INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Hybridized Neural Network Based Approaches Used for Video Shot Boundary Detection

Shrikant Chavate[1]*, Ravi Mishra[2], S K Singh[3], Dinesh Sharma[4]

1 Ph.D. Scholar, Department of Electronics and Telecommunication Engineering, G H Raisoni University, Amravati, India
2 Associate Professor, Department of Electronics and Telecommunication Engineering, G H Raisoni Institute of Engineering and Technology, Nagpur, India
3 Professor, School of Digital Technology, Atlas Skill Tech University, Mumbai, India
4 Assistant Professor, Department of Electronics and Telecommunication Engineering, Chandigarh College of Engineering and Technology, Chandigarh, India

OPEN ACCESS

Received: 07-07-2023
Accepted: 29-07-2023
Published: 08-09-2023

Citation: Chavate S, Mishra R, Singh SK, Sharma D (2023) Hybridized Neural Network Based Approaches Used for Video Shot Boundary Detection . Indian Journal of Science and Technology 16(33): 2670-2680. https://doi.org/10.17485/IJST/v16i33.1671

* **Corresponding author**.

spchavate@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: The objective of this work is to detect video shot boundary which is an important phase in the domain of video processing. The applications such as indexing, retrieval and browsing of video are majorly depends on accurate detection of shot boundary so the detection is highly necessary in the area of video processing. **Methods:** Many researchers have contributed to the field of Shot Boundary Detection (SBD) to improve video processing throughout the previous decade. Some of the techniques have detected abrupt as well as gradual transitions with remarkable performance metrics values, but they compromised with the execution time and operating speed. Some of the techniques proposed and implemented the simpler approaches, but failed to detect the gradual transitions. Additionally, the effects like illuminations and camera motion lead to produce false results. Thus, taking these aspects into consideration, the novel approach for SBD has to be implemented. **Findings:** In this paper, fusion of transform and filtering at an early stage has been applied for reducing the false hits that occurs due to illumination noise. The fusion of transform with appropriate classifier enhances the detection performance. Another deep learning based technique with Rider bald eagle search optimization given the solution for the SBD. **Novelty and applications:** The proposed two approaches give the detection of abrupt and gradual transitions. It has been noticed that the said approaches found effective against the illuminations. To evaluate the performance it uses precision, recall and F1 score as performance metrics. The experiments are performed on latest TRECVID datasets and few open source video datasets.

Keywords: Shot Boundary Detection; Walsh Hadamard Transform (WHT); Dual Tree Complex Wavelet Transform; Deep Belief Network; Deep Convolutional Neural Network

1 Introduction

A frame is the fundamental unit of video. Multiple frames are used to create shots, which are then continuously streamed as video. In most cases, there is a color or pixel arrangement discrepancy between two frames. The continuous sequence of frames leads to a video. As we discussed, there are two major types of shot transitions, as abrupt and gradual. Because the abrupt transition causes significant changes in successive frames, therefore determining the shot boundary is simple. The changes, on the other hand, are sluggish in the gradual transitions over the number of frames. As a result, determining the shot boundary is extremely challenging[1,2]. This section presented the overview of the existing techniques adopted for SBD. There are the methods existed where separate algorithms were implemented for separate detection of Abrupt Transition (AT) and Gradual Transition (GT).

In recent decades, the fast expansion of technical improvements in the multimedia stream has increased data generation and their utilization in internet. Among the most popular data types on the Internet is videos. YouTube is a significant video resource for all of us to develop ourselves. The survey says, every minute, 300 hours of videos are posted to YouTube. As the demand for information from cyberspace expands in different areas such as education, sports, entertainment, scientific research, and so on, video sharing on YouTube and other websites has become a large business model[3]. Addressing a video clip's location is crucial for retrieving the relevant video clip. Shot boundary detection is used to address the required video clip. In the last two decades, many researchers have made significant contributions to the topic of SBD using a variety of methods. As a result, the synthesis of all of these studies will make it easier for researchers, particularly young researchers, to perceive techniques. As a result, this paper also includes the contribution made for detection of shot boundary by considering the major challenges in this filed. Firstly, let's talk about the core principles of videos and SBD in the parts ahead. Videos are primarily made up of several shots, each of which is made up of many frames. To construct a shot and consequently a video, all of the frames are joined in a sequential way. A single shot is described as a collection of connected, consecutive photos taken by a camera to capture continuous movement in time and space. In the video, the shot transition occurs in two ways as abrupt and gradual. The quick and extreme changes of frames in a video transition are known as abrupt transitions. With contrary, in a Gradual transition, frame changes occur in a step-by-step manner with flexible time and pace to meet the needs of video makers. The gradual transition may be of dissolve, wipe, fade-in and fade-out[4].

Lifang Wu et al. mentioned the methods for detection of gradual and abrupt shot transitions separately. They implemented the hybrid of color histogram with deep features to detect the cuts in a video, and gradual transitions were detected using 3D convolution method. Ravi Mishra et al[3] uses the fusion of WHT and DTCWT, this combination was used for extraction of valuable features from the frames. Author also added the preprocessing techniques for removal of noise at the early stage of SBD. This claimed to locate the transition correctly under illumination effects. Saptarshi Chakraborty et al[4] proposed employing an adaptive strategy with adaptive threshold to extract the probable transitions across the films, taking into account the effects of variation in contrast structure.

B. S. Rashmi et al[5] uses the fusion of local and global features, this method derived a block-based MCSH from each edge gradient fuzzified frame. To detect AT and GT in the video, the RSD statistical metric was applied to the resulting MCSH. The implementation of feature fusion and clustering algorithms was done by Feng-Feng Duan et al[6]. In this, the SURF and fingerprint features from both the compressed and uncompressed domains were retrieved. They eventually underwent clustering after making a fuse. Finally, the process of determining the boundary of a video shot was carried out. Hrishikesh Bhaumik et al[7] used another fusion method of features. The goal is to create a strategy that can build an improved distance feature from existing distance measurements by combining different distance measures. Here, variety of distance characteristics added and merged together. The resulting feature was not only more robust, but also resistant to ineffective features.

Shekhar Dhiman et al[8] suggested an SBD approach where it contains three phases as CSS, AT and GT detection. To speed up the SBD, this work used a pixel-based approach with candidate segment selection. This technique used DCT for detection and Image Histogram and Pattern Matching for Gradual Transition detection. Ning Su et al[9] proposed and implemented unsupervised clustering based real time SBD method. This claimed to have greater efficiency, accuracy and faster results. Tejaswini Kar et al[10] presented a robust technique for detecting abrupt shot boundaries when the frames are partially or completely influenced by illumination, fire, and flicker, as well as high motion linked with an object or camera. Furthermore, they created a model for automatically generating a threshold that made the claim that it took into account the feature vector's statistics, which quadrate with changes in the video's contents. Rong-Kuan Shen et al[11] used the combination of HLFPN with keypoint matching for the SBD work. The HLFPN model combined with histogram difference techniques, and further SURF features were extracted. It detects accurate transition under effects of lighting and flashes. This claimed to have higher precision value. According to the current literature, several studies for the task of Video SBD have been reported. The issues like illumination effects and object/camera motion serve as a primary source of inspiration for contributing research in this field[12]. These are the challenges that encouraged to contribute the research in video shot boundary detection field. This paper

gives the implementation of methodologies mentioned in section 2 and presented the comparative analysis of them.

2 Methodology

As from the state-of-the-art, many researchers presented several techniques to correctly detect the abrupt and gradual transitions. Considering challenges in correct detection of shot boundary under illumination effects and accurate identification of gradual transitions, we proposed and implemented the two different approaches that is, first is using fusion of DTCWT and WHT with DBN and SSDOA and another is using DCNN with RBESO. We used the TRECVID datasets of year 2007, 2016, 2017, 2018, 2019 and some open source videos for performing the experiments.

2.1 Detection of Shot boundary with hybrid of DTCWT and WHT with DBN and SSDOA

The below Figure 1 shows the steps and flow of the SBD approach using combination of DTCWT and WHT with DBN and SSDOA. Below discussion explains the steps involved in this methodology in depth.

a. Pre-processing

This method used the Fast Averaging Peer Group (FAPG) filters that eliminates the impulsive noise at initial stage and enhances the contrast. Also it does well restoration of frames by conserving the edge and other little characteristics[13,14]. The mechanism of these filters based on the membership of centre pixel keeping consideration of peer group size which can be calculated using local neighborhood. Then it does the inspection and replacement of the pixel if it found corrupted. Membership degree has to be formed for centre pixel and local neighborhood window take part for the inspection of a pixel. Whereas, the Weighted Average Filters (WAF) replaces the corrupted pixel or outlier pixel which come under the pixel replacement part.

b. Computation of Color distance using HSV histogram

The proposed method implements the HSV color histogram for distance computation among the frames. This is the simple approach to get the difference values among the frames for locating the abrupt transition in a video. The HSV histogram has a number of advantages over the RGB color space, including being more intuitive. The detection of gradual transitions may not be possible using this way as the gradual transition contains the special editing effects. We used the DTCWT and WHT in a separate manner and compared their results in detection of both the transitions.

c. Dual Tree Complex Wavelet Transform

This uses two real DWTs as component of real and imaginary part of the transform. The separate filter banks are assigned to get the analysis and synthesis of DTCWT and also its inverse.

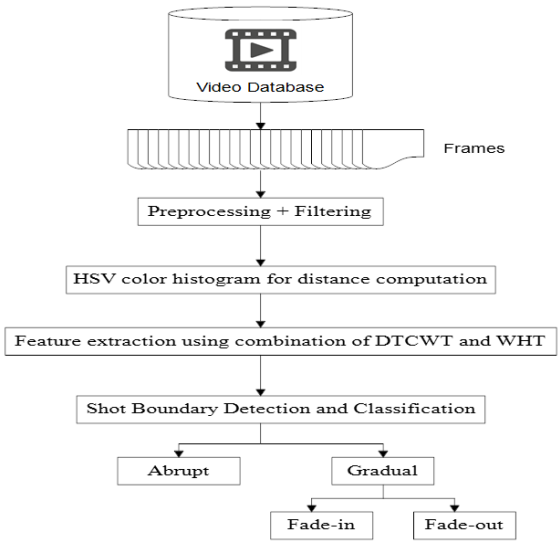

Fig 1. Block diagram of proposed system for SBD

In order to approximate the analytic information, the combination of two filter sets have to be made. These sets are designed using two real wavelet transforms as $\psi h(t)$ and $\psi g(t)$. From this set of filters, each of them meets the PR requirements and the filters are chosen such that they provide the complex wavelet that to be approximately analytic as, $\psi(t) := \psi h(t) + j\psi g(t)$

For the upper FBs, h0(n) assigns for the low pass filters and h1(n) assigns for high pass filters; similarly, g0(n) assigns for low pass filters and g1(n) assigns for high pass filters. They are built in such a way that g(t) approximates the Hilbert transform of ψh(t) [denoted ψg(t) $\approx$ H{ψh(t)}]. Directional selectivity and shift invariance are the major benefits of DTCWT [13,14].

d. Walsh Hadamard Transform

Due to its simplicity and other significant qualities, the WHT is considered to be appealing in terms of implementation [13] [15]. WHT is non-sinusoidal, simple, transforms quickly, and is orthogonal. The WHT matrix can be used to create the WHT. Because the size of the matrix is usually a power of two, it exists for N>2. The WHTM for order N=4 is,

$$D = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \tag{1}$$

This uses WHT kernels for all WHTM basis vectors of order 4, demonstrating that the kernels may be represented in vector space as $V = (v_1, v_2, \ldots, v_{16}\}$. To extract features such as colour, edge, and texture, the average color component will be employed, which will be projected on basis vectors. The various image frequencies were also considered, with the higher frequencies corresponding to edge features and the lower frequencies corresponding to image brightness.

Let, $F_m = \left(f_1^m,\ f_2^m, f_3^m, f_4^m\right\}$, where the number of blocks given by 'm'. F_m is the projection values of the block obtained as the inner product of m^{th} block (B_m) and V_j.

In addition, j=1,2,3,4 correspondingly. Each frame in a block must have its colour, edge, and texture attributes computed. Starting with colour feature extraction, the following equation must be used,

$$X = f_1^m \{B_m, v_1\} \tag{2}$$

Following that, egde features must be extracted from each frame block. Lighting effects, camera movements, and other processes are less susceptible to this method. Using the equation, the gradient vector's magnitude will determine the edge strength,

$$Y = \sqrt{\left(f_2^m\right)^2 + \left(f_3^m\right)^2} \tag{3}$$

To make the computation easier, modify the preceding equation as follows:

$$\mathrm{Y} \approx \left(f_2^m\right)^2 + \left(f_3^m\right)^2 \tag{4}$$

The vector is utilised to represent the finite sum in the following feature extraction, which is texture features. The block projection on vector space can be performed as follows:

$$F = f_1^m v_1 + f_2^m v_2 + f_3^m v_3 \tag{5}$$

$$F = \sum_{i=1}^{3} \left(B_m, v_j\right) v_j \tag{6}$$

The texture feature representation can be done in the following way,

$$Z = \left|B_m^2 - F^2\right| \tag{7}$$

2.1.1 Designing a Continuity Signal

We focused on extracting various features that served as inputs for determining similarity/dissimilarity across video frames in the preceding sections. For computation of continuity signal, the extracted feature and the distance metric will be combined. The crucial decision for locating a boundary must be made in this step of the proposed method. The city block distance measure is used to form the continuity function that also includes the color, edge and texture features.

$$\mu(k) = \mathrm{Dm}(h, h+1)X = \sum_{m=1}^{n} \left|X_{m,h} - X_{m,h+1}\right| \tag{8}$$

$$\sigma(k) = \mathrm{Dm}(h, h+1)Y = \sum_{m=1}^{n} \left|Y_{m,h} - Y_{m,h+1}\right| \tag{9}$$

$$\delta(k) = Dm(h, h+1)Z = \sum_{m=1}^{n} \left|Z_{m,h} - Z_{m,h+1}\right| \tag{10}$$

$$\gamma(k) = 10 * \log_{10}\left[\frac{v^2}{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}(o_{ij} - CmF_{ij})^2}\right] \tag{11}$$

Three crucial features are colour, edge and texture, and these values are described in equations 8), 9) and 10). The temporal domain structure determines the motion strength to be approximated, which can be derived from a series of frames. Equation 11) must be used to calculate the measure of similarity/dissimilarity. The resulting continuity values must be normalised between 0 and 1, and these values are then used as an input for shot boundary identification. To obtain the single continuity function, distinct continuity signals (viz,μ,σ,δ,γ)) are fused.

$$\Omega = \omega_1\mu + \omega_2\sigma + \omega_3\delta + \omega_4\gamma$$

Where, ω_1, ω_2, ω_3, ω_4 are weights assigned to the exctracted features.

2.2 Detection of shot boundary with hybrid feature extraction and DCNN

The another approach for SBD uses the DCNN with RBESO algorithm, as

a. Pre-processing :

In this approach Cross-guided bilateral filtering (CGBF) has been used that helped to take away illumination noise from the input frames. It was built up of an input image M, a guided image G, and a filtered output image O. The position of identical pixels in each neighborhood is shown by the guide image G[12]. Whereas, guide picture G has been used in place of the input image. The connection to robust estimation is misplaced as a result of this modification. This connection is reestablished to maintain the photometric weight P and produce G into a third weight T in cost J[12].

$$\sum_{t\in S_m} S_{\text{weight}}\,(||J||)T(G(x) - G(x+1)\phi((O(x) - P(x+1))^2) \tag{13}$$

Let us discuss this methodology along with below Figure 2.

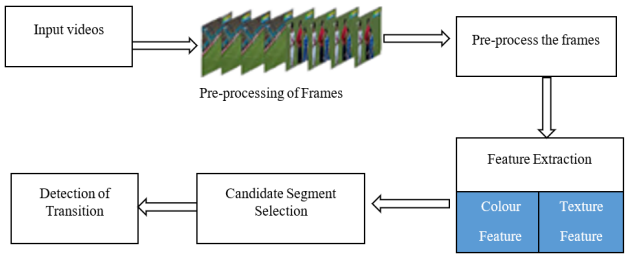

Fig 2. Proposed approach for SBD using DCNN with RBESO

2.2.1 Feature Extraction

Features like color and texture are retrieved in this work. For video shot boundary identification, color features that display color in intensity value format are essential. This is due to the fact that color does not change whether an image is scaled, rotated, or translated.

Extraction of texture feature gives the highest degree of discriminations. Density, coarseness, contrast, and uniformity are frequently found among the texture characteristics.

Here the pre-processed frames are set to extract features like color and texture. The different levels of features have been derived from suggested method that further generates the final feature vector. To extract the color and texture features it uses fuzzy histogram and DTCWT, this further generates the feature vectors.

Let's assume that the fuzzy membership values of all the pixels corresponding to each frame of the color feature vector are used to generate the fuzzy linking histogram. The degree of relationship between each pixel and the region that corresponds to the Cr, Cb, and Y components is assessed. Additionally, the pixel is assigned to the outputs using the fuzzy inference rule.

Each frame yields a color feature vector, which is represented in the following equation:

$$C_{\text{color}} = f(p_1, p_2, p_3, \ldots\ldots\ldots, p_{10}) \tag{14}$$

Where, P_1, P_2, ……., P_{10} shows the number of pixel belongs to each output.

The pre-processed frames are used by the DTCWT to extract the texture feature. For the breakdown of frames, initial and upper stage low pass and high pass filters are utilized. Additionally, the filters get the filter coefficients in order to produce the DTCWT filter coefficients. The LL component, SD, and energy associated to sub bands may all be assessed at each step [12]. The DTCWT kth sub band's SD and energy calculations are expressed mathematically as follows:

For frame fragmentation, the initial and upper LP and HP filters are used and the DTCWT filter coefficients get generated. LL, SD and energy related with sub bands be examined at every step. This is given by following equation as,

$$Fea_k = \frac{1}{a \times b} \sum_{i=1}^{a} \sum_{j=1}^{b} (X_k(i, j)) \tag{15}$$

$$\sigma_k = \frac{1}{a \times b} \left[\sum_{i=1}^{a} \sum_{j=1}^{b} (X_k(i, j) - \delta_k)^2 \right]^{1/2} \tag{16}$$

Where, a x b is kth sub band's size, X_k(i, j) is kth sub band's wavelet decomposition, and δ_k is kth sub band's mean. Sub-band energy and SD combine to create the feature vector. By combining these feature vector components, the texture feature vector is provided by,

$$T_{\text{texture}} = \{F_1, \sigma_1, F_2, \sigma_2 + F\{\delta_1, \delta_2, \delta_3, \delta_4\} + \sigma\{\delta_1, \delta_2, \delta_3, \delta_4\}\} \tag{17}$$

Following the procedure, the final feature vector gets created.

$$\text{FeaVect} = \{C_{\text{color}}, T_{\text{texture}}\} \tag{18}$$

2.2.2 Creation of continuity function and candidate segment selection

Non-transition frames are recognized and removed throughout this procedure. This procedure was carried out keeping a fact into consideration as, there is always a strong correlation between the subsequent frames in a shot that spans a brief temporal segment. Due to the significant degree of resemblance between the segment's initial and last frames, it is considered that this segment is non-boundary. An adaptive threshold is used in this study to determine the suitable complicated video material. The candidate section is chosen using the next several procedures.

Let's calculate the distance between the start and end frames of each section using the equation below.

$$t(n, (n+1)) = \sum_p \sum_q |G(p, q; n) - G(p, q; (n+1))| \tag{19}$$

Where G(p, q) denotes pixel intensity at (p, q) in the frame.

The mathematical expression for the calculation of adaptive threshold is,

$$AT = \delta_1 + c \left[1 + \ln \left(\frac{\delta_g}{\delta_1} \right) \right] \sigma_1 \tag{20}$$

As a result, the non-transition frames are detected and discarded. The possible transition frames are then used to create the continuity matrix, which arranges the frames in a linear sequence with no gaps. Any gaps between the frames are taken into account when determining which frames belong in the next row of the continuity matrix. In order to identify transitions, the gaps between the frames are therefore filled in and the frames are put into the suggested deep learning model in sequential order.

2.2.3 Classification of shot transitions

DCNN is used for the transition detection procedure once the continuity matrix has been created. Making use of EBWOA, the weights of DCNN are updated and this reduces the loss function. The DCNN model is created by deriving the CNN (convolutional neural network) from it.

Figure 3 shows DCNN architecture that consist of pooling layer, convolutional layer (Conv), and fully connected layer (FCL), these three important layers in the DCNN model, are in charge of heavy lifting and processing. Convolution, the first layer, may pull out a number of information from the video frames. Between the convolutional layers and the second layer lies the pooling layer. These poling layers do a subsampling of the feature maps.

Additionally, this layer reduces the spatial size of the representation to manage overfitting and lessen network computation. Third layer that is, the FCL, used for classification that handles all of the connections produced by the preceding levels.

Assuming that C is the input provided to the DCNN model and that the convolutional layer's output is,

$$(C_x^u)_{l,q} = (M_x^u)_l + \sum_{\theta_1=1}^{v_1^{\theta-1}} \sum_{\theta_2=-v_1}^{v_1^u} \sum_{\theta_{3=-v_2}}^{v_1^u} \left(DW_{x,\theta_1}^u\right)_{\theta_3,\theta_2} * \left(C_\theta^{u-1}\right)_{l+\theta_1,q+\theta_3} \tag{21}$$

DW_{x,θ_1}^u : weights of uth Conv.

M_θ^u : bias of uth Conv,

* : convolutional operator that generates local patterns from Conv,

$(C_x^u)_{l,q}$: output from uth Conv centred as (l, q)

θ_1, θ_2, and θ_3 : feature map created from each convolutional filter by scanning the input[12].

Additionally, the computation is made simpler and the negative values are removed using the activation function. The activation function of the (u -1)th layer is the output from the pth convolutional layer and it is provided as,

$$C_x^u = Ac\left(C_x^{u-1}\right) \tag{22}$$

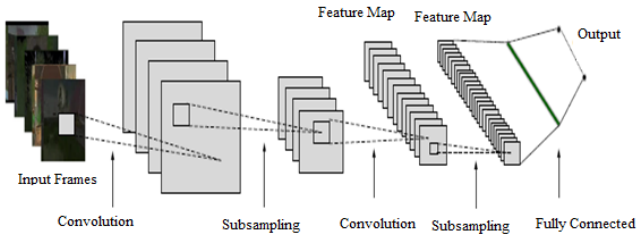

Fig 3. DCNN structure

Conv's output is passed to the pooling layer, which carries out a few predetermined actions. The pooling layer's features are sent to the FCL. The CT lung pictures are then classified once the image has been converted to a vector. The output of FCL is described by equation (16).

$$C_x^u = \eta\left(C_x^u\right) \text{ with } {}_x^u = \sum_{\theta_1=1}^{v_1^{\theta-1}} \sum_{\theta_2=-v_1}^{v_1^u} \sum_{\theta_{3=-v_2}}^{v_1^u} \left(DW_{x,\theta_1}^u\right)_{\theta_3,\theta_2} * \left(C_\theta^{u-1}\right)_{l+\theta_1,q+\theta_3} \tag{23}$$

To get the best result, the RBESO updates the weight of the DCNN model and minimizes the loss function[12]. The method of DCNN training is crucial. When dealing with complicated optimized solutions, existing training techniques like BP (back propagation) have drawbacks including poor convergence speeds and becoming caught in local optima. As a result, BP is unable to improve accuracy. The BP algorithm performance is also announced by the initial solution of learning rates, weights and biases. Even if the learning objective is met, this will have a negative impact on BP's performance. Therefore, to solve these issues, optimization techniques are applied. RBESO is utilized in this study to improve the weight coefficients of the DCNN employed to conduct shot transition categorization.

The bald eagles choose the finest location to pursue their target during this period.

$$W_{\text{new},i} = W_{\text{opt}} + \delta X k\left(W_{\text{mean}} - W_i\right) \tag{24}$$

Where the parameter controlling position changes is represented as, which has a range of [1.5,2], and the random integer with a range of [0,1] is denoted as k.

Bald eagles may select the region based on the information at their removal. The bald eagles randomly select a different search area.

The location is however close to the prior search area. The bald eagle now chooses the search space, which is denoted by W_{opt}, based on the ideal position. The eagles that made use of all the data from the preceding point are referred to as W_{mean}.

3 Result and Discussion

3.1 DTCWT-WHT with DBN-SSDOA approach

MATLAB 2020a is used to implement the proposed algorithm. The suggested system's experimental findings are compared to previous SBD approaches. The findings are analyzed using TRECVID datasets, and the efficiency of the proposed system is evaluated using performance measures.

Performance metrics such as precision, recall rate, and F1 Score are used in this analysis; these three factors will evaluate the algorithm's usefulness and reliability[16,17].

$$\text{Recall} = \frac{T}{T+M} \tag{25}$$

$$\text{Precision} = \frac{T}{T+F} \tag{26}$$

$$F1\ \text{Score} = 2 * \frac{P*R}{P+R} \tag{27}$$

Where, T = transitions correctly located

M = Missed no. of transitions

F = False transitions

Precision(P) gives the measure of relevant frames identified from all frames, Recall rate(R) refers to accurate identification of relevant shots and F1 score(F1) is the weighted average of P and R,[13,14,18,19]

The Figure 4 denotes the identification of AT and Figure 5 a) shows the GT detection and b) shows frames under illumination respectively. The experiments are performed on TRECVID datasets of year 2007 and few videos from recent TRECVID dataset from year 2016 to 2019 and some open source videos.

Table 1 shows the performance metrics for the abrupt and gradual transition detection for different test videos from TRECVID and other open source videos.

Table 1. Performance metrics of shot boundary detection using WHT-DTCWT with DBN-SSDOA

Dataset	Gradual Transition			Abrupt Transition		
	P	R	F1	P	R	F1
TRECVID 2007	91.25	89.65	90.22	99.15	97.14	98.13
TRECVID 2016	88.79	89.59	88.13	98.89	99.54	99.21
TRECVID 2017	87.33	88.26	87.15	99.55	99.12	99.33
TRECVID 2018	87.9	87.9	86.46	99.15	96.36	97.74
TRECVID 2019	87.13	87.13	85.46	99.15	96.12	97.61
Open Source Dataset	89.25	87.14	88.21	99.55	99.13	99.34

3.2 DCNN with RBESO approach

We implemented this system on MATLAB 2020a, following Table 2 shows the values of performance metrics for abrupt and gradual transition detection.

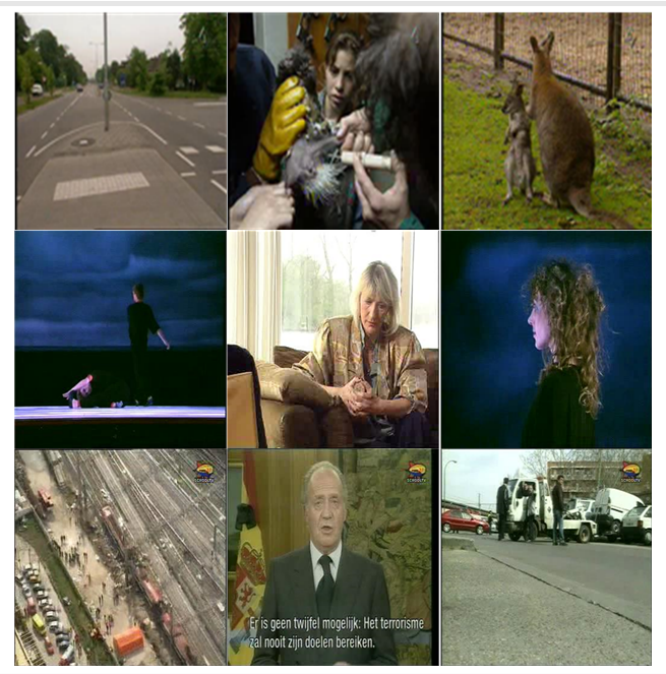

Fig 4. Abrupt Transition detection

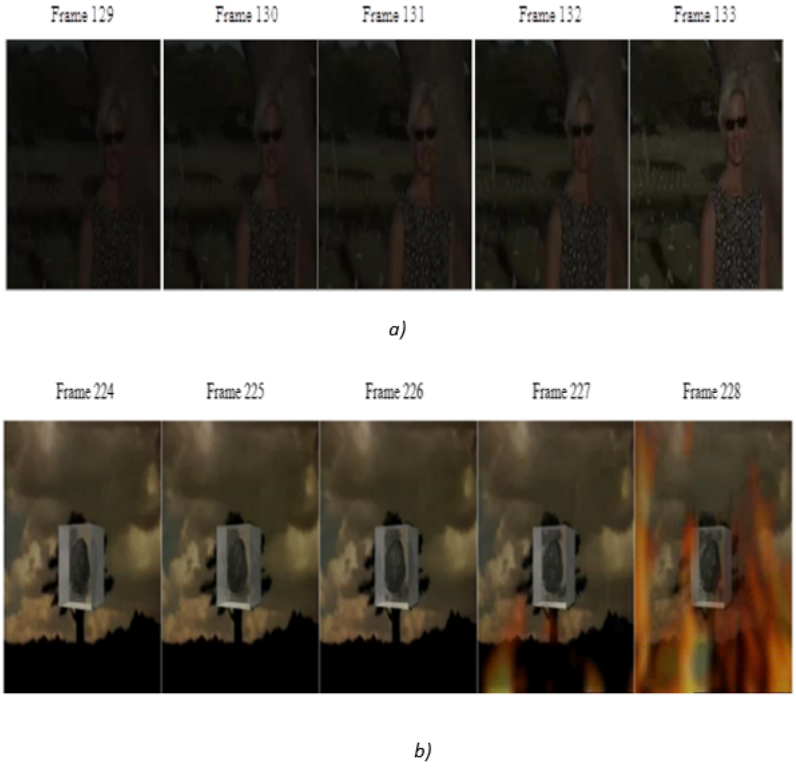

Fig 5. a) Gradual Transition detection (fade-in and fade-out) b) frames under illumination effect

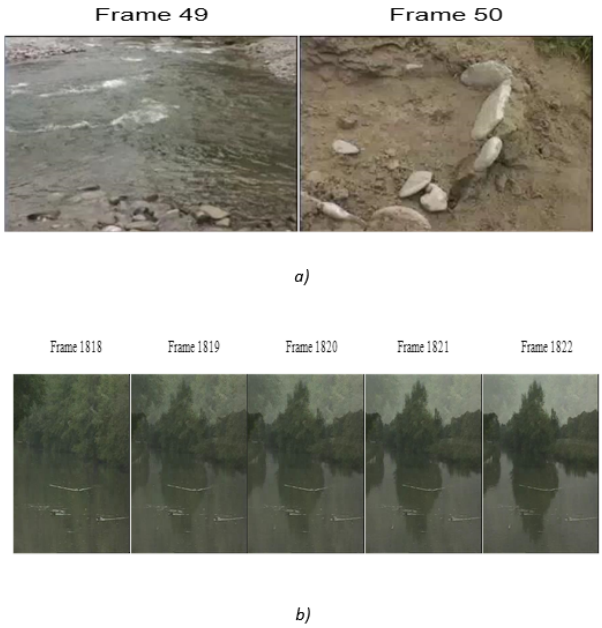

Fig 6. a) Abrupt and b) Gradual transition detection using DCNN With RBESO

Table 2. Performance metrics of shot boundary detection using DCNN with RBESO

Dataset	Gradual Transition			Abrupt Transition		
	P	R	F1	P	R	F1
TRECVID 2007	90.3	88.14	89.21	97.25	96.57	96.91
TRECVID 2016	86.45	88.65	87.54	97.42	98.44	97.93
TRECVID 2017	87.01	87.29	87.15	98.44	98.19	98.31
TRECVID 2018	86.25	86.95	86.60	98.98	95.84	97.38
TRECVID 2019	86	86.5	86.25	99.1	95.99	97.52
Open Source Dataset	85.25	86.59	85.91	99.02	98.74	98.88

Table 3. Comparison of performance metrics of shot boundary detection using DTCWT-WHT with DBN-SSDOA approach and DCNN with RBESO approach

Method	Abrupt Transitions			Gradual Transitions		
	P	R	**F1**	**P**	**R**	**F1**
	%	**%**	**%**	**%**	**%**	**%**
WHT	89	91	90	87	85	86
DTCWT	82	88.3	85.2	79.8	78.9	73.6
DTCWT + WHT with DBN - SSDOA	99.24	97.90	98.56	88.60	88.27	87.60
DCNN with RBESO	86.88	87.35	87.11	98.37	97.30	97.82

4 Conclusion

In this study, a hybridized WHT and DTCWT is proposed to extract useful features and combination of DBN - SSDOA which is used to classify the type of shot transition. Another approach of DCNN with RBESO has been implemented. The experiments are performed on TRECVID and other open source video datasets. From these two approaches, the abrupt and gradual transitions are detected. Table 3 shows the analysis of these two approaches. First approach of WHT-DTCWT with DBN – SSDOA is found to be more accurate than that of DCNN with RBESO approach. The use of FAPG filters in WHT-DTCWT with DBN – SSDOA method provided by the efficient detection results under illumination effects as compared to CGBF filtering action in DCNN with RBESO approach.

Acknowledgement

The authors thank the researchers who contributed their efforts into the field of video shot boundary detection that motivated us to work in this field. Authors also thank to NIST for providing the required video datasets for performing the experiments.

References

1) Hato E, Abdulmunem ME. Fast Algorithm for Video Shot Boundary Detection Using SURF features. *2019 2nd Scientific Conference of Computer Sciences (SCCS)*. 2019;p. 81–86. Available from: https://doi.org/10.1109/SCCS.2019.8852603.
2) Wu L, Zhang S, Jian M, Lu Z, Wang D. Two Stage Shot Boundary Detection via Feature Fusion and Spatial-Temporal Convolutional Neural Networks. *IEEE Access*. 2019;7:77268–77276. Available from: https://doi.org/10.1109/ACCESS.2019.2922038.
3) Mishra R. Video shot boundary detection using hybrid dual tree complex wavelet transform with Walsh Hadamard transform. *Multimedia Tools and Applications*. 2021;80(18):28109–28135. Available from: https://doi.org/10.1007/s11042-021-11052-2.
4) Chakraborty S, Thounaojam DM. SBD-Duo: a dual stage shot boundary detection technique robust to motion and illumination effect. *Multimedia Tools and Applications* . 2021;80(2):3071–3087. Available from: https://doi.org/10.1007/s11042-020-09683-y.
5) Rashmi BS, Nagendraswamy HS. Video shot boundary detection using block based cumulative approach. *Multimedia Tools and Applications*. 2021;80(1):641–664. Available from: https://doi.org/10.1007/s11042-020-09697-6.
6) Duan FFF, Meng F. Video Shot Boundary Detection Based on Feature Fusion and Clustering Technique. *IEEE Access*. 2020;8:214633–214645. Available from: https://doi.org/10.1109/ACCESS.2020.3040861.
7) Bhaumik H, Bhattacharyya S, Chakraborty S. A vague set approach for identifying shot transition in videos using multiple feature amalgamation. *Applied Soft Computing*. 2019;75:633–651. Available from: https://doi.org/10.1016/j.asoc.2018.10.053.
8) Dhiman S, Chawla R, Gupta S. A novel video shot boundary detection framework employing DCT and pattern matching. *Multimedia Tools and Applications*. 2019;78(24):34707–34723. Available from: https://doi.org/10.1007/s11042-019-08170-3.
9) Su N, Zhang J, Zhang Y, Zhang G. Unsupervised Clustering Based Real-time Shot Boundary Detection for Live Broadcasting. *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. 2019;p. 135–140. Available from: https://doi.org/10.1109/ICCC47050.2019.9064356.
10) Kar T, Kanungo P. Motion and illumination defiant cut detection based on Weber features. *IET Image Processing*. 2018;12(10):1903–1912. Available from: https://doi.org/10.1049/iet-ipr.2017.1237.
11) Shen RKK, Lin YNN, Juang TTYTY, Shen VRL, Lim SY. Automatic Detection of Video Shot Boundary in Social Media Using a Hybrid Approach of HLFPN and Keypoint Matching. *IEEE Transactions on Computational Social Systems*. 2018;5(1):210–219. Available from: https://doi.org/10.1109/TCSS.2017.2780882.
12) Mishra R. Hybrid feature extraction and optimized deep convolutional neural network based video shot boundary detection. *Concurrency and Computation: Practice and Experience*. 2022;34(25):1–18. Available from: https://doi.org/10.1002/cpe.7256.
13) Chavate S, Mishra R. An Efficient Approach for Shot Boundary Detection in Presence of Illumination Effects using Fusion of Transforms. *International Journal of Engineering Trends and Technology*. 2022;70(4):418–432. Available from: https://ijettjournal.org/assets/Volume-70/Issue-4/IJETT-V70I4P236.pdf.
14) M RS, A K, S A. The Detection of Video Shot Transitions Based on Primary Segments Using the Adaptive Threshold of Colour-Based Histogram Differences and Candidate Segments Using the SURF Feature Descriptor. 2022. Available from: https://doi.org/10.3390/sym14102041.
15) Ashok PKP, A JD. Deep Learning Approach for Video Shot Boundary Detection. *International Journal of Emerging Technologies and Innovative Research* . 2020;7(6):411–417. Available from: https://www.jetir.org/view?paper=JETIR2006058.
16) Chavate S, Mishra R, Yadav P. A Comparative Analysis of Video Shot Boundary Detection using Different Approaches. *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*. 2021;p. 524–530. Available from: https://ieeexplore.ieee.org/document/9676246.
17) Chakraborty D, Chiracharit W, Chamnongthai K, Charoenpong T. Wipe Scene Change Detection in Object-Camera Motion Based on Linear Regression and an Inflated Spatial-Motion Neural Network. *IEEE Access*. 2023;11:33080–33099. Available from: https://ieeexplore.ieee.org/document/10086505.
18) Idan ZN, Abdulhussain SH, Mahmmod BM, Al-Utaibi KA, Al-Hadad SAR, Sait SM. Fast Shot Boundary Detection Based on Separable Moments and Support Vector Machine. *IEEE Access*. 2021;9:106412–106427. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9496657.
19) Chavate S, Mishra R. Efficient Detection of Abrupt Transitions Using Statistical Methods. *ECS Transactions*. 2022;107(1):6541–6552. Available from: https://iopscience.iop.org/article/10.1149/10701.6541ecst/pdf.