly classified into three types: large object, camera pan, and zoom-in/out motion. Such diverse object-camera motion has made video indexing, content-based video searching, and video retrieval tasks more challenging. These tasks enable high-level semantic information analysis to efficiently run video-based applications such as video surveillance, video production, and creation [3], [4]. To this end, video shot boundary detection (SBD) and scene change detection (SCD) are the essential preliminary steps that provide the semantics of the video content [5], [6]. A video sequence can be arranged in descending order of scenes, shots, and frames. Several continuous frames form

The associate editor coordinating the review of this manuscript and approving it for publication was Zahid Akhtar.

shots. A change from one shot to the next shot is defined by a shot transition, and this transition can be consisting of one or several frames known as shot boundaries [7]. A scene is made of several shots, and when a specific shot transition contains a change from one scene to another scene, it is called a scene change. Hence, scene changes and shot changes are interconnected [6], [8]. Video shots or scene changes are created by video editing effects which can be divided into two main types: abrupt transition and gradual transition [5], [7]. In abrupt or cut transitions, shot changes occur suddenly, also known as cut shot boundaries. In gradual transitions, shot changes occur slowly with several frames changing in a pattern. Gradual shot transitions can be of two types, a single pattern type (i.e., dissolve, fade-in/out) and a multiple pattern type, that is, wipe [9], [10]. According to statistical survey records, most of the existing gradual transition detection (SBD) methods have mainly focused on dissolve and fade-in/out transitions [11]. Research studies conducted on wipe transition detection have not focused on sufficient object-camera motion-based wipe transitions. The importance of wipe transition detection in object-camera motion can be described as follows:

- Wipe transitions carry important low-level semantic information of the context and content of a video to obtain high-level information; hence, their automatic detection is a pivotal step in assisting content-based video analysis [13], [16].
- Wipe transitions are widely used in digital media (news, sports, and movies) to provide audiences with a better content-watching experience [12]. The popular video editing software Adobe Premiere Pro CC has more than 30 commonly used wipe patterns [15].
- Not only do wipe transitions create disturbances in other gradual scene change detection methods, but they also generate velocity imbalance and similar shape pattern confusion problems when combined with object-camera motion, as explained in Section III in detail.

Due to their complexity and diverse changing patterns, wipe transitions are difficult to detect compared to other transition types as shown in Fig. 1.

In wipe transitions, the pixels in the current scene are gradually replaced by the pixels in the next scene with or without object-camera motion activities. This transition occurs with a moving border in a shape pattern between the current and next scene. These moving borders consist of arbitrary shapes, directions, and speeds [16]. Most importantly, when arbitrary object-camera motion occurs in the same frame as moving borders, it results in a velocity imbalance between them. Existing wipe transition detection methods suffer from this problem [15]. Wipe transition classification methods can be divided into three groups: 1) rule-based methods, 2) machine learning methods, and 3) rule-machine learning methods [10]. These methods have reduced the difficulties of wipe transition detection to some extent which can be described by dividing wipe transitions into two groups depending on their movement as follows:

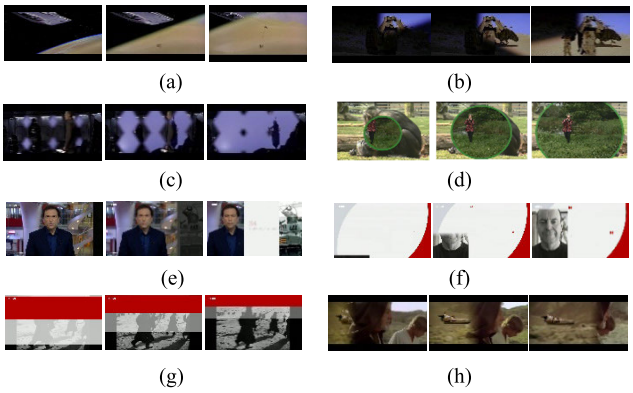

FIGURE 1. Types of wipe scene changing patterns; (a) Diagonal (no-motion); (b) Clockwise (no-motion); (c) Iris shapes (no-motion); (d) Circle (no-motion); (e) Half horizontal (no-motion); (f) Half vertical (no-motion); (g) Vertical (in-motion); (h) Horizontal (in-motion).

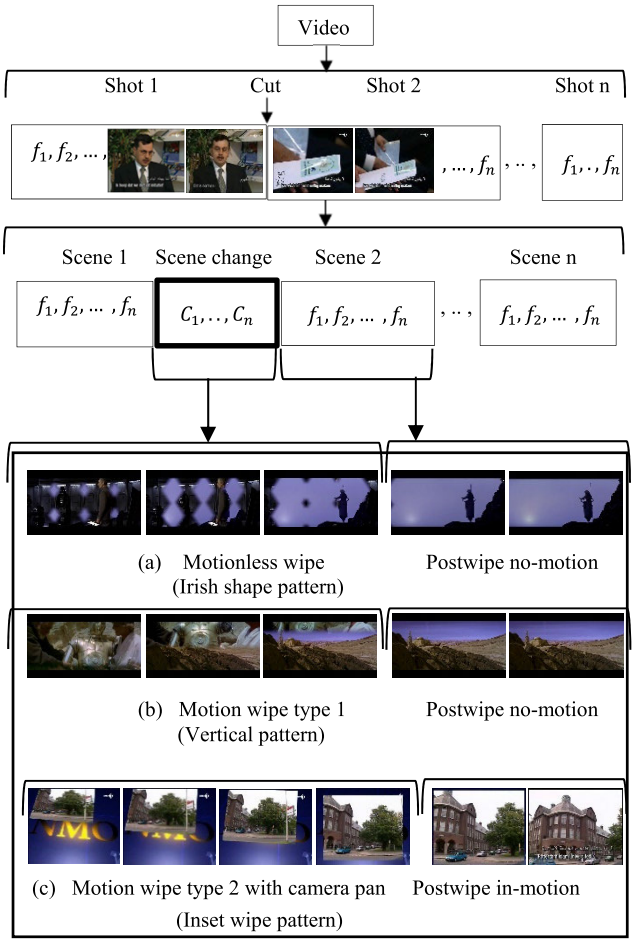

FIGURE 2. Architecture of a video with frames, shots, and scenes, where shot changes occur with cut shot boundaries, and scene changes occur with motionless and motion wipe transitions.

- Motionless wipe transitions (WIP): The key feature in this group is that during a transition, the current shot and next shot remain globally motionless, i.e., no foreground or background movement occurs between wipe-in and wipe-out scenes as shown in Fig. 2. However, borders might move at any arbitrary speed and in

any direction, such as horizontal, vertical, or diagonal directions, or according to geometrical shapes such as rectangular, circular, diamond or any random object's shape. Rule-based methods related to visual features can be performed by edge-based [15], [18], histogram-based [12], [17], [58], pixel-based [26], [27], and visual rhythm based [20] algorithms.

- Motion wipe transitions (WIM): In contrast, motion wipe transitions can be divided into two types.

Type 1 motion wipe is where the foreground and background movements have global motion only, i.e., either the current wipe-in scene 1 is moving onto the next wipe-out scene 2, or the next wipe-out scene 2 is moving on to the current wipe-in scene 1. Hence, foreground and background movements have the same velocity as the moving wipe. Transformation-based [21], motion vector-based [22], [25], and machine learning [23], [38], [56], [57] algorithms can be employed for motion type 1 wipe transition detection.

Type 2 motion wipe is where object-camera motion occurs in foregrounds or backgrounds, while type 1 motion wipe is occurring. Hence, wipe scenes have different velocities, that is, the local velocity during object-camera movements in the foreground and background and the global velocity during type 1 motion wipe. Object-camera motion may occur due to large objects, camera panning, and zoom-in/out movements, as shown in Fig. 2. Block motion vector-based [16] and motion -machine learning-based [24] algorithms are generally used for type 2 motion wipe transitions.

However, these methods do not classify motionless and motion wipe transitions in various object-camera motions. In addition, motion-machine learning methods that have detected other gradual transitions (dissolve, fade-in/out) in object-camera motion require high computational costs [14]. Therefore, the correct detection of motion wipe transitions with fast computation is urgently needed.

The main contributions of this paper are as follows:

- Video segmentation: A fast computational method is developed to segment the video using dimensionality reduction by principal component analysis (PCA) and adaptive threshold techniques. To do so, cut shot boundary detection is selected as the segmentation criterion. Since the cut transition feature is similar to the beginning and ending frame features of the wipe transition, its presence in the input video causes disturbances in wipe scene change detection. Therefore, we chose Cut SBD for the preprocessing step.
- Candidate selection to locate wipes: A suitable candidate key frame selection algorithm is developed to locate motion and motionless wipe transitions that can eliminate nonwipe frames and other frames consisting of small object-camera motion activity. Thus, we employ linear regression analysis for edge feature changes in each segmented video clip to predict the temporal location of nonwipe and wipe frames. This technique reduces the computational time by enabling processing with fewer frames in the classifier.
- Very deep spatial-motion feature extraction and classification: Elimination of falsely detected large object-camera motion frames from the selected candidates is performed by analyzing very deep global-local motion information corresponding to their spatial features using an inflated 3DCNN and optical flow velocity neural network (I3DCNN). Hence, we create a training dataset and set the final classification output criteria by employing a prediction threshold T_s. This technique prevents selected candidates from having a prediction probability (fully connected layer) that is less than the threshold, considers them as nonwipe object-camera motion clips, and provides the final classified output in motionless wipe and motion wipe (type 1, type 2) transitions with high accuracy.
- Detection reviewing and merging: The classified wipe transitions are reviewed by detecting their shapes and merging the detected wipe clips to form a complete wipe transition. We utilize the advantages of the transfer learning-based CNN-LSTM network (less training time, less data) for 12 different commonly used wipe shape classifications (horizontal, vertical, Irish, half wipe, etc.); then, correctly detected segmented candidate wipe transitions of each complete transition are merged with the proposed merging strategy. We choose this technique as the postprocessing step because it increases the detection confidence of the proposed method.

This paper is organized as follows: related works are illustrated in Section II. Section III describes the mathematical analysis of wipe scene change in object-camera motion, followed by Section IV, which explains the proposed WBSC method. The experimental results and discussions are provided in Sections V and VI, respectively. Finally, some conclusions are presented in Section VII.

II. RELATED WORKS

Most related papers in recent years have mainly focused on single transition pattern detection, such as cut and gradual (dissolve, fade-in/out) transitions using different approaches [53], [54], [55], [56], [57], [58], [59]. In addition, these methods face challenges in reducing the processing time while preserving accuracy. In contrast, wipe transitions are considered multitransition patterns due to various shapes with arbitrary motion. To overcome this issue, existing research has either included wipe transitions with single transition pattern detection methods or have exclusively proposed methods for wipe transition detection, as illustrated in subsections A, B, and C and further summarized in Table 1.

A. FEATURE EXTRACTION-BASED APPROACH

The objective of this approach is to extract low-level visual features from video frames. A detailed literature review is conducted to analyze feature differences among various techniques.

TABLE 1. Wipe Scene change detection algorithms based on different approaches.

Algorithm name	Brief methodology	Highlights	Limitations
	Feature Extraction-based Approach		
Pixel-based (PBA) [26]	Pixel values + statistical and structural property + pixelwise difference + threshold + merging + gap elimination	Robust to various wipe types and lengths	Sensitive to motion activity
Pixel-based (PBA) + Hough [27]	Pixel values + statistical and structural property + threshold + pixelwise difference + Hough transformation + majority voting candidates	Detailed wipe pattern classification and no external threshold requirement	Pattern similarity
Pixel-based (PBA) + DC [19]	DC image + pixelwise luminance difference + threshold + statistical and structural property	Wipe transitions in small motion activity	Suitable for few wipe patterns
Visual rhythm-based [20]	Lines/curves on VRS + intensity discontinuity signal + statistical and structural property + adaptive threshold	Real-time	Fast motion sensitivity
Edge pixel-based [15],[18]	Edge or border pixel values + continuity/discontinuity signal + adaptive threshold	Wipe transitions in average object motion activity	Camera motion sensitivity
Histogram-based [33]	Color histogram (CbCr) + histogram intersection + color wise difference	Spatial-temporal feature analysis	Motion sensitivity
Histogram-based + filter [17]	Color histogram (Luminance, CbCr, RGB, HSV) + color wise difference adaptation + filter function modeling + threshold	Unified model	Motion sensitivity
Histogram-based + linearity [12]	Color histogram + color wise difference + linearity property in histogram space	Wipe transitions in motion	Camera motion sensitivity
Histogram-based + SAD [40]	Gaussian blur + SAD + block motion (32×32) + color histogram (YCbCr) + CDF + histogram values and thresholds	Temporal features (cut and dissolve)	No wipe detection
Histogram-based + LBP-HF [53]	Wiener filter + LBP-HF features + Dissimilarity (block LBP-HF + Canny edge difference) + threshold	Robust to illumination and motion effect	Cut SBD detection only.
Histogram-based (MCSH) [54]	Sobel gradient + Fuzzification + Block cumulative sum + Mean + MCSH histogram + Dissimilarity (RSD difference)	MCSH dissimilarity performs consistently well	Camera motion sensitivity
Discontinuity Signal + B [28]	2-D wavelet decomposition + pixelwise dissimilarity + B-line interpolation curve fitting + threshold	Wipe in object motion	Few wipe patterns
Discontinuity Signal (block) [21]	Db-4 wavelet transition + multifeatured subimage (color, edge) + block dissimilarity signal (8×8) + Hough transform + threshold	Dissimilarity signal of motion and gradual transitions	Fast object-camera motion
Discontinuity Signal-GMM [36]	Energy minimization using GMM + multifeatured subimage (color, edge) + block dissimilarity signal (8×8) + threshold	Background and foreground separation	Motion sensitivity
Continuity Signal + Fuzzy [29]	Six different features (color histogram, edge, intensity variance, etc.) + HHD + continuity signal + Fuzzy logic	Avoid hard cut thresholds and large training data	Motion sensitivity
Multi-modal visual features [13]	SURF + RGB + Cosine Similarity and multifeature dissimilarity + threshold	Good performance on cut and gradual transitions	Camera motion sensitivity
	Motion Activity-based Approach		
MPEG coded domain [16],[30]	Macroblock type feature information + PBBI SGOP + threshold	Spatial and temporal feature information analysis	Brightness sensitivity
Motion Activity [31]	Motion activity descriptors (MADs) + Dominant color descriptors (DCDs) + interframe difference (TAR matrix) + threshold	Motion and motionless wipe transitions	Only Ho and Ve type wipe
A unified model [24]	Middle level features (dominant colors, block motion, RGB color histogram) + interframe difference + SVM classifier	Motion and motionless wipe transitions	Only graphical wipe
Han *et al.* [22]	3-DWT + GWMH distance + SW extraction + direction of MV	Motion and motionless wipe transitions	Fewer wipe pattern
DWT and motion activity [32], [34]	DWT + mean of wavelet coefficients + graphical pattern + frame number statistics	Various wipe transitions	Lengthy motion sensitivity
Optical flow based [25],[35],[37]	Visual feature + Optical flow motion vectors features + difference + transition type determination	Local temporal motion information, fast computation	Camera motion sensitivity
	Machine Learning-based Approach		
Shen *et al.* [49]	Histogram difference + HLFPN model based on statistical ML + SURF Keypoint matching	Removed false detections obtained by existing methods	Inconsistent recall value
Bezerra *et al.* [44]	H and V slices on frames + LCS vectors + maximum matching distance (mmd) + filtering + K-means clustering (mmd)	No thresholds, new concept of distance computation	Object motion sensitivity

TABLE 1. *(Continued.)* **Wipe Scene change detection algorithms based on different approaches.**

	Machine Learning-based Approach (Cont'd)		
CNN-based [38]	Candidate selection (intensity difference +threshold) + CNN features (fc-6 layer) + binary Classification	Semantic information	Insufficient temporal feature
Fully CNN [39]	Feature extraction with Fully convolutional layer (no FC layer) + binary classification	Ridiculously fast SBD with 121x real-time	In wipe, Ho type only
ShotCoL [41]	Self-supervised learning-MLP: encoder network for audio-visual modality + similarity signal by momentum contractive learning	Excellent learning features on limited labeled data	Investigated on cut SBD only.
DeepSBD [23]	16-frame segments + 3D-CNN features extraction + SVM classification + Merging + histogram-temporal differencing	Spatial and temporal deep features	Processing time
Idan *et al.* [10]	Preprocessing + STKP local-global feature extraction on OP + Candidate selection by temporal DS + binary SVM classification	High cut transition accuracy	Limited scene transition types
Abdulhussain *et al.* [42]	Frame active area + Candidate selection (moments + thresholds) + STKP local feature extraction + OP + DS+ SVM classification	High hard cut transition accuracy	Limited scene transition types
New VSBD [55]	Siamese recurrent architecture-based feature extraction (CNN-LSTM) + 20 frame segments + Similarity + Candidate Selection	High cut transition accuracy, invariant to flash and motion	Wipe is not investigated
VSBD-POCS model [56]	Training (POCS-feature extraction + Dissimilarity + Temporal continuity signal learning) + Testing (Bagged Tree classifier)	Multiboundary classification (CT, grouped GT, and NT)	Insufficient wipe analysis
Multifeature + SVM [57]	Multifeature extraction (ECR, CLD, SIFT) + Continuity Signal + SVM classification	High performance of cut and dissolve transition	Wipe is not investigated
TSSBD [14]	Segmentation by CNN+HSV + 16-frame sliding window + C3D feature extraction + multiclass classification + Merging	Good performance on cut and various gradual transition	Wipe is not investigated

1) VISUAL FEATURE EXTRACTION

Pixel-based algorithms (PBAs) or pixelwise comparisons are some of the simplest and most popularly used methods for visual feature extraction [26], [27]. Therefore, these features are used for statistical property-based computations (mean and standard deviation) applied for wipe transition detection. However, this technique suffers from a high false alarm rate due to small motion activity. Wu et al. [19] proposed an overall pixelwise comparison approach for direct current (DC) images and statistical properties, however their method can only detect a few wipe patterns. To speed up the efficiency of pixel-based algorithms, Seo et al. [20] proposed subsampling pixels from a specific position from each frame to represent the visual content, known as visual rhythm- based algorithm. However, pixel-based algorithms have been found to be sensitive to local and global motion in many research studies. To overcome this issue, edge-based [15], [18] and histogram-based [12], [17], [40], [53] algorithms have been proposed. Li and Lee [15] proposed independence and completeness properties based on edge-pixel feature changes to obtain the structure of wipe transitions and locate them. The main drawback of this paper is that this method is only implemented considering motionless wipe transitions and motion wipe transitions without camera panning motion, resulting in a high false alarm rate in frames with large object-camera motion transitions.

Histogram-based algorithms [12], [17], [33], [54] have been proposed to construct a unified cut and gradual transition (dissolve, fade-in/out, and wipe) detection model. These histograms (color and HSV) are used for obtaining thresholds to classify (rule-based classifier) transition types. Since color histograms [50] do not incorporate the spatial distribution information of different colors, they are less sensitive to small object-camera motion transitions. However, this method is not expressive enough to accurately classify different transition types and is sensitive to object-camera motion due to fast changes in color information.

2) CONTINUITY SIGNAL CONSTRUCTION

Global temporal feature analysis using visual features is an effective way to construct frame continuous (similarity) and discontinuous (dissimilarity) signals for localizing wipe transitions. In general, a discontinuity occurs between two scenes due to a cut or gradual transition. Otherwise, the shot maintains its continuity. However, when large object-camera motion appears in continuous nontransition frames (normal), such frames have temporal changes similar to gradual transitions, causing discontinuities.

A discontinuous or continuous signal can be obtained by calculating the adjacent interframe feature distances. Nam and Tewfik [28] calculated pixelwise dissimilarity and B-line interpolation curve fitting techniques to measure the linearity of wipe transitions. However, their method only detects a few wipe transitions that are linear in the specified direction. A multifeatured block dissimilarity computation technique was proposed in [21]. Thomas et al. [36] proposed an approach for minimizing the energy (color-edge) of a shot's background to analyze temporal dissimilarity. However, their method is sensitive to large foreground movements.

In contrast, Refaey et al. [29] analyzed similarity using the hue histogram distance (HHD) to achieve robustness against illumination changes. However, this method is sensitive to

large object-camera motion. Tippaya et al. [13] have computed cosine similarity signals by using SURF matching and RGB histograms. However, their method has not been used to investigate wipe transitions.

B. MOTION ACTIVITY-BASED APPROACH

The objective of the motion activity-based approach is to analyze local-global object-camera movements in normal and transitional frames.

Some studies have proposed macroblock type temporal feature information analysis in the MPEG coded domain to locate wipe scene changes [16], [30]. However, they suffer from a high false alarm rate due to illumination and brightness changes. Mackowiak and Relewicz [31] proposed motion activity and color dominant feature-based descriptors, but only evaluated horizontal and vertical wipe transition patterns. Han et al. [24] proposed a unified SBD model using dominant color features and motion vectors. However, their method also detects only a few graphical wipe transitions. Discrete transformation-based approaches (DCT and DWT), which represent motion changes by transforming image signals into the transform domain, have been proposed in [22], [32], and [34]. In addition, Yufeng et al. [21] used a Hough transformation on a combined color-edge feature-based wavelet transformation image signal to obtain the patterns of wipe transitions. However, these methods are sensitive to fast-motion activity. Chavan et al. [34] analyzed wipe transitions for only one kind of motion, i.e., large object motion only; hence, this method is sensitive to camera panning and zoom-in/out motion.

An effective optical flow-based motion vector estimation approach was proposed in [25], [35], and [37]. These motion vectors leverage the system to obtain local temporal information with fast computation, which is necessary for motion wipe detection. Hence, unlike the existing methods, we inflate the optical flow vectors with a CNN classifier to preserve maximum temporal feature information in correspondence to spatial features.

C. MACHINE LEARNING-BASED APPROACH

A statistical machine learning approach was proposed by Shen et al. [49], who used speeded-up robust features (SURF) and fuzzy logic, known as high-level fuzzy petri net (HLFPN) for key point matching. However, the recall value is inconsistent with limited features in their method. Bezerra [44] proposed a k-means clustering-based classification technique; however, this method has only been employed for horizontal and vertical wipe transition patterns. Therefore, a deep learning-based convolutional neural network (CNN) approach was proposed in [38], [39], and [55] to obtain high-level semantic features by combining low-level features. However, their method has not detected wipe transitions. Chen et al. [41] proposed a self-supervised machine learning approach using audio-visual features. However, their method only works for cut scene change detection. Hassanien et al. [23] proposed a spatial-temporal feature-based CNN that detects wipe transitions and other gradual transitions. However, the processing time is higher in this method. To resolve this issue, in [10] and [42], a candidate segmentation method was proposed to select candidates and then only the candidates were fed into a support vector machine (SVM) classifier. However, they did not precisely classify wipe transitions in object-camera motion. Following the same objective, we have proposed a new linear regression-based candidate selection approach. Wu et al. [14] proposed a CNN-HSV and 3DCNN approach (TSSBD) for analyzing deep spatial-temporal features to detect gradual transitions (dissolve, fade-in/out, and swipe) in the presence of large object motion and illumination changes. However, their method is sensitive to fast camera movements. Hence, to achieve higher wipe transition detection accuracy in large object-camera motion we have proposed an inflated 3DCNN and optical flow velocity (I3DCNN) for extracting very deep motion features corresponding to the spatial features.

In the algorithms mentioned in the three subsections above, scene change detection approaches start by processing entire video frames to extract visual and motion features using different techniques. Although these techniques have worked well for cut and dissolve transition detection, they still lack sufficient feature information for wipe transition detection. As discussed in Section I, wipe transitions have multiple shape pattern features including their independent motion pattern from other moving objects or camera movements. Hence, a good semantic correlation balance between spatial and motion features information is required for wipe transition detection, and it is analyzed in Section III of this paper. Once feature extraction is performed, all these extracted features are fed through either rule-based or machine leering based algorithms for the scene change detection. Apparently, processing entire features decreases the robustness of the system. Addressing this issue, Benoughidene and F. Titouna [55] proposed a candidate frame selection-based algorithm for potential scene transition localization; however, wipe scene transitions were not investigated with their proposed method. Motion sensitivity and lack of focus in wipe transition detection are the two most common limitations that can be observed in Table 1 of related works on scene transition detection.

Therefore, to summarize, the two major problems that we aim to solve are 1) the high computational time by proposing a fast computational video segmentation and candidate selection method and 2) the high false alarm rate in wipe transition detection due to object-camera motion by proposing a very deep spatial-motion (global-local temporals) feature analysis technique using an inflated 3DCNN and the optical flow velocity network.

III. ANALYSIS OF WIPE SCENE CHANGES IN OBJECT-CAMERA MOTION

Diverse geometrical shape patterns make wipe transitions a distinct type of transition among all the others. Whereas

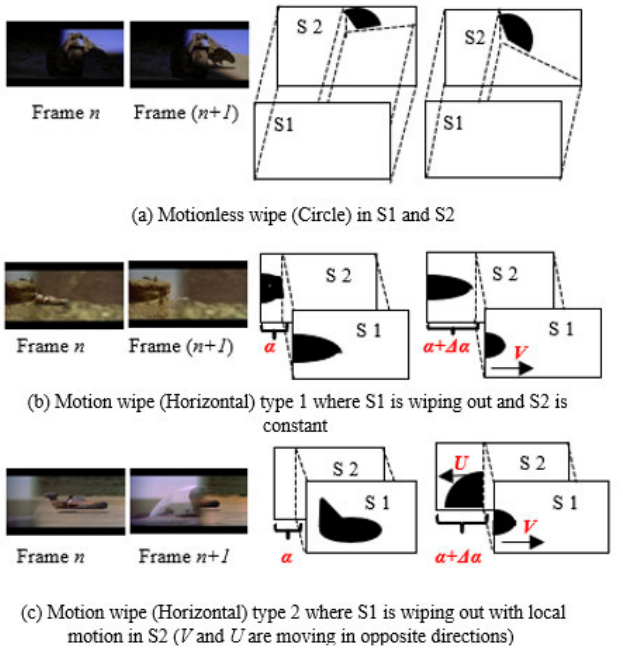

FIGURE 3. Structural representation of wipe transitions in no-motion and in-motion situations, where S1 and S2 represent Scene 1 and Scene 2, respectively; (α, $\alpha + \Delta\alpha$) represents the change in the distances of the wipe transition from frame n to next frame $n + 1$; V and U represent wipe transition motion in S1 and large object motion in S2, respectively.

camera movement (pan or zoom-in/out) can be called global motion, large object movements can be called local motion.

Depending on the correlation between global-local motion and wipe transition motion, a wipe transitions can be represented as three types, as shown in Fig. 3. In a motionless wipe (WIP) transition, no global motion occurs. Similarly, in the type 1 motion wipe transition, small local motion might occur, but no global motion occur; hence, it has the behavior of a motionless wipe transition. However, in the type 2 motion wipe transition, either both global and local motion occur or any of the motions occur. The methods proposed in [15] and [32] can be used to efficiently detect motionless and motion wipe transitions without large global-local motion, as shown in Fig. 3 (a) and (b). Furthermore, a DWT-based method was proposed to detect wipe transitions with local motion [34], as shown in Fig. 3 (c). However, this method is unsuitable for wipe transitions with global motion (camera pan or zoom-in/out), as shown in Fig. 4. The object-camera motion creates two major problems for wipe transition detection: 1) velocity imbalance and 2) pattern similarity, which is mainly responsible for the low accuracy rate in existing methods, as shown in Fig. 4 and Fig. 5.

According to case study 1 regarding the velocity imbalance in Fig. 4, existing wipe transition detection methods fail to detect the beginning and ending frames of wipe transitions in lengthy transitions. This occurs because the beginning and ending frames of wipe transitions possess less wipe pattern changing motion.

According to case study 2, due to the shape pattern similarity between a wipe transition and object motion, existing

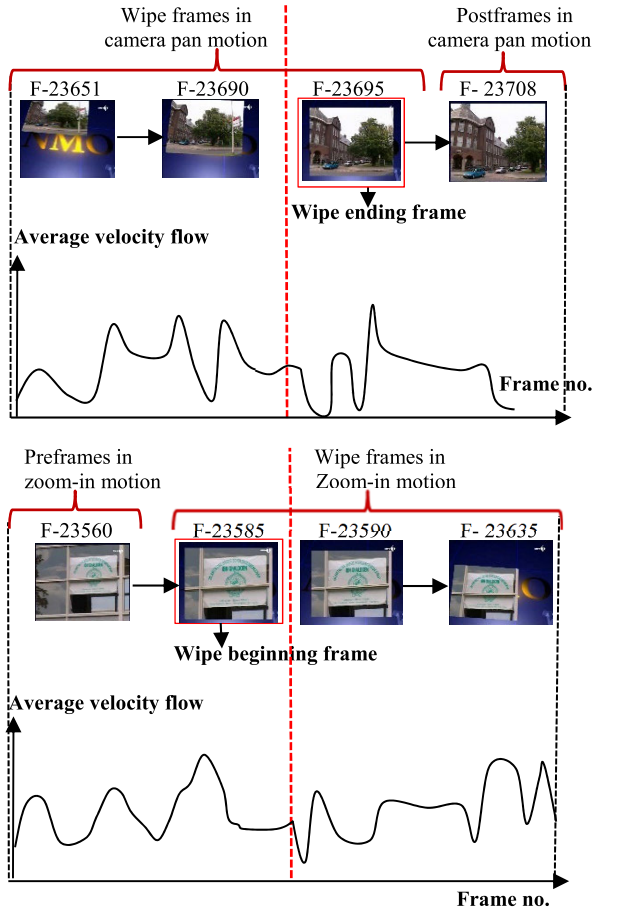

FIGURE 4. Case study 1 (Velocity imbalance); Missed detections of beginning and ending frames of wipe transitions due to a lack of wipe changing velocity flow. Frame numbers (no.) belong to the TRECVID 2007 dataset.

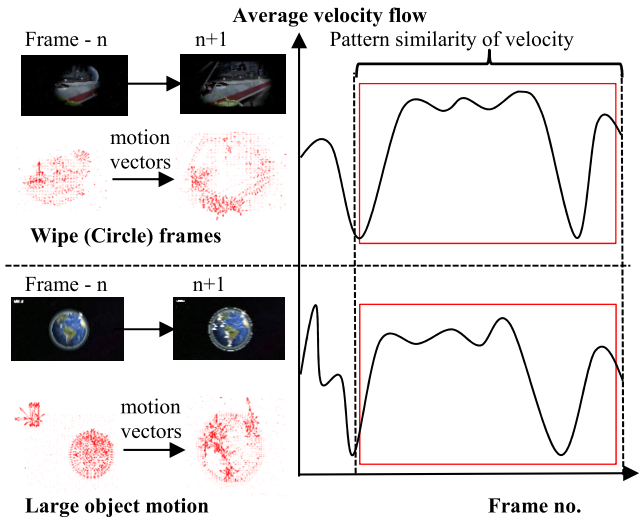

FIGURE 5. Case study 2 (Pattern similarity). False detection occurs due to the similar flow of motion vectors between consecutive frames.

methods suffer from the false detection of wipe transitions, as shown in Fig. 5.

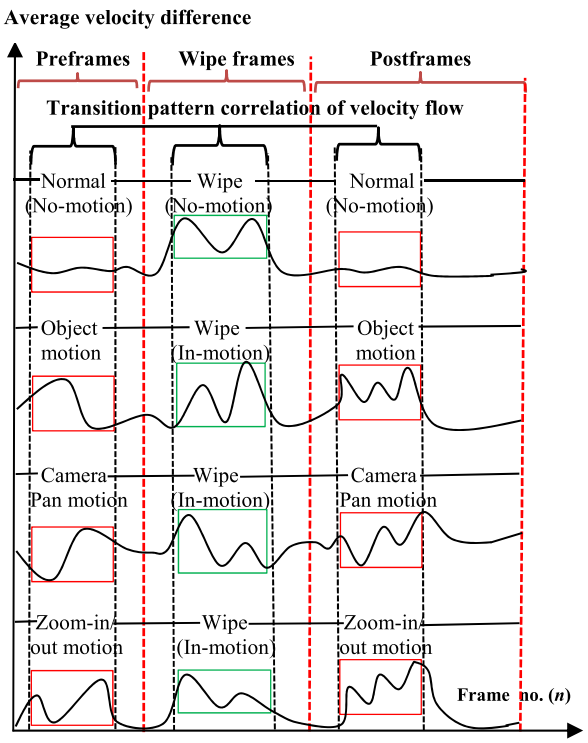

FIGURE 6. Representation of average velocity feature difference of wipe transitions; previous, and postwipe transitions in normal or motionless, large object, camera pan, and zoom-in/out motion frames.

According to the average velocity feature difference in wipe transitions, motion wipe with object-camera motion can be distinguished from motionless wipe, as shown in Fig. 6

Most existing works have employed average spatial and temporal information to classify wipe transitions from other gradual scene changes. RGB features are commonly used as spatial information. The temporal or motion feature information equation in existing methods can be derived according to information theory, as in (1).

$$\begin{aligned} \text{Classify}\,(I\,(W_t)) &= (I\,(d_{n-1}) + I\,(s_{n-1})), \\ \text{or, Classify}\,(I\,(W_t)) &= \log(1/p\,(d_{n-1})) \\ &\quad + \log(1/p\,(s_{n-1})). \end{aligned} \tag{1}$$

where, I represents the information and W_t represents the wipe scene changes in no-motion or in-motion. d_{n-1} and s_{n-1} represent motion vector flow (velocity) information in the preframes or in scene 1 and in the postframes or in scene 2 for motionless wipe transitions, respectively.

However, two more crucial features need to be added, namely, the interframe distance and the peak height correlation of motion vector information between the wipe and object-camera motion. These two pieces of information have been incorporated in this paper for wipe scene change

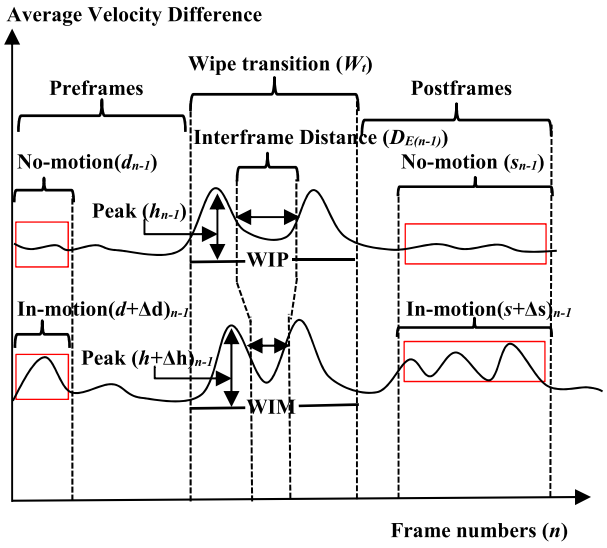

FIGURE 7. Representation of a wipe transition (Horizontal) depending on the presence of motion in pre- and postframes.

detection in object-camera motion and are described in (2).

$$\begin{aligned} \text{Classify}\,(I\,(W_t)) &= (I\,(s + \Delta s)_{n-1} + I\,(d + \Delta d)_{n-1} \\ &\quad + I\,(D_{E(n-1)})) + (h + \Delta \text{h})_{n-1}), \\ \text{or, Classify}\,(I\,(W_t)) &= \log(1/p\,(d + \Delta \text{d})_{n-1}) \\ &\quad + \log(1/p\,(D_{E(n-1)}) \\ &\quad + \log(1/p\,(h + \Delta \text{h})_{n-1}) \\ &\quad + \log(1/p\,(s + \Delta \text{s})_{n-1}). \end{aligned} \tag{2}$$

where, $(d + \Delta d)_{n-1}$,$(s+\Delta \text{s})_{n-1}$, $D_{E(n-1)}$, and $(h + \text{h}\Delta)_{(n-1)}$ represent the motion vector flow (velocity) information in the preframes and postframes, the interframe distance, and the peak height of wipe transitions in the object-camera motion category, respectively. Therefore, we propose a slope-angle function using linear regression and a two-stream inflated 3DCNN-based method for wipe scene change detection that not only efficiently detects wipe scene changes but also reduces the computational time, as explained in Section III. The experimental results show that our proposed method increases the accuracy of wipe in-motion scene change detection by 11.9%, as illustrated in Section IV.

IV. PROPOSED WIPE SCENE CHANGE DETECTION IN OBJECT-CAMERA MOTION

In this section, the proposed wipe scene change detection (WSCD) method is described in detail. As depicted in Fig. 8, the proposed method is implemented in three major stages, which are explained in subsections A, B, and C. These stages include video segmentation by cut SBD using principal component analysis (PCA) and interframe distance calculations, candidate selection by generating 20 frame sliding windows from the segmented video clips, wipe transition localization using Canny edge feature extraction and slope angle change

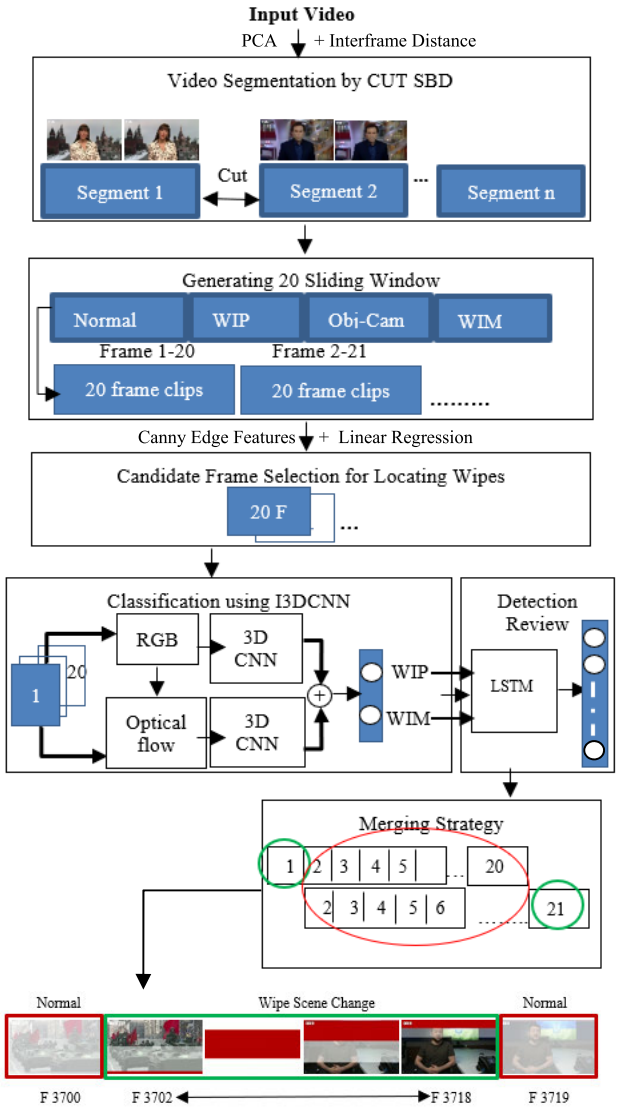

FIGURE 8. Architecture of the proposed WSCD method.

calculations with linear regression, and selected candidate classification in motionless wipe and motion wipe transitions by applying an inflated 3DCNN model that uses two streams, i.e., a 3DCNN for deep spatial feature extraction and an optical flow network for motion feature extraction. Finally, the proposed WSCD method is reviewed using a postprocessing step by classifying detected wipe transitions into 12 commonly used wipe shape patterns using CNN-LSTM network and the corresponding wipe clips are merged into a complete wipe transition.

A. VIDEO SEGMENTATION USING CUT SHOT BOUNDARY DETECTION

Video segmentation helps to analyze the dissimilarities between scene changes and object-camera motion by segmenting scenes into small batches for future steps. A significant distinct feature of the wipe and cut transition is the disparity between the number of frames in the transition phase. Considering this, cut shot boundaries are detected first and then, wipe transitions are detected.

1) PRINCIPAL COMPONENT ANALYSIS

The statistical monitoring of principal component analysis (PCA) in [45] indicates that it is a well-known method for dimensionality reduction of a dataset with extremely fast computation while minimizing the loss of information. Thus, we used PCA on our large input video dataset to select the most appropriate pixel change features in each frame by performing eigenvector and eigenvalue calculations. We obtain the feature vector (F_n) for n frames by calculating the highest eigenvectors from the first and second principal components, as in (3).

$$F_n = (v1\ v2)_n. \tag{3}$$

$$\lambda_n = (\lambda 1\ \lambda 2)_n. \tag{4}$$

where $n = 1, 2, 3, .., n$.

v1 and v2 represent the highest eigenvectors corresponding to eigenvalues $\lambda 1$ and $\lambda 2$, as in (4), of the first and second principal components, respectively. We consider the most suitable feature vectors that best define the frames and achieve continuity and discontinuity between frames, as described in Section IV-A2.

2) EIGENVECTOR FEATURE BASED INTERFRAME DISTANCE CALCULATION

To obtain the interframe distances (dissimilarity signals), we considered the peak values from the eigenvector features. The reason for considering the peak value is that when an abrupt discontinuity occurs between two continuous shots, the eigenfeatures vary significantly, which results in higher peaks compared to the adjacent preframe and postframe; this is known as the cut shot boundary [14]. Many researchers have used the Euclidean distance calculation method, and it has been mentioned as one of the best methods [14], [46]. Hence, the eigenvector feature differences of the peak and its adjacent preframe and postframe values are as follows:

$$\mathrm{PkF_nDif_i} = (\mathrm{Euc}\ (\mathrm{PrF_{i-1}}, \mathrm{PkF_n})\ , \mathrm{Euc}(\mathrm{PkF_{i-1}}, \mathrm{PoF_{i+1}}). \tag{5}$$

$$\mathrm{PkF_n} = (F_n)_\beta\ . \tag{6}$$

where, frame i represents frame n corresponding to $\mathrm{PkF_n}$.

$\mathrm{PkF_n}$ represents eigenvector feature matrices with the highest peaks and the minimum prominence range β. Good results can be obtained when β is $(3 - 4 \times 10^{-6})$, as in (6). $\mathrm{PrF_{i-1}}$, $\mathrm{PoF_{i+1}}$, and Euc represent eigenvector feature matrices of the adjacent preframe and postframe of $\mathrm{PkF_n}$ and the Euclidean distance, respectively, as in (5).

The eigenvector feature differences $D_E(\mathrm{PrF_{i-1}}, \mathrm{PoF_{i+1}})$ between adjacent $\mathrm{PrF_{i-1}}$ and $\mathrm{PoF_{i+1}}$ of $\mathrm{PkF_n}$ are as follows:

$$D_E\left(\mathrm{PrF_{(i-1)j}}, \mathrm{PoF_{(i+1)j}}\right) = \mathrm{Euc}(\mathrm{PrF_{(i-1)j}}, \mathrm{PoF_{(i+1)j}}). \tag{7}$$

$D_E\left(\mathrm{PrF_{(i-1)j}}, \mathrm{PoF_{(i+1)j}}\right)$ represents the feature information of $j-th$ eigenvector feature matrix of frame i. However, high peaks also occur due to object-camera motion and other gradual scene changes. Hence, we must consider another distinct feature that can locate cut shot boundaries accurately, as described in Section IV-A3.

3) ADAPTIVE THRESHOLD

The feature vectors of preframes and postframes with abrupt discontinuities always have values close to zero, whereas wipe and other transitions have gradual changes that include multiple frames; thus, the feature vectors of such preframes and postframes do not provide abrupt discontinuities or values close to zero. Considering this feature, we adaptively determine the thresholds.

The adaptive threshold T for N D_E values used to determine the cut shot boundary can be obtained as follows:

$$T = 1/N \sum_{1}^{N} D_E\left(\mathrm{PrF_{(i-1)j}}, \mathrm{PoF_{(i+1)j}}\right). \tag{8}$$

After detecting the cut shot boundary, the collection of segmented shots or video clips S_h is defined as follows:

$$S_h = \{\mathrm{Vid}\,(1 \rightarrow (i-1)), \ldots., \mathrm{Vid}\,(h\,(i+1) \rightarrow n)\}. \tag{9}$$

where, Vid represents the video clip, frame i represents frame n corresponding to $\mathrm{PkF_n}$, and h represents the adjacent postframe of frame i. These shots or video clips are fed into the next stage of our proposed method.

Although this threshold can provide good results, some false detections still occur because of sudden movements. Reference [47] proposed an adaptive threshold for flash detection and large-object motion; however, when different types of motion appear such as camera pan and zoom-in/out motion, threshold determination becomes very challenging. Therefore, we propose a candidate selection method to select input video clips for feature extraction and classification using I3DCNN, as described in subsection B.

B. CANDIDATE FRAME SELECTION

In the second stage of our proposed WSCD method, we have described our strategies for selecting candidate frames from the segmented video clips, S_h. Our candidate selection strategy focuses on two points. First, localizing wipe transitions, and second, the video clips are prepared by eliminating normal frames, including object-camera motion, which can reduce the computational time and falsely detected cut shot boundaries. Considering this, sliding windows are generated and their edge feature changes are analyzed to select potential wipe transition candidates, as described in the following steps.

1) EDGE FEATURE-BASED SLIDING WINDOW

First, we need to finely analyze S_h to select candidates. Hence, edge features are extracted first, and then the edge feature-based matrices are segmented into 20 frame sliding windows. If C_k represents the edge features of $k-th$ video clips from S_h, then C_k after generating 20 frame sliding windows is written as follows:

$$C_k = \left\{\left(C_{(1+20)}\right), \left(C_{(2+21)}\right), .., \left(C_{(w+r)}\right)\right\}. \tag{10}$$

where frame $w = 1, 2, \ldots, k$, and $r = w + 1$.

These edge feature-based sliding windows contain information on pixel changes in the edges for any movement in the foreground or background of the frame. Therefore, normal frames with continuity have similar edge pixel values, whereas wipe transitions and object-camera motion related transitions have unsteadied and fluctuating edge pixel values. Considering these distinct edge features, we applied linear regression statistics to calculate the slope angles of each sliding window, as described in Section IV-B2.

2) LINEAR REGRESSION AND THE SLOPE ANGLE

To obtain the slope angle, we first need to draw the best curve-fitting slope on each edge-feature-based window. We use linear regression to obtain the slope. Linear regression is a polynomial regression of the first degree.

Therefore, if (f_{kq}, C_{kq}) are the data points for frames f_{kq} and edge features C_{kq} for the $q-th$ 20-frame sliding window from the $k-th$ video clip, then the linear regression equation $R(f_{kq})$ of a first-degree polynomial can be obtained as follows:

$$R(f_{kq}) = \sum_{q=1}^{k} [C_{kq} - \left(mf_{kq} + b\right)]^2. \tag{11}$$

where, $q = \{1, 2, \ldots., 20\}$, m represents slope and b is the C_{kq}-intercept.

By solving the partial derivatives of (11) with respect to slope m and intercept b, we can obtain $\left(\frac{b}{m}\right)$ matrices as follows:

$$\partial R/\partial b = \sum_{q=1}^{k} 2(b + mf_{kq} - C_{kq}). \tag{12}$$

$$\partial R/\partial m = \sum_{q=1}^{k} 2f_{kq}(b + mf_{kq} - C_{kq}). \tag{13}$$

From the slope for each $q-th$ sliding window, the slope angle is written as follows:

$$a_{kq} = \tan^{-1} m. \tag{14}$$

The distinguishable slope angle feature among normal frames in object-camera motion (camera pan, large object motion, and zoom-in/out), and wipe frames with object-camera motion is that the transition in normal frames in object-camera motion is steadier than that is wipe transition frames in object-camera motion. Hence, this results in negative slope angles in normal frames in object-camera motion, and positive slope angles in wipe transition frames in object-camera motion. This is because when wipe transitions occur, the frames receive new wipe patterns along with object-camera motion or no motion, and most importantly the velocity of a wipe pattern transition does not match the velocity of object-camera motion. Therefore, we consider sliding windows with positive slope angles as our candidates.

However, as the beginning 2-4 frames of wipe transitions contain very few wipe patterns, negative slope angles

are obtained for those frames, resulting in misclassification. Hence, these key frames are fed into the I3DCNN in the next stage of our proposed method for extracting very deep spatial-temporal features, and the features are classified into binary classes, as described in subsection C.

C. WIPE SCENE CHANGE DETECTION IN OBJECT-CAMERA MOTION

There are several existing 3D-ConvNet-based (3DCNN) models for deep feature extraction because 3D-ConvNet not only preserves features from the time domain but also preserves the best information in the temporal domain, thus making it very suitable for video data classification. Reference [14] used 3D-ConvNet for deep feature extraction in the spatial and temporal domains and then performed classification tasks for gradual shot changes (fade-in/out, dissolve, and swipe), which does not include wipe scene changes. In their method, 3D-ConvNet extracts significant information of RGB features and motion features, but to handle the velocity imbalance between large-scale object-camera motion and wipe pattern change motion, we need to include a separate motion-based algorithm along with 3D-ConvNet that can extract very deep motion features and improve the performance. Hence, we use a two-stream inflated 3D-ConvNet (I3DCNN) which is fused with the optical flow velocity algorithm, as shown in Fig. 9. This I3DCNN model is inflated with two subnetworks, one is 3D-ConvNet, and the other is the optical flow velocity algorithm, where 3D-ConvNet provides the prediction results from the RGB stream and the optical flow velocity gives the prediction results from the motion stream, which are then combined in the last step to provide the finale prediction result. This approach is known as two-stream inflated 3D-ConvNet (I3DCNN). Reference [43] recorded the statistical results on the advantages of I3DCNN in detail. Adding the optical flow velocity significantly improved the performance, even though 3D-ConvNet itself can learn spatial-temporal features from the RGB stream.

1) SPATIAL FEATURE EXTRACTION

All sliding window video clips, including key frames from the last subsection B, are fed to the I3DCNN model for very deep spatial-temporal feature extraction, and spatial-temporal feature learning from the RGB stream is discussed in this step. As shown in Fig. 10, our 3D-ConvNet architecture consists of 22 convolutional layers (these layers contain a total of 9 inception modules), 5 pooling layers (4 max-pooling and 1 average-pooling layers), 1 fully connected layer for spatial-temporal feature learning from the RGB stream, and finally 1 softmax layer to classify wipe transitions in object camera motion. To inflate 2D-ConvNet to 3D-ConvNet, we add another dimension of time by changing the square size N×N to into cubic N×N×N filter size, weight size, and bias size. The number of convolutional kernels is set to 64. In the first inception layer, the first convolutional network has stride 2, and after the 7 × 7 average-pooling layer, the model proceeds

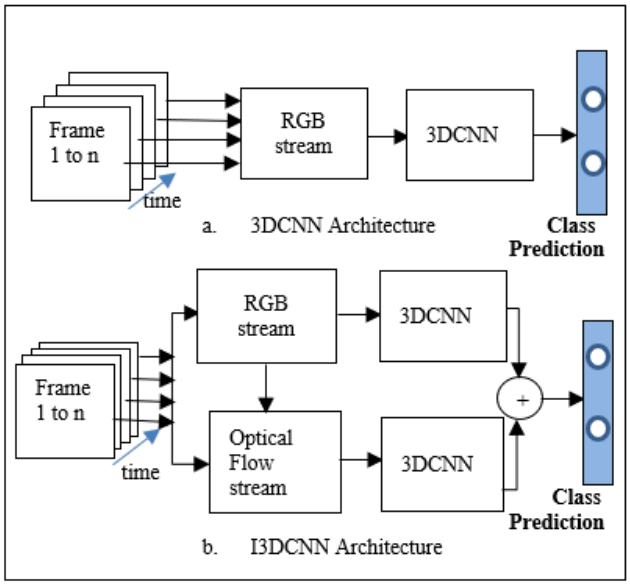

FIGURE 9. Comparison between the 3DCNN and I3DCNN architectures.

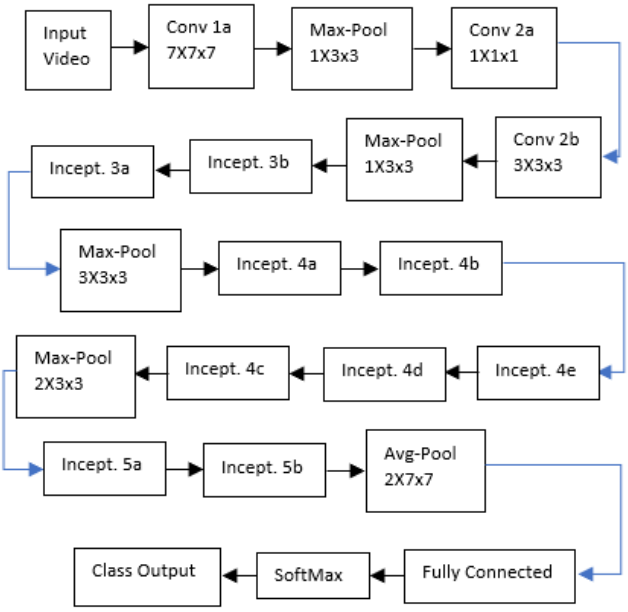

FIGURE 10. Representation of the 3D-ConvNet architecture in the I3DCNN where Incept. represents the inception module.

to the linear classification layer. Reference [43] processed 25 frames per second as the input video frames; however, we use 20 frame sliding windows. In the first and second max-pooling layers, we use 1 × 3×3 kernel with a stride in time; in all other max-pooling layers, we use a 3 × 3×3 kernel with stride 2 in time. The finale average pooling layer has used 2 × 7×7 kernel. Before the inception layer, the first convolutional layer has a kernel size of 7 × 7×7 with stride 2, the second convolutional layer has kernel size of 1 × 1×1 and the third convolutional layer has a kernel size of 3 × 3×3. The kernel sizes of the convolutional layers inside

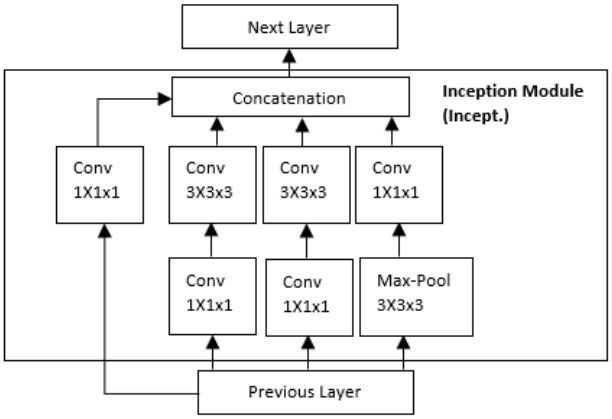

FIGURE 11. Representation of the inception module (Incept.) in the 3D-CNN architecture of I3DCNN.

the inception layers are shown in Fig. 11. In these inception layers, which contain convolutional layers and pooling layers, semantic feature vectors that can define the spatial-temporal features of wipe transitions are stored. However, to predict wipe changes in large-scale object-camera motion, we need to add the optical flow velocity to improve the performance, as described in Section IV-C2.

2) MOTION FEATURE EXTRACTION

As the second stream of the I3DCNN method, optical flow velocity-based temporal feature extraction is discussed in this step. Whereas the temporal feature extraction process runs iteratively in the optical flow fields, 3D-ConvNet runs through feedforward computation while directly learning both spatial-temporal features from RGB inputs [43]. Because of this lack of recurrence, the I3DCNN improves the performance for large-scale motion information with wipe pattern change information, where stream one, including 3D-ConvNet, is trained on RGB inputs and stream two including velocity information is computed on optical flow inputs. Stream two has two channels for velocity information: (V_x, V_y). V_x and V_y represent the velocity flow toward the x and y components, that is, the change in position of each pixel in the (x, y) space with respect to time or frame numbers. We train the two subnetworks separately and then obtain combined prediction results that classified the sliding window video clips into wipe no-motion transitions and wipe in object-camera motion transitions. Classifying wipe transitions as in-motion and no-motion helps eliminate the only motion or normal frames, as only motion frames do not carry the optical flow velocity information of wipe pattern changes. The experimental results section describes the accuracy achieved by our proposed WSCD method.

3) DETECTION REVIEWING AND MERGING STRATEGY

Finally, we need a merging strategy that can eliminate the overlapped wipe frames from the sliding windows, and then the nonoverlapping frames need to be merged to obtain

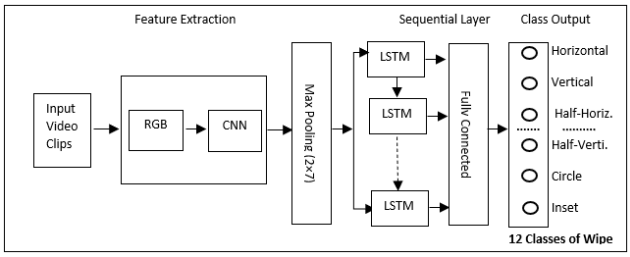

FIGURE 12. Architecture that classifies input video clips into 12 types of wipe shape patterns using CNN-LSTM.

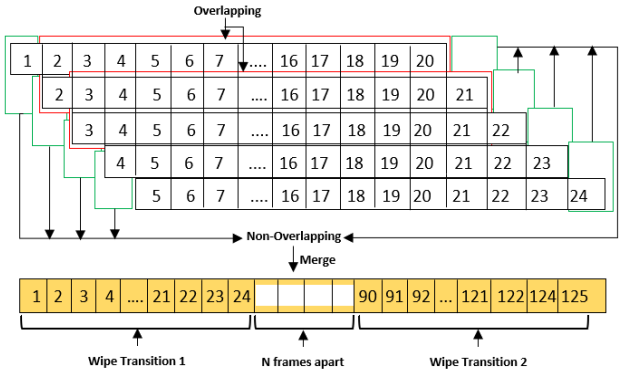

FIGURE 13. Merging strategy of the proposed method.

a complete wipe transition. However, before merging the videos, we need to review the detection results to eliminate further false detections due to object-camera motion in the sliding windows. To do so, we further classify the pattern of the wipe type from the detected wipe in-motion and no-motion classes by using transfer learning (ResNet-50 CNN) for feature extraction and an LSTM sequential layer for classification, as shown in Fig. 12. Finally, sliding windows are merged to obtain a complete wipe transition, as shown in Fig. 13.

In the 20-frame sliding window, except for the first frame, the other 19 frames overlap twice each up to the $20^{-\text{th}}$sliding window. For example, between the $q - th$ sliding window, which has frames $q = \{1, 2, \ldots, 20\}$, and another $q + \text{w}$ sliding window, which has frames $q+w = \{21, 22, \ldots, 40\}$, we have 19 sliding windows, in which each frame $q = \{1, 2, \ldots, 20\}$ has two overlapped sections. Hence, in our merging strategy, we eliminate frames that occur two times and merge frames that occur once. To merge corresponding wipe frames, we compare the continuity of frame numbers in classified sliding windows with the key frame numbers of $k - th$ shot from subsection B. If the ending frame number of the $q - th$ sliding window is N frames away from the beginning frame number of the adjacent sliding window, then we consider it as the start of another wipe transition, and therefore obtain G as the output.

We summarize our proposed wipe scene change detection in object-camera motion (WSCD) method in Alg.1.

Algorithm 1 Proposed Wipe Scene Change Detection (WSCD) in Object-Camera Motion Based on Linear Regression and an Inflated Spatial-Motion Neural Network

Input:

$X = [x_1, x_2, \ldots, \ldots, x_n]-$ frames of one test video.

$S = [S_1, S_2, \ldots, \ldots, S_h] -$ segmented shots or video clips by cut shot boundary detection.

$C = [C_1, C_2, \ldots, \ldots, C_k]-$ Selected candidate clips in a 20-frames sliding window.

Output:

$G = [c_1, c_2, \ldots, \ldots, c_k]-$ Detected motionless wipe and motion wipe scene change clips.

Step 1. A dimensionality reduction method for large input that preserves the most appropriate eigenvector feature information by principal components analysis (PCA) and an interframe discontinuity calculation is proposed for cut shot boundary detection and segmentation of X into S with fast computation.

Step 2. A candidate selection method by fusing Canny edge features and linear regression is proposed by calculating the slope angle information changes on each 20-frame sliding window C from S.

Step 3. A very deep two-stream inflated 3DCNN and optical flow velocity model are utilized to classify key frame C in the wipe scene change in object-camera motion and wipe scene change in no-motion.

Step 4. A deep LSTM model is used to further classify the detected wipe scene change in 12 types of wipe patterns for detection review.

Step 5. A merging strategy is used to merge corresponding wipe frames in a complete wipe scene change transition by eliminating the overlapped wipe frames from sliding windows and obtaining output G.

V. EXPERIMENTAL ANALYSIS

Our proposed WSCD method is tested on two datasets: one is the TRECVID dataset, and the other is our own Multimedia dataset. As our approach is to detect wipe transitions in object-camera motion, we needed a well-balanced dataset for training and evaluation, but the TRECVID dataset lacks this; hence, we created our own Multimedia dataset for training and evaluation, as explained in subsections A and B.

A. TRECVID DATASET

We perform our experiments on the TRECVID 2001 and TRECVID 2007 datasets, which are provided by the US National Institute of Standards (NIST) benchmark dataset [60]. Although the content in this dataset vary and includes News, Sports, and TV series, which is sufficient for creating training datasets for other gradual transitions (dissolve and fade-in/out), it is not sufficient for wipe transitions in object-camera motion. Hence, we created a Multimedia dataset. We split the Multimedia dataset for training and evaluation and use the TRECVID dataset for evaluation only. To prove the superiority of our proposed method, we primarily compared the performance of cut shot boundary detection with the HSV histogram, HLFPN-Keypoint matching, and C3D-based methods and the performance of wipe scene change detection with pixel-border, macroblock-type information, and deepSBD-based methods.

B. MULTIMEDIA DATASET

Our Multimedia dataset contains videos of news reports and movies. We collected news report videos from the BBC news channel and collected ''Star Wars'' and ''Sherlock Holmes'' movie datasets available on social media. In the ''News Reporting'' category we have collected 30 videos that are 4-6 minutes long, and in the ''Movies'' category we collected 10 videos that are 2-4 minutes long. We split this dataset into training and evaluation datasets. Unlike the TRECVID datasets, our Multimedia dataset contains various wipe types with object-camera motion from recent years. We prepared a ground-truth dataset and used it to evaluate the performance of the proposed method.

C. PERFORMANCE EVALUATION CRITERIA

To illustrate the efficiency of our proposed WBSC method, we compare the performance of our proposed method with that of the existing methods using the following criteria:

$$\text{Precision}\,(P) = \{Z_c/(Z_c + Z_f)\} \times 100. \quad (15)$$

$$\text{Recall}\,(R) = \{Z_c/(Z_c + Z_m)\} \times 100. \quad (16)$$

$$F - \text{Score}\,(\text{F1}) = \{2PR/(P + R)\} \times 100. \quad (17)$$

where Z_c, Z_f, and Z_m represent the number of correctly detected frames, number of falsely detected frames, and number of missed detected frames in the cut or wipe transitions, respectively. Precision (P) and Recall (R) are the rates of false positives and false negatives, respectively. P and R provide higher values when false and miss detection rates are low. The F − Score (F1) is a measure that considers both the P and R values.

D. IMPLEMENTATION DETAILS

The proposed method is built on the deep learning I3DCNN model, which uses GoogleNet and the optical flow algorithm as two of the base subnetworks and MATLAB software version R2021b. Our 3D-ConvNet has three channels, and the optical flow has two channels. We found it helpful to resize the video frames to 112 × 112 to preserve maximum temporal information in the optical flow subnetwork. For the 3D-ConvNet subnetwork, we resized the video frames to 224 × 224. We set the mini-batch size to 10 video clips, iteration to 500, the base learning rate to 1×10^{-4}, and the momentum to 0.9. All experiments are conducted on an Nvidia Titan 1650i GPU with Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz with a 64-bit operating system and MATLAB software version R2021b.

TABLE 2. Performance of the proposed method on cut SBD.

Dataset		Actual n	Detected n	P	R	F1
TREC 2001	D1	98	93	0.958	0.948	0.952
	D2	42	38	0.950	0.904	0.926
	D3	40	35	0.921	0.875	0.897
TREC 2007	D4	107	101	0.961	0.943	0.951
	D5	88	85	0.977	0.965	0.970
	D6	105	100	0.943	0.952	0.947
Multi media	M1	76	71	0.959	0.934	0.946
	M2	89	82	0.964	0.920	0.941
	M3	45	41	0.891	0.911	0.901
	M4	60	54	0.885	0.900	0.892
Average				0.941	0.925	0.932

TABLE 3. Comparison of frameworks for cut SBD on the TRECVID dataset.

Method	P	R	F1
HSV histogram [50]	0.911	0.853	**0.878**
HLFPN + Keypoint Matching [49]	0.865	0.723	**0.769**
C3D based [14]	0.813	0.796	**0.804**
Ours	0.941	0.925	**0.932**

E. CUT SHOT BOUNDARY DETECTION

We evaluated the proposed method on three videos from TRECVID 2001 dataset, three videos from the 2007 dataset, and four videos from our Multimedia dataset. The value of minimum peak prominence β is set to 4×10^{-6}. The detection accuracy of our proposed method on the TRECVID and Multimedia datasets is presented in Table 2.

Here, n represent total number of frames in the cut transition, and (D1, D2, . . . , M3, M4) represent the testing video clips of the corresponding dataset.

To demonstrate the efficiency of our proposed method, we compare the proposed algorithm with the blocked histogram and C3D-based method. The method comparison results are shown in Table 3. By using appropriate eigenvector features from the first and second principal components only, the dimensionality of the large input dataset is reduced with a minimal loss of information and extremely fast computation.

Generating eigenvector feature matrices by PCA and employing the threshold calculation strategies in our proposed method reduces the rate of missed detection, thereby providing a higher F1 value. Although the HSV color histogram feature-based method can provide a good precision value, it suffers from more missed detections than the proposed method. The C3D-based method uses CNN features that can detect cut and gradual transitions simultaneously; however, it suffers from false detections and requires a longer computational time.

TABLE 4. Comparison of candidate frame selection frameworks.

Method	FRP on TREC 2007 (%)
Single-Plane [48]	0.217
STKT+Bisection [10]	**0.183**
Ours	**0.193**

Additionally, many research papers have proposed two or multiple feature-based methods, such as SURF+HSV [13] for cut shot boundary detection, which can analyze spatial and temporal information, resulting in a lower false detection rate. In addition, many recent related works have achieved higher precision (more than 96%), however wipe transitions have not been investigated in these approaches [53], [54], [55], [56], [57]. Therefore, we propose a candidate selection method that can increase the overall accuracy of the proposed WSCD method, as illustrated in the next subsection.

F. CANDIDATE FRAME SELECTION

The experiment in this subsection is conducted on segmented shots or video clips to determine the possible location of wipe scene changes. The main objective of candidate key frame selection is to reduce the processing time during classification in a neural network by feeding a smaller number of frames while obtaining a higher scene change detection accuracy. As the output from this experiment, we obtain segmented candidate video clips containing wipe in-motion or no-motion transition key frames. References [10] and [48] evaluated candidate key frame selection criteria based on the following equation:

$$\mathrm{FRP} = (Z_{seg}/Z_{tot}) \times 100. \tag{18}$$

where Z_{seg} and Z_{tot} represent the total number of key frames in the candidate segments and the total number of frames in the video shot, respectively.

Hence, we compare the performance of our proposed method with existing works [10], [48] using the FRP measurements, as shown in Table 4. As the TRECVID 2001 dataset does not contain precise wipe transition effects, we have compared it with the TRECVID 2007 dataset. Finding the location of wipe scene changes requires temporal information; hence, we found it helpful to segment the shots into a 20-frame sliding window and track the slope angle changes or fluctuations. Because of the velocity imbalance between the wipe pattern transition and large-scale object-camera motions, such as camera pan, zoom-in/out, and large

object transition motions, slope angles received higher values than normal frame transition. Hence, we considered positive values for the location of wipe scene changes. On the other hand, we considered all the negative values of slope angles as normal frame changes and only motion frame changes, and therefore eliminated them from the candidates.

TABLE 5. Dataset 1 distribution for I3DCNN.

Class	Train (Multimedia)	Validation (Multimedia)	Test (Multimedia + TREC 2007)
WIP	2510	740	1856
WIM	2487	622	1690
Total	4997	1362	3546

WIP and WIM represent no-motion and in-motion wipe transitions, respectively.

TABLE 6. Dataset 2 distribution for CNN-LSTM.

Wipe Type	Train	Test
Horizontal (Ho)	256	35
Vertical (Ve)	260	36
Door (Do)	140	28
Clockwise (Cl)	125	25
Diagonal (Di)	130	30
Checker (Ch)	130	24
Irish (Ir)	155	25
Circle (Ci)	245	30
Pinwheel (Pi)	125	20
Inset-Wipe (In)	140	23
Half-Horizontal (Hh)	260	35
Half-Vertical (Hv)	264	36
Total	2230	346

Our combined edge feature and slope angle calculation technique provide a higher wipe scene change detection accuracy with a lower FRP rate. Although reference [10] achieved a lower FRP rate than our method, our specific aim is to detect wipe scene changes that have unique characteristics compared to other gradual transitions, and reference [10] did not specifically mention detecting various wipe scene changes in large-scale object-camera motion in the paper. However, 2-4 beginning and ending frames of wipe scene changes still suffer from missed detections using our candidate selection method. In addition, some object-camera motion frames that have shapes that are similar to the wipe pattern changes obtain positive slope angle values; hence, they are falsely detected as wipe transitions. Therefore, we employed a dense temporal feature extraction technique as described in subsection G.

G. WIPE SCENE CHANGE DETECTION IN OBJECT-CAMERA MOTION

Experiments are conducted on the selected candidate window clips. The dataset distribution for training and validation is 80:20. We prepared two sets of datasets: Dataset 1 for two-stream I3DCNN classification and Dataset 2 for LSTM classification, as listed in Tables 5 and 6, respectively.

As we classify candidate sliding window clips into WIP and WIM classes, we set a score threshold $T_s = 0.80$ to label the final output. Any sliding window with a score lower than

TABLE 7. Performance of the proposed method on wipe transitions detection.

Data	Class	P	R	F1	Time (s)
TREC 2007	WIP	0.971	0.906	0.937	51.23
	WIM	0.883	0.963	0.921	
Multi media	WIP	0.979	0.968	0.973	88.04
	WIM	0.963	0.976	0.969	
Average		**0.949**	**0.953**	**0.950**	**69.63**

TABLE 8. Performance of the proposed method on wipe patterns detection.

Class	P	R	F-1
Horizontal (Ho)	0.970	0.942	0.956
Vertical (Ve)	0.921	0.972	0.945
Door (Do)	0.925	0.892	0.908
Clockwise (Cl)	0.884	0.920	0.901
Diagonal (Di)	0.903	0.933	0.917
Checker (Ch)	0.833	0.869	0.850
Irish (Ir)	0.807	0.840	0.823
Circle (Cr)	0.900	0.900	0.900
Pinwheel (Pi)	0.894	0.850	0.871
Inset-Wipe (In)	0.904	0.863	0.881
Half-Horizontal (Hh)	0.888	0.914	0.901
Half-Vertical (Hv)	0.939	0.861	0.898
Average	**0.897**	0.896	**0.896**

T_s is considered as a nonwipe object-camera motion; hence, it is not considered in the final output set. In the next step, we classified detected wipe transitions in to 12 different patterns by adding CNN-LSTM layers as our detection review technique. Thus, our proposed system efficiently provides higher accuracy with lower computational time, as shown in Table 7 and Table 8. The improvement in the Precision(P) and Recall(R) demonstrates reduced falsely detected wipe transitions and reduced missed detected wipe transitions, respectively. A good F1 rate represents the harmonic mean of the P and R rates. Therefore, our proposed WSCD method achieves promising results with a good balance of the P and R rates.

To clarify the obtained results in Table 7 and Table 8, a graphical representation of the training and validation accuracy with the loss is presented in Fig. 14, and confusion matrices of the experimented dataset are depicted in Fig. 15. These confusion matrices represent the number of correctly and incorrectly detected wipe transitions and the patterns of each class. Among the total number of test datasets, 1886 data points were actual wipe video clips, and the rest were falsely detected as localized wipe transitions during the candidate key-frame selection step because of large

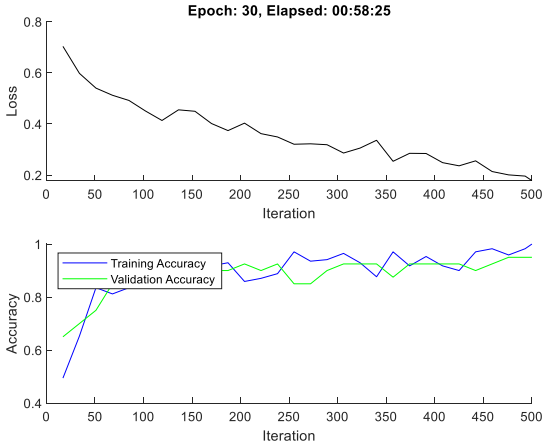

FIGURE 14. Representation of training and validation accuracy with loss.

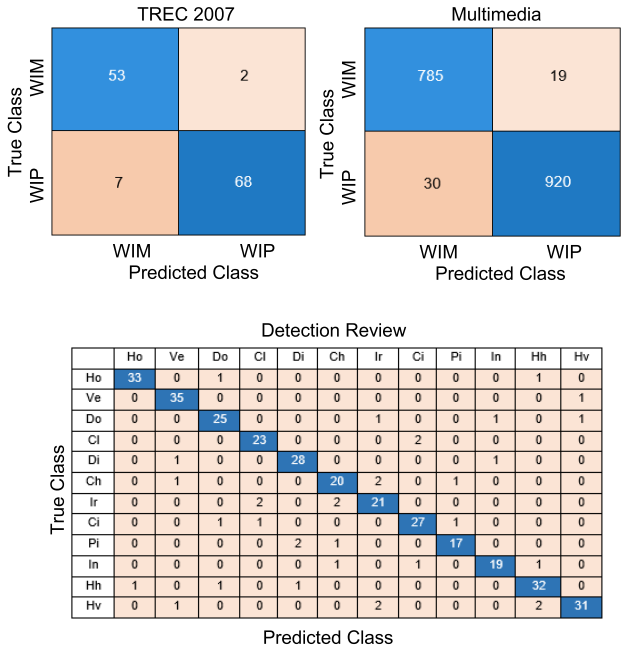

FIGURE 15. Confusion matrix representation for the obtained results.

object-camera motion. As the final class output label, our classifier correctly classified 1828 clips, and 58 windows were incorrectly classified into WIP and WIM classes. From these correctly classified video clips, we have randomly selected 346 video clips and classified them into 12 types of wipe patterns. A total of 311 clips were correctly classified and 35 clips were incorrectly classified.

The proposed detection reviewing technique demonstrates high performance for the wipe pattern classes "*Horizontal (Ho)*" and "*Vertical (Ve)*", as indicated by the highest achieved F-1 rates. However, comparatively lower F-1 rates are observed in wipe pattern classes "*Checker (Ch)*" and "*Irish (Ir)*". Pattern similarity in long wipe transitions increases false detection rate, as shown in Fig. 16.

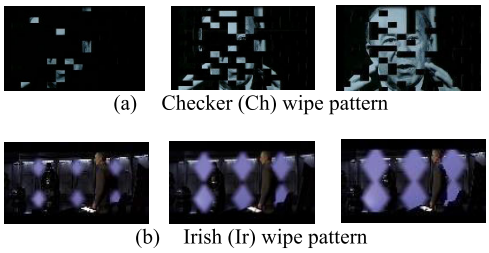

FIGURE 16. Cases of false detection in wipe pattern classes.

TABLE 9. Comparison with different wipe detection techniques.

Method	WIP			WIM		
	P	R	F1	P	R	F1
Pixel-Border [15]	0.924	0.928	**0.925**	--	--	--
Histogram Space [12]	--	--	--	0.631	0.967	**0.763**
MB+SGOP [16]	0.888	0.969	**0.926**	0.705	1.000	**0.826**
DeepSBD [23]	0.976	0.936	**0.956**	--	--	--
Ours (WSCD)	0.975	0.937	**0.956**	0.923	0.969	**0.945**

TABLE 10. Comparison with different gradual transition detection techniques evaluated on the TRECVID dataset.

Method	P	R	F1
SURF+ HSV [13]	0.957	0.861	**0.906**
WHT-SBD [51]	0.928	0.894	**0.911**
Best TRECVID Performer [52]	0.791	0.788	**0.790**
STKP+SVM [10]	0.992	0.934	**0.961**
TSSBD [14]	0.932	0.938	**0.935**
Ours (WSCD)	0.949	**0.953**	**0.950**

We compare our methods with existing techniques under two conditions: first, it is compared with the existing techniques that include wipe in-motion (WIM) and wipe no-motion (WIP) detections, as shown in Table 9; second, it is compared with the existing techniques that do not include wipe detection but detect average gradual shot boundaries or scene changes, as shown in Table 10.

Our deep motion feature extraction and detection reviewing technique improves the motion wipe scene change detection accuracy by 11.9%. The pixel-border trajectory-based method [15] and histogram feature [12] provide higher accuracy in wipe no-motion class detection only, but for large object-camera motion, we need a high temporal feature. In contrast, the Macroblock and SGOP-based (threshold set 2) [16] methods extract temporal features; however, in the case of large-scale object-camera motion, a good balance of spatial and temporal features is lacking.

To address large-scale motion information, these methods need to be improved by very deep motion extraction

methods, such as the two-stream I3DCNN. Our proposed method improves the overall accuracy to 95.1% compared to the existing related works. The TSSBD [14] method detects cut SBD first and then detects other gradual scene changes; however, their system processes all video frames, which is time-consuming. The WHT-SBD method uses multifeature for wipe scene change detection; however, their method also suffers from problems similar to TSSBD. The STKP+SVM [10] method can also detect candidate frames and has a lower FRP rate than our method; however, our method can detect not only the locations but also the various patterns of wipe scene changes.

VI. DISCUSSION

In our proposed WSCD scheme, the post- and preframe pair interframe difference D_E and Threshold T have a direct impact on the candidate selection that locates potential wipe transition segments. This process requires two steps: cut shot boundary detection for video segmentation and candidate frame extraction from segmented video clips for potential wipe selection.

Initially, we selected the frame pair interframe difference D_E and minimum prominence β as cut shot boundary detection criteria, where β was set from 3×10^{-6} to 4×10^{-6}. All peaks higher than the range of β were considered cut shot boundaries, as well as video segmentation regions. However, many large object movements, and flash effects are in the same β range as the cut shot boundary in the video, resulting in missed detection. Therefore, the threshold T is carefully adapted, and it is determined based on the correlation between the preframes and postframes of cut shot boundaries. T is set from –0.003 to +0.005; the peaks that have satisfied all three parameters, D_E, β, and T are considered cut shot boundaries; and unsatisfied peaks are discarded. Hence, setting all three parameters can achieve significantly good precision while maintaining good recall in cut shot boundary detection. Thereafter, the videos are segmented into clips using the detected cut shot boundaries.

Second, the process of candidate selection from segmented video frames has a direct influence on wipe transition detection. The purpose of this process is to eliminate normal and object-camera motion frames such as camera pan, zoom-in/out, and large object movements from each segment of a video, which has been one of the main limitations in many alternative works, as discussed in the literature review in Table 1. Therefore, we adapt the combined Canny edge pixel value feature and linear regression-based mathematical model for potential wipe localization, where the parameters C_k and a_{kq} are the criteria to distinguish potential wipe transitions from object-camera motions. To obtain sufficient motion information, sliding windows are generated on each video segment, and the length of each window is set to 20-frames. Thereafter, Canny edge feature information C_k is extracted from each window on which the slope angles a_{kq} are calculated. According to the analysis of edge feature changes, normal and object-camera motion transitions are steadier than wipe transitions; therefore, object-camera motions provide more $-a_{kq}$ values, whereas wipe transitions provide more $+a_{kq}$ values. All positive values of the slope angle $+a_{kq}$ are selected as our candidate segments, and thereafter only fed these candidate frames into the classification step.

TABLE 11. Comparison with recent techniques for gradual transition detection evaluated on the TRECVID 2007 dataset (2018-2022).

Method	P	R	F1
	Machine learning based approach		
TSSBD [14]	0.932	0.938	**0.935**
VSBD-POCS [56]	0.919	0.885	**0.904**
Ours (WSCD)	0.949	0.953	**0.950**

In this paper, we investigate our candidate frames selection method using the FRP rate, which is proportional to the total number of candidate frames Z_{seg} and inversely proportional to the total number of frames in segment Z_{tot}. A lower FRP rate indicates a better candidate selection method, and our method achieves 2.4% lower FRP rate than the existing single-plane-based method [48]. However, in some cases, the beginning and ending 2-4 frames of a complete wipe transition are miss detected as $-a_{kq}$ due to small changes in the transition pattern, and fast object-camera motion transitions are falsely detected as $+a_{kq}$. Therefore, we classify our selected candidates using an inflated very deep 3DCNN and dense optical flow motion-based (I3DCNN) model into wipe no-motion (WIP) and wipe in-motion (WIM) classes.

The performance of the classification step is highly dependent on the training dataset. The traditional TRECVID 2007 dataset has a smaller number of wipe transition patterns. Therefore, we used Multimedia dataset to train the network. Our trained I3DCNN model is tested on the TRECVID 2007 dataset, and it achieves good recall and precision values for both the WIP and WIM classes. To prove the efficiency of our proposed scheme, we compare our results with recent machine learning-based approaches on the TRECVID 2007 dataset, as shown in Table 11. Our proposed scheme efficiently improves the precision, recall, and F1−score by 3.0%, 6.8%, and 4.6%, respectively, compared to recent alternative works [14], [56].

We also investigate 12 different common wipe patterns used in recent years by classifying the WIP and WIM clips using the CNN-LSTM network. The purpose of this method is to review detected WIP and WIM clips. However, in some cases, the wipe transition in lengthy zoom-in/out (more than 50 frames) motion frames achieves low precision, which is a limitation of our proposed method. Another limitation is that fast illumination changes directly affect the cut shot boundary detection method. These two problems can be solved using histogram-based illumination change analysis and motion pattern classification, which will be explored in our future work.

In contrast, our proposed method shows a remarkable improvement in wipe scene change detection in the presence

of several object-camera motions. Moreover, it reduces the processing time by proposing a candidate frame selection method and resolves motion sensitivity issues in existing related works. Furthermore, the comparison results with the existing alternative methods on the TRECVID 2001 and TRECVID 2007 datasets show that our proposed method provides high accuracy for the overall detection with the interframe distance, the candidate selection method, and the wipe in-motion or no-motion classification approach.

VII. CONCLUSION

Content-based video analysis is a challenging task for object-camera motion-based videos in Multimedia applications. With the extensive growth of digital video usage, various object-camera motion-based complex video editing effects are being widely broadcast for viewers' attention. Wipe scene change detection is therefore considered to be a highly important preliminary step toward obtaining high-level content analysis using low-level information.

Wipe transitions are multipattern and arbitrary speed-based gradual scene changing effect that often generates velocity imbalance and pattern similarity confusion in the presence of object-camera motion (camera pan, zoom-in/out, or large object movement). Because of this, not only do existing methods suffer from lower accuracy rates in wipe scene change detection, but these transitions also make detecting other scene change effects difficult. Moreover, conventional methods have the limitations of high computation costs and time.

Therefore, our proposed wipe scene change detection method is designed to focus on detecting nonmotion and object-camera motion-based wipe transitions by analyzing velocity transition patterns. In addition, a candidate segment selection approach is implemented by observing interframe distances to locate wipe transitions, aiming to process fewer frames to minimize the computational cost. Our system improves the wipe in-motion detection accuracy and the overall detection accuracy by 11.9% and 7.45%, respectively. Furthermore, a detection reviewing technique is performed using various wipe shape pattern classifications to re-evaluate our detection confidence.

The experimental results show that our proposed method outperforms existing wipe scene change detection methods in terms of various object-camera motion-based wipe transition detection tasks and fast processing. To reduce the velocity imbalance and pattern similarity confusion, we obtain a combination of spatial-motion feature-based patterns from different motion activities. We also test two new types of half-wipe patterns that can be easily confused with object motion. Our conclusion is that lengthy zoom-in/out motion transition frames and fast illumination changes are the current limitations of our proposed method, which directly affect the performance of the candidate frame selection method. Therefore, our future work will focus on motion similarity analysis under illumination changes to improve scene change detection.

ACKNOWLEDGMENT

The authors acknowledge the King Mongkut's University of Technology Thonburi and Srinakharinwirot University for their kind co-operation and support.

REFERENCES

[1] G.M.D.T Forecast. (Feb. 2019). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022*. White Paper Cisco Public. [Online]. Available: http://media.mediapost.com/uploads/CiscoForecast.pdf

[2] C. Zhang, Z. Liu, C. Bi, and S. Chang, "Dependent motion segmentation in moving camera videos: A survey," *IEEE Access*, vol. 6, pp. 55963–55975, 2018, doi: 10.1109/ACCESS.2018.2872733.

[3] S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Trajectory-based surveillance analysis: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1985–1987, Jul. 2019, doi: 10.1109/TCSVT.2018.2857489.

[4] M. Wang, G.-W. Yang, S.-M. Hu, S.-T. Yau, and A. Shamir, "Write-a-video: Computational video montage from themed text," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–13, Nov. 2019, doi: 10.1145/3355089.3356520.

[5] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, pp. 168–186, Feb. 2007, doi: 10.1109/TCSVT.2006.888023.

[6] L. Baraldi, C. Grana, and R. Cucchiara, "Recognizing and presenting the storytelling video structure with deep multimodal networks," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 955–968, May 2017, doi: 10.1109/TMM.2016.2644872.

[7] C. Cotsaces, N. Nikolaidis, and I. Pitas, "Video shot detection and condensed representation. A review," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 28–37, Mar. 2006, doi: 10.1109/MSP.2006.1621446.

[8] D. Rotman, D. Porat, and G. Ashour, "Robust and efficient video scene detection using optimal sequential grouping," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 275–280, doi: 10.1109/ISM.2016.0061.

[9] S. H. Abdulhussain, A. R. Ramli, B. M. Mahmmod, M. I. Saripan, S. A. R. Al-Haddad, and W. A. Jassim, "Shot boundary detection based on orthogonal polynomial," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 20361–20382, 2019, doi: 10.1007/s11042-019-7364-3.

[10] Z. N. Idan, S. H. Abdulhussain, B. M. Mahmmod, K. A. Al-Utaibi, S. A. R. Al-Hadad, and S. M. Sait, "Fast shot boundary detection based on separable moments and support vector machine," *IEEE Access*, vol. 9, pp. 106412–106427, 2021, doi: 10.1109/ACCESS.2021.3100139.

[11] S. H. Abdulhussain, A. R. Ramli, M. I. Saripan, B. M. Mahmmod, S. A. R. Al-Haddad, and W. A. Jassim, "Methods and challenges in shot boundary detection: A review," *Entropy*, vol. 20, no. 4, p. 214, Mar. 2018, doi: 10.3390/e20040214.

[12] R. A. Joyce and B. Liu, "Temporal segmentation of video using frame and histogram space," *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 130–140, Feb. 2006, doi: 10.1109/TMM.2005.861285.

[13] S. Tippaya, S. Sitjongsataporn, T. Tan, M. M. Khan, and K. Chamnongthai, "Multi-modal visual features-based video shot boundary detection," *IEEE Access*, vol. 5, pp. 12563–12575, 2017, doi: 10.1109/ACCESS.2017.2717998.

[14] L. Wu, S. Zhang, M. Jian, Z. Lu, and D. Wang, "Two stage shot boundary detection via feature fusion and spatial–temporal convolutional neural networks," *IEEE Access*, vol. 7, pp. 77268–77276, 2019, doi: 10.1109/ACCESS.2019.2922038.

[15] S. Li and M. C. Lee, "Effective detection of various wipe transitions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 663–673, Jun. 2007, doi: 10.1109/TCSVT.2007.896621.

[16] S.-C. Pei and Y.-Z. Chou, "Effective wipe detection in MPEG compressed video using macro block type information," *IEEE Trans. Multimedia*, vol. 4, no. 3, pp. 309–319, Sep. 2002, doi: 10.1109/TMM.2002.802841.

[17] J. Bescos, G. Cisneros, J. M. Martinez, J. M. Menendez, and J. Cabrera, "A unified model for techniques on video-shot transition detection," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 293–307, Apr. 2005, doi: 10.1109/TMM.2004.840598.

[18] H.-H. Yu and W. Wolf, "A multi-resolution video segmentation scheme for wipe transition identification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 1998, pp. 2965–2968, doi: 10.1109/ICASSP.1998.678148.

[19] M. Wu, W. Wolf, and B. Liu, "An algorithm for wipe detection," in *Proc. Int. Conf. Image Processing. (ICIP)*, Oct. 1998, pp. 893–897, doi: 10.1109/ICIP.1998.723664.

[20] K.-D. Seo, S. J. Park, and S.-H. Jung, "Wipe scene-change detector based on visual rhythm spectrum," in *Dig. Tech. Papers Int. Conf. Consum. Electron.*, Jan. 2009, pp. 831–838, doi: 10.1109/icce.2009.5012396.

[21] L. Yufeng, Y. Yinghua, and L. Guiju, "A novel wipe transition detection method based on multi-feature," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining*, Jan. 2010, pp. 451–454, doi: 10.1109/WKDD.2010.99.

[22] B. Han, H. Ji, and X. Gao, "A 3D wavelet and motion vector based method for wipe transition detection," in *Proc. 7th Int. Conf. Signal Process. (ICSP)*, Sep. 2004, pp. 1207–1210, doi: 10.1109/ICOSP.2004.1441541.

[23] A. Hassanien, M. Elgharib, A. Selim, S.-H. Bae, M. Hefeeda, and W. Matusik, "Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks," 2017, *arXiv:1705.03281*.

[24] B. Han, Y. Hu, G. Wang, W. Wu, and T. Yoshigahara, "Enhanced sports video shot boundary detection based on middle level features and a unified model," *IEEE Trans. Consum. Electron.*, vol. 53, no. 3, pp. 1168–1176, Aug. 2007, doi: 10.1109/TCE.2007.4341601.

[25] Z. Lu, L. Wu, M. Jian, S. Zhang, D. Wang, and X. Wang, "Shot boundary detection with key motion estimation and appearance differentiation," in *Proc. IEEE Int. Conf. Signal, Inf. Data Process. (ICSIDP)*, Dec. 2019, pp. 1–7, doi: 10.1109/ICSIDP47821.2019.9173023.

[26] A. M. Alattar, "Wipe scene change detector for use with video compression algorithms and MPEG-7," *IEEE Trans. Consum. Electron.*, vol. 44, no. 1, pp. 43–51, Feb. 1998, doi: 10.1109/30.663729.

[27] W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull, "Wipe scene change detection in video sequences," in *Proc. Int. Conf. Image Process.*, Oct. 1999, pp. 294–298, doi: 10.1109/ICIP.1999.817120.

[28] J. Nam and A. H. Tewfik, "Detection of gradual transitions in video sequences using B-spline interpolation," *IEEE Trans. Multimedia*, vol. 7, no. 4, pp. 667–679, Aug. 2005, doi: 10.1109/TMM.2005.843362.

[29] M. A. Refaey, K. M. Elsayed, S. M. Hanafy, and L. S. Davis, "Concurrent transition and shot detection in football videos using fuzzy logic," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 4341–4344, doi: 10.1109/ICIP.2009.5413648.

[30] M. Sugano, Y. Nakajima, H. Yanagihara, and A. Yoneyama, "A fast scene change detection on MPEG coding parameter domain," in *Proc. Int. Conf. Image Process. (ICIP)*, Oct. 1998, pp. 888–892, doi: 10.1109/ICIP.1998.723663.

[31] S. Mackowiak and M. Relewicz, "Wipe transition detection based on motion activity and dominant colours descriptors," in *Proc. ISPA 4th Int. Symp. Image Signal Process. Anal.*, Sep. 2005, pp. 480–483, doi: 10.1109/ISPA.2005.195459.

[32] M. D. Bobade, S. Chavan, and S. G. Akojwar, "Development of quick algorithm for wipe transition," *Int. J. Appl. Innov. Eng. Manage. (IJAIEM)*, vol. 5, no. 7, pp. 92–99, Jul. 2016. [Online]. Available: https://www.ijaiem.org/pabstract.php?vol=Volume5Issue&pid=IJAIEM-2016-07-20-17

[33] M. S. Drew, Z.-N. Li, and X. Zhong, "Video dissolve and wipe detection via spatio-temporal images of chromatic histogram differences," in *Proc. Int. Conf. Image Process.*, Sep. 2000, pp. 929–932, doi: 10.1109/ICIP.2000.899609.

[34] S. Chavan, M. Narayana, and L. K. Rao, "Detection and classification of wipe transitions in sport videos in presence of object motion," in *Proc. SPC Int. J. Eng. Technol.*, 2018, pp. 536–539, [Online]. Available: https://www.sciencepubco.com/index.php/ijet/article/download/9111/4427

[35] K. Iwamoto and K. Hirata, "Detection of wipes and digital video effects based on a pattern-independent model of image boundary line characteristics," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2007, pp. VI-297–VI-300, doi: 10.1109/ICIP.2007.4379580.

[36] S. S. Thomas, S. Gupta, and K. S. Venkatesh, "An energy minimization approach for automatic video shot and scene boundary detection," in *Proc. 10th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Aug. 2014, pp. 297–300, doi: 10.1109/IIH-MSP.2014.80.

[37] Y. Gao, Y. Lai, and Y. Liu, "Fast video shot boundary detection based on visual perception," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–4, doi: 10.1109/ICCE.2019.8662083.

[38] J. Xu, L. Song, and R. Xie, "Shot boundary detection using convolutional neural networks," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2016, pp. 1–4, doi: 10.1109/VCIP.2016.7805554.

[39] M. Gygli, "Ridiculously fast shot boundary detection with fully convolutional neural networks," in *Proc. Int. Conf. Content-Based Multimedia Indexing (CBMI)*, Sep. 2018, pp. 1–4, doi: 10.1109/CBMI.2018.8516556.

[40] S. Bhattacharya, A. Prakash, and R. Puri, "Shot change and stuck pixel detection of digital video assets," *SMPTE Motion Imag. J.*, vol. 127, no. 6, pp. 34–43, Jul. 2018, doi: 10.5594/JMI.2018.2827278.

[41] S. Chen, X. Nie, D. Fan, D. Zhang, V. Bhat, and R. Hamid, "Shot contrastive self-supervised learning for scene boundary detection," 2021, *arXiv:2104.13537*.

[42] S. H. Abdulhussain, S. A. R. Al-Haddad, M. I. Saripan, B. M. Mahmmod, and A. Hussien, "Fast temporal video segmentation based on krawtchouk-tchebichef moments," *IEEE Access*, vol. 8, pp. 72347–72359, 2020, doi: 10.1109/ACCESS.2020.2987870.

[43] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," 2017, *arXiv:1705.07750*.

[44] F. N. Bezerra, "A longest common subsequence approach to detect cut and wipe video transitions," in *Proc. 17th Brazilian Symp. Comput. Graph. Image Process.*, Oct. 2004, pp. 154–160, doi: 10.1109/SIBGRA.2004.1352956.

[45] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 374, no. 2065, Apr. 2016, Art. no. 20150202, doi: 10.1098/rsta.2015.0202.

[46] J. Zhou and X.-P. Zhang, "Video shot boundary detection using independent component analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2005, pp. 541–544.

[47] V. B. Thakar and S. K. Hadia, "An adaptive novel feature based approach for automatic video shot boundary detection," in *Proc. Int. Conf. Intell. Syst. Signal Process. (ISSP)*, Mar. 2013, pp. 145–149, doi: 10.1109/ISSP.2013.6526891.

[48] S. Dhiman, R. Chawla, and S. Gupta, "A novel video shot boundary detection framework employing DCT and pattern matching," *Multimedia Tools. Appl.*, vol. 78, pp. 34707–34723, Sep. 2019, doi: 10.1007/s11042-019-08170-3.

[49] R.-K. Shen, Y.-N. Lin, T. T.-Y. Juang, V. R. L. Shen, and S. Y. Lim, "Automatic detection of video shot boundary in social media using a hybrid approach of HLFPN and keypoint matching," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 1, pp. 210–219, Mar. 2018, doi: 10.1109/TCSS.2017.2780882.

[50] Z.-M. Lu and Y. Shi, "Fast video shot boundary detection based on SVD and pattern matching," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5136–5145, Dec. 2013, doi: 10.1109/TIP.2013.2282081.

[51] L. G. G. Priya and S. Domnic, "Walsh–Hadamard transform kernel-based feature vector for shot boundary detection," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5187–5197, Dec. 2014, doi: 10.1109/TIP.2014.2362652.

[52] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of TRECVid activity," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 411–418, 2010, doi: 10.1016/j.cviu.2009.03.011.

[53] A. Singh, D. M. Thounaojam, and S. Chakraborty, "A novel automatic shot boundary detection algorithm: Robust to illumination and motion effect," *Signal, Image Video Process.*, vol. 14, no. 4, pp. 645–653, Jun. 2020, doi: 10.1007/s11760-019-01593-3.

[54] B. S. Rashmi and H. S. Nagendraswamy, "Video shot boundary detection using block based cumulative approach," *Multimedia Tools Appl.*, vol. 80, no. 1, pp. 641–664, Jan. 2021, doi: 10.1007/s11042-020-09697-6.

[55] A. Benoughidene and F. Titouna, "A novel method for video shot boundary detection using CNN-LSTM approach," *Int. J. Multimedia Inf. Retr.*, vol. 11, no. 4, pp. 653–667, Oct. 2022, doi: 10.1007/s13735-022-00251-8.

[56] A. Sasithradevi and S. M. M. Roomi, "A new pyramidal opponent color-shape model based video shot boundary detection," *J. Vis. Commun. Image Represent.*, vol. 67, Feb. 2020, Art. no. 102754, doi: 10.1016/j.jvcir.2020.102754.

[57] J. T. Jose, S. Rajkumar, M. R. Ghalib, A. Shankar, P. Sharma, and M. R. Khosravi, "Efficient shot boundary detection with multiple visual representations," *Mobile Inf. Syst.*, vol. 2022, pp. 1–14, Oct. 2022, doi: 10.1155/2022/4195905.

[58] H. M. Nandini, H. K. Chethan, and B. S. Rashmi, "An efficient method for video shot transition detection using probability binary weight approach," *Int. J. Comput. Vis. Image Process.*, vol. 11, no. 3, pp. 1–20, Jul. 2021, doi: 10.4018/IJCVIP.2021070101.

[59] S. Chavate and R. Mishra, "Efficient detection of abrupt transitions using statistical methods," *ECS Trans.*, vol. 107, no. 1, p. 6541, 2022, doi: 10.1149/10701.6541ecst.

[60] NIST. *Homepage of TREC Video Retrieval Evaluation*. Accessed: Feb. 2022. [Online] Available: https://www-nlpir.nist.gov/projects/trecvid/

DIPANITA CHAKRABORTY received the B.Sc. degree in physics from West Bengal State University, India, in 2017, and the M.Eng. degree in electrical and information engineering from the King Mongkut's University of Technology Thonburi, Thailand, in 2020, where she is currently pursuing the Ph.D. degree in electrical and information engineering technology. Her research interests include video and image processing, content-based video retrieval, semantic event understanding, computer vision, machine learning, data analytics in multimedia, and human–computer interaction.

WERAPON CHIRACHARIT (Member, IEEE) received the B.Eng. degree in electronics and telecommunication engineering, the M.Eng. degree in electrical engineering, and the Ph.D. degree in electrical and computer engineering from the King Mongkut's University of Technology Thonburi (KMUTT), Thailand, in 1999, 2001, and 2007, respectively. He is currently an Assistant Professor with the Department of Electronic and Telecommunication Engineering, Faculty of Engineering, KMUTT. His research interests include digital image processing and computer vision.

KOSIN CHAMNONGTHAI (Senior Member, IEEE) received the B.Eng. degree in applied electronics from the University of Electro-Communications, in 1985, the M.Eng. degree in electrical engineering from the Nippon Institute of Technology, in 1987, and the Ph.D. degree in electrical engineering from Keio University, in 1991. He is currently a Professor with the Electronic and Telecommunication Engineering Department, Faculty of Engineering, King Mongkut's University of Technology Thonburi, and a Vice President-Conference of the APSIPA Association, from 2020 to 2023. His research interests include computer vision, image processing, robot vision, signal processing, and pattern recognition. He is a member of IEICE, TESA, ECTI, AIAT, APSIPA, TRS, and EEAAT. He has served as the Chairperson of the IEEE COMSOC Thailand, from 2004 to 2007, and the President of the ECTI Association, from 2018 to 2019. He has served as an Editor for *ECTI E-Magazine*, from 2011 to 2015, and an Associate Editor for *ECTI-EEC Transactions*, from 2003 to 2010, and *ECTI-CIT* Transactions, from 2011 to 2016.

THEEKAPUN CHAROENPONG received the B.Eng. degree in electronics engineering from the King Mongkut's Institute of Technology Ladkrabang, Thailand, in 2001, the M.Eng. degree in electronics and information engineering from the King Mongkut's University of Technology Thonburi, Thailand, in 2005, and the D.Eng. degree from the Graduate School of Engineering, University of Fukui, Japan, in 2008. He is currently an Associate Professor with the Department of Biomedical Engineering, Faculty of Engineering, Srinakharinwirot University, Thailand. His research interests include feature selection, computer vision, and image processing.