

A Video Summarization Technique using Multi-Feature DWHT and GMM for CBVR System

Original Scientific Paper

Dappu Asha*

Jawaharlal Nehru Technological University Hyderabad,
Department of Electronics and Communication Engineering
Telangana, India
ashamanickrao@gmail.com

Y. Madhavee Latha

Malla Reddy Engineering College for Women, affiliated to JNT University,
Department of Electronics and Communication Engineering
Telangana, India
madhaveelatha2009@gmail.com

*Corresponding author

Abstract – The increasing utilization of multimedia data and digital information in present times presents a vast scope for research in content-based retrieval systems. An improved CBVR System is proposed to extract video streams effectively using DWHT Multi-features and GMM. Our CVBR method performs VSBD for identifying Video shots by computing DWHT on video frames for multi-feature extraction, and then key frames are identified. A summarized frame is developed using the VS algorithm based on GMM on the UCF Dataset. Later, a procedure is applied for the input query video stream, and correlation coefficients are calculated between the query and the database multi-feature vectors, giving us similarity measures. Lastly, our experimental results validate the efficiency of our proposed CBVR System, achieving an average precision of 0.821 and a loss of 0.179, outperforming existing CBVR systems using DCT and optimized perceptual VS, which have precision values of 0.6475 and 0.71, respectively, along with losses of 0.3525 and 0.29.

Keywords: Content-based Video Retrieval (CBVR), Discrete Walsh-Hadamard Transform (DWHT), Video Shot Boundary Detection (VSBD), Video Summarization (VS), Gaussian Mixture Model (GMM).

Received: June 30, 2025; Received in revised form: September 19, 2025; Accepted: September 22, 2025

1. INTRODUCTION

Advancements and improvements in technology have made a large amount of information available on the web [1]. Due to this, the demand for automatic tools for browsing, retrieving, intelligent surveillance, and ranking of information has gained importance [2]. Since video is a significant source of information available on the web, it occupies a large memory size and requires machinery for analyzing [3]. Content-based retrieval is essential because text-based retrieval is limited by human errors and manipulations [4, 5]. The first two levels in the CBVR framework are Video Shot Boundary Detection (VSBD) and Video Summarization (VS). A video shot is an assembly of similar frames formed by still or moving camera images [6]. The detection of the transition from one shot to the next shot is called shot detection. The shot transitions are

categorised into CT (Cut Transition) and GT (Gradual Transition) [7]. CT is an abrupt change between one video shot and the succeeding video shot, whereas GT is a slow change that occurs in the video stream, and it continues for many video frames that arises due to video editing. Several kinds of video editing effects exist, such as dissolve, fadeout, fade-in, etc [7]. The method of mechanically segmenting a video stream into video shots or scenes is termed VSBD [7]. VS is a crucial step in the CBVR system, reducing the video's dimensionality to a single frame.

Our proposed CBVR System comprises online and offline processes. The offline process is carried out on database videos, and the online process is computed on the query video. There are four steps in the CBVR process. It starts with VSBD to identify shots and key frames, then VS is applied to summarize the key frames. Next, from the summarized frame, DWHT-based multi-

features are extracted, and lastly, similarity is measured to retrieve similar videos.

The framework of our paper is ordered as follows: Section 2 bounces on a literature review of the current CBVR techniques. In Section 3, our proposed CBVR system is illustrated. In Section 4, experimental results are presented. Finally, Section 5 discusses the conclusion of our work.

2. LITERATURE REVIEW

Numerous Automatic Video Retrieval systems have been proposed in the past few years. From the literature reviews, different retrieval methods like text-based [8], content-based, query image-based [9], and sketch-based [10] were developed. The Literature survey is presented in Table 1, providing a brief overview of features, datasets, results, advantages, and limitations.

Table 1. Literature Survey on CBVR System

Author /Year of publication	Search Type	Features	Methods/ Techniques	Database and Results	Advantages	Limitations
Palanivelu et al. (2024) [11]	Query input	Pertinent visual features using ResNet50	CNN	TREC02, TREC10, YTAD09, and IDV01 Accuracy of 58.33, 91.67, 92.08, and 23.08, respectively	CNN can automatically learn complex features from raw video data	Poor performance on certain types of complex datasets, the computational cost is high and requires more labelled data
Farhan et al. (2021) [12]	Query by Example	Color features	Discrete Cosine Transform (DCT)	Real World 8 Classes each contain videos Precision of 0.6475	Effective and automatic feature extraction from video content	Did not consider semantic features and evaluated on a small database
Sathiyaprasad et al. (2020) [13]	Query input	I-GLCM (Improved Gray Level Co-Occurrence Matrix)	RPCNN (Region-based Pre-Convolved Neural Network)	MNIST (4000 images), KAGGLE Precision of 0.9067	Combines I-GLCM and R-PCNN, which aims to optimize accuracy	Using local identifiers and descriptors increases the computational cost
Dyana et al. (2010) [14]	Query by Example	MST-CSS (Multi-Spectro-Temporal Curvature Scale Space)	Multiscale and multispectral Filters	480 real-world video Shots 50 classes each contain 20 videos Precision of 0.71	Combining shape contour and motion trajectory through multiscale and multispectral processing	Operates only on static backgrounds
Shivanand et al. (2019) [15]	Query Input	Semantics contents	ROI and ACF Detector	Own 70 video Dataset captured from a mobile phone No evaluation metric considered	Focuses on techniques for detecting the Region of Interest (ROI)	Self-collected dataset, ROI detection is primarily focused on the signboards only
Mallick et al. (2019) [16]	Query Input	Motion Vector	Spatial Pyramid matching (Haar Transformation -4 Level)	UCF Dataset VCD Dataset Precision of 0.8862	Utilizes motion vector-based key frame extraction as a video summarization technique to recapitulate video content	Fails in guaranteeing its efficiency for conspicuous motion
Thomas et al. (2019) [17]	Query Input	Single frame-based approach	Human visual system, optimization	Standard video data sets UCF, MED, CCV, BBC, OVP Precision of 0.71	Uses a single summarized frame for indexing instead of multi-frame indexing, the method reduces computational complexity and memory demands for video databases	Incapability lies in its sensitivity to segmentation, background extraction errors, and its inability to effectively summarize crowded foreground activities
Asha et al. (2018) [18]	Query Input	Color Distributions Texture & Motion Binary Patterns	LBP and SAD	40 videos from Google 4 classes each contain 10 videos Precision of 0.80	A multiple-feature approach improves performance	High Computational cost leading to more execution time

Reference [11] highlights the significant advancements in Content-Based Video Retrieval through the application of deep learning, particularly using models like Inception ResNet. It enables the extraction of intricate, high-level features from video frames, leading to more precise and efficient video search and retrieval capabilities. Reference [12] used a transform-based

CBVR system grounded on single DCT color features; the study achieved an average result of 0.6475 by implementing DCT on a database comprising 100 videos across various categories, with 5 videos in each category. These findings underscore the efficacy of using DCT in video retrieval processes. while [13] illustrated region-based Improved-GLCM on RPCNN using image

query, this approach aims to address the computational constraints and accuracy limitations of existing CBVR systems. Reference [14] discusses a unified approach using MST-CSS attributes generated by multi-spectral filters to efficiently represent video objects. This approach integrates spatial and temporal information within a unified framework. Reference [15] explains the semantic features and ROI, which require additional user inputs. Reference [16] is based on motion vector features and spatial pyramid matching; it narrows the search space. This approach is motivated by its ability to overcome limitations of Bag-of-Features methods by considering feature spatial layout. It involves partitioning an image or key frame into finer sub-regions and computing local features for each sub-region. [17] explains video summarization using human perception and optimization for redundancy removal. It highlights the growing need for efficient video summarization due to increased video consumption. It identifies shortcomings in existing methods, particularly their inability

to accurately represent video events and adapt to different scene types. The proposed solution focuses on a context-driven, perceptually optimized framework that creates a single summarized frame, promising enhanced retrieval performance and reduced resource demands. [18] signifies the use of multiple features for improving retrieval, it highlights that an effective CBVR system requires considering both spatial and temporal features to achieve accurate results, distinguishing it from Content-Based Image Retrieval (CBIR).

3. IMPLEMENTATION OF PROPOSED CBVR SYSTEM

In this section, we enlighten on our CBVR system. VSBD is the first step in the CBVR system where DWHT is applied. The second step is Video Summarization (VS) using a Gaussian Mixture Model (GMM), and the third step is video retrieval using the extracted Multi-Features. The block diagram of the proposed CBVR System is shown in Fig.1.

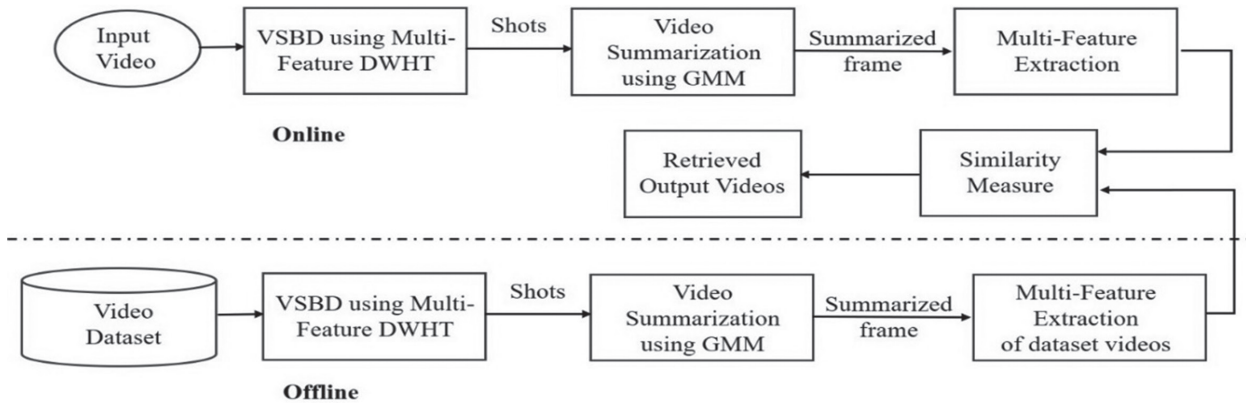


Fig. 1. Proposed CBVR System Block Diagram

3.1. WALSH-HADAMARD TRANSFORM

Discrete Walsh Hadamard Transform (DWHT) [19] is immensely used in numerous applications of image and video processing because of its robustness, energy compaction, fast computation, less memory storage space, and flexibility. DWHT is defined below:

Let $f_i(x, y)$ be the i^{th} frame of size $M \times N$. The forward discrete Walsh-Hadamard $X_i(u, v)$ can be expressed as in (1)

$$X(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f_i(x, y) g(x, y, u, v) \quad (1)$$

for x, y are spatial coordinates and u, v are coordinates in transform domain
 $x, u = [0, 1, 2, \dots, M-1]$ and $y, v = [0, 1, 2, \dots, N-1]$
 $g(x, y, u, v)$ is forward Mask

The forward kernel of DWHT is defined as in (2)

$$g(x, y, u, v) = \frac{1}{N} (-1)^{\sum_{i=0}^{m-1} [b_i(x)p_i(u) + b_i(y)p_i(v)]} \quad (2)$$

Where $N=2m$ is the size of the transform matrix. The summation in exponent is performed in modulo 2 arithmetic, and $b_i(y)$ is the i^{th} bit in the binary representation of y .

The DWHT matrix of order 8 ($N=8$) is shown in (3)

$$\text{For } N = 8: \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (3)$$

The blending functions of DWHT are characterized in mask vectors as $W = \{w_1, w_2, \dots, w_{64}\}$ aligned from top to bottom and left to right of the DWHT masks for $N=8$, given in Fig. 2. These masks of the DWHT help in extracting multi-feature vectors.

3.2. VSBD TECHNIQUE

The VSBD process is accomplished in four steps: computing DWHT kernels, multi-feature extraction, composing a continuous vector, and identifying the video shot boundary. Various VSBD techniques used previously are presented here. Reference [20] projected a technique for extracting key frames of multi-features, the algorithm leverages deep prior information and

multi-feature fusion to enhance saliency extraction. [7] Proposed content-based VSBD using the Haar transform to accurately detect abrupt and gradual video transitions. [21] combines candidate segment selection with SVD for dimensionality reduction and employs distinct pattern matching techniques. [22] discusses WHT and a procedure-based identification process to distinguish all shot transitions, and [23] discusses cut detection using a histogram where a single feature is considered and the GT is neglected. We observe that transform-based techniques are more accurate, and multi-feature extraction with a procedure-based approach will help in detecting CT and GT. Video is read into the system, and multi-features are extracted by protrusive DWHT kernels on each video frame. The dissimilarity and similarity among succeeding video frames are measured by calculating the correlation between consecutive feature vectors. The procedure-based VSBD algorithm is useful to identify shot transitions. The block diagram of the VSBD Technique is shown in Fig. 3.

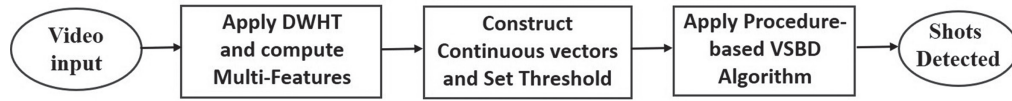


Fig. 3. Block diagram of the VSBD Technique

3.3. MULTI-FEATURE VECTOR EXTRACTION

In the video shot boundary detection, the multi-feature vector extraction phase is very significant. We extract features like motion, color, shape, and texture vectors by applying DWHT blending functions on video streams.

The blending functions or kernels used for color, shape, and texture feature extraction are shown in equation (4). We use kernel w_2 to w_{64} for high-frequency and w_1 for low-frequency demonstrations. The kernels w_1 , w_{33} , and w_{37} are shown below in equation (4):

$$\begin{aligned}
 W_1 = w_1 &= \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\
 W_2 = w_{33} &= \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \\
 W_3 = w_{37} &= \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}
 \end{aligned} \quad (4)$$

The shape feature vectors are computed by projecting the w_{33} and w_{37} masks. The texture feature vectors are computed by projecting w_1 , w_{33} , and w_{37} masks.

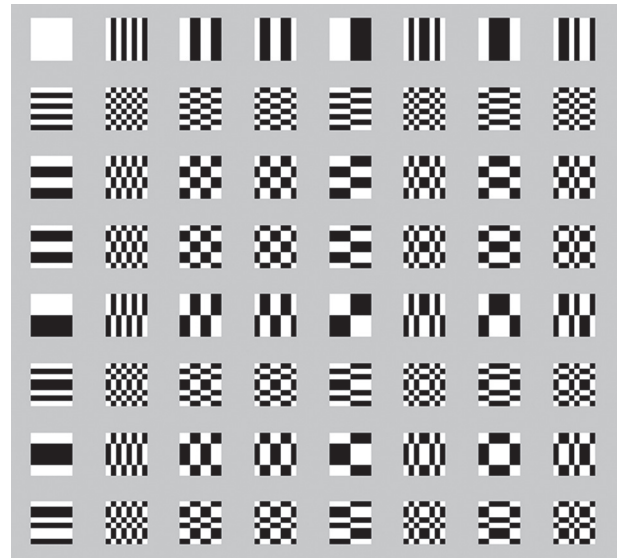


Fig. 2. DWHT (Discrete Walsh-Hadamard Transform) Kernels

The motion feature vectors are calculated by following the steps

1. Project W_4 =DWHT kernel for $N=8$ as in (3) on succeeding frames.
2. Calculate Motion Vector (MV) using the SAD (Sum of Absolute Difference) method.
3. Extract motion features by subtracting MV from the projected W_4 successive frames.
4. Calculate the correlation between succeeding motion strength frames.

3.4. FEATURE EXTRACTION PROCEDURE

Let $X_m = \{x_{m1}, x_{m2}, x_{m3}, x_{m4}\}$, X_m as in (5), signify the anticipated values of blocks by designing the inner product of $K(K_m)$ where $m = [1, 2, \dots \text{No. of blocks}]$ and $W_j, j=1, 2, 3, 4$, respectively.

X_m are computed using below equation:

$$X_m = \{x_1^m = \langle K_m, W_1 \rangle, x_2^m = \langle K_m, W_2 \rangle, x_3^m = \langle K_m, W_3 \rangle, x_4^m = \langle K_m, W_4 \rangle\} \quad (5)$$

Where $\langle K_m, W_1 \rangle = \sum_{i=1}^p K_{mi} * W_{ji}$.

Here, W_{ji} is the i^{th} value of W_j blending vector, K_{mi} is the i^{th} value of K_m , and p is the number of pixels in each block K_m .

- i. The Color Feature Vector (C_m) of the consequent block is obtained as in (6):

$$C_m = x_1^m = \langle K_m, W_1 \rangle \quad (6)$$

- ii. The Shape Feature Vector (E_m) of the consequent block is obtained as in (7):

$$E_m = \sqrt{(x_2^m)^2 + (x_3^m)^2} \quad (7)$$

iii. The Texture Feature Vector (T_m) of the consequent block is obtained as in (8) and (9):

$$T_m = |K_m^2 - Z^2| \quad (8)$$

$$Z = x_1^m W_1 + x_2^m W_2 + x_3^m W_3$$

$$K_m = \sum_{i=1}^3 \langle K_m, W_i \rangle W_i \quad (9)$$

iv. The Motion Feature Vector (M_m) of the consequent block is obtained as in (10):

$$M_m = |x_4^m - ME| \quad (10)$$

3.5. CONSTRUCTION OF CONTINUOUS VECTOR

Subsequently, after identifying multi-features, the next phase in the VSBD involves computing a continuous vector to determine the similarity between successive frames, as outlined in the equations below, where P represents the Correlation coefficient. The estimated correlation coefficients between the succeeding frames are calculated among the blocks of the f and $f+1$ frames [21]. Hence, for all feature vectors of color (C), Texture (T), Shape (S), and Motion (M), the equivalent continuous vector equations are given as in (6-10) [23].

After constructing individual features, a continuous vector as in (11-14), we calculate the mean of all individual vectors as in (15). Continuous vector is in the range of [0,1]. On these combined coefficients (μ), a procedure-based VSBD algorithm is applied for recognizing shot transitions.

$$\alpha(f) = P(f, f+1) = \sum_{m=1}^{\text{no of blocks}} \text{corrcoef}(C_{m,f} - C_{m,f+1}) \quad (11)$$

$$\beta(f) = P_S(f, f+1) = \sum_{m=1}^{\text{no of blocks}} \text{corrcoef}(E_{m,f} - E_{m,f+1}) \quad (12)$$

$$\gamma(f) = P_T(f, f+1) = \sum_{m=1}^{\text{no of blocks}} \text{corrcoef}(T_{m,f} - T_{m,f+1}) \quad (13)$$

$$\delta(f) = P(f, f+1) = \sum_{m=1}^{\text{no of blocks}} \text{corrcoef}(M_{m,f} - M_{m,f+1}) \quad (14)$$

$$\mu(f) = \frac{1}{4} \{ \alpha(f) + \beta(f) + \gamma(f) + \delta(f) \} \quad (15)$$

3.6. PROCEDURE FOR THE VSBD ALGORITHM

Our proposed procedure for the VSBD algorithm is grounded on the following guidelines to identify the CT and GT. If the sequential frames are identical, then

the continuous vector will be high and when frames differ, the values will be low. A Threshold value (T_h) is computed to identify the shots, and T_h is calculated by taking the mean of the continuous signal as in (16).

$$Th = \frac{1}{n} \left(\sum_{f=1}^n \mu(f) \right) \text{ where } n = \text{No. of Frames in video.} \quad (16)$$

According to the continuous correlation coefficient values, it is easy to recognize the presence of CT. The continuous correlation values for GT in the video sequence between the successive frames will be lower. An example of a VSBD plot is shown in Fig. 4.

Procedure for finding Valley points:

1. Compute all continuous signal $\mu(f)$ as in (15) where $f=1,2,\dots,n$, $n=\text{No. of frames}$.

Calculate the Threshold value (Th) by using equation (16).

Find the valley points $V(k)$ that are less than the threshold Th .

2. Location of the valley is stored as $Lop(m) = k$, where the valley occurred at the k^{th} frame, $m = m+1$, till finding all valley points, and the procedure ends.

Procedure-based Shot Detection

Step 1: Read $\mu(f)$, $V(k)$, nv : number of valleys, Lop : location of valley point.

Step 2: Set the Threshold (Th).

Step 3: If the values before and after the valley point are greater than Th , then identify the shot transition as CT.

Step 4: Else identify the shot as GT.

Step 5: If the valley point values are in a gradual transition and are less than 0.6, then identify as a fade transition.

Step 6: Compare the valley point values with the previous values and decrement $c2$ until the condition is false, where $c2$ represents the starting point of the fade transition.

Step 7: Compare the valley point values with the upcoming values, incrementing $c1$ until the condition is false. Here, $c1$ represents the end point of the fade transition.

Step 8: If the valley point values in GT exceed 0.6, then identify it as a dissolve transition.

Step 9: Compare the valley point values with the previous values and decrement $c2$ until the condition is false, where $c2$ represents the starting point of the dissolve transition.

Step 10: Compare the valley point values with the upcoming values, incrementing $c1$ until the condition is false. This represents the end point of the dissolve transition.

Step 11: Repeat from step 3.

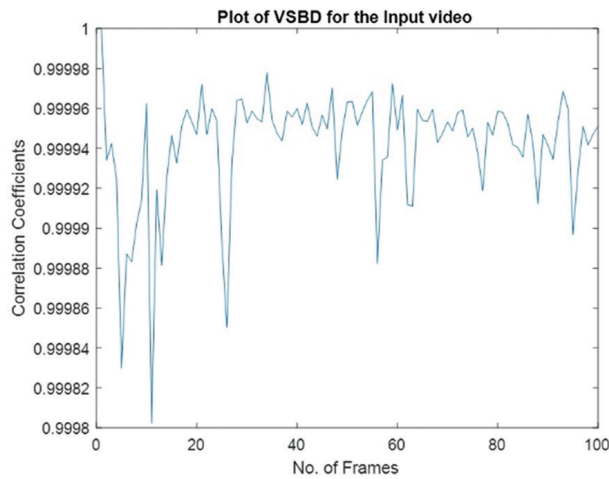


Fig. 4. VSBD plot

3.7. VS USING GMM

VS using the GMM method for detecting foreground [24, 25] is illustrated here. GMM is built for foreground Modelling. It is an extensively common procedure for moving object detection. It executes a soft clustering method to categorize each pixel as foreground or background by assigning a score to each pixel indicating the strength of the pixel [26, 27].

GMM is computed on each pixel using equations (17) and (18) below.

$$P(X_t) = \sum_{i=1}^K w_{i,t} \cdot \Omega(X_t, \mu_{i,t}, \sigma_{i,t}) \quad (17)$$

where X_t : pixel in t^{th} frame

K : the number of components

$w_{i,t}$: weight of the K^{th} component in t^{th} frame

$\mu_{i,t}$: the mean of K^{th} component in t^{th} frame

$\sigma_{i,t}$: the standard deviation of K^{th} component in t^{th} frame

where $\Omega(X_t, \mu_{i,t}, \sigma_{i,t})$ probability density function

$$\Omega(X_t, \mu, \sigma) = \frac{1}{(2\pi)^{1/2} |\sigma|^{1/2}} \exp^{-\frac{1}{2} (X_t - \mu)^T \sigma^{-1} (X_t - \mu)} \quad (18)$$

The Procedure for VS using the GMM Algorithm is given below, and the flowchart is shown in Fig.5.

1. Read frames from video.
2. Extract shots using VSBD using Multi-Feature DWHT.
3. Extract key frames from shots, apply foreground detection using the GMM method, and create a mask.
4. Apply 2D Gaussian and morphological filtering for smoothing and noise removal (the outputs of different stages are shown in Fig.6).
5. Select the frame and stitch objects to get the summarized frame shown in Fig.7.

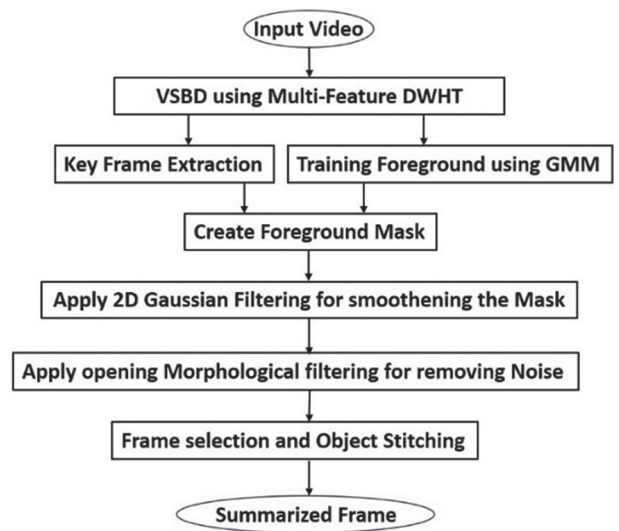


Fig. 5. Flowchart for VS Technique using GMM

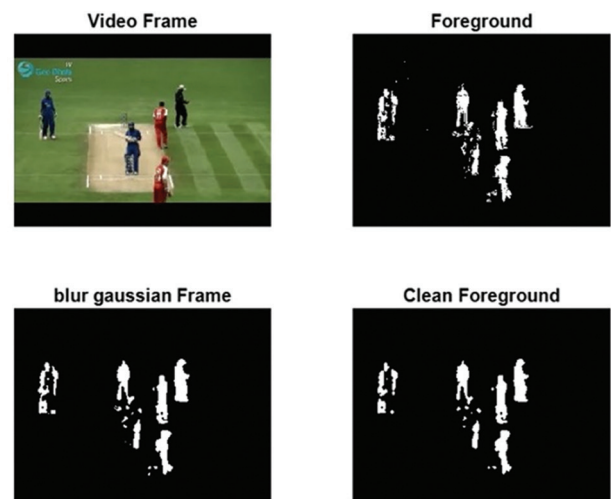


Fig. 6. VS using GMM Algorithm Outputs

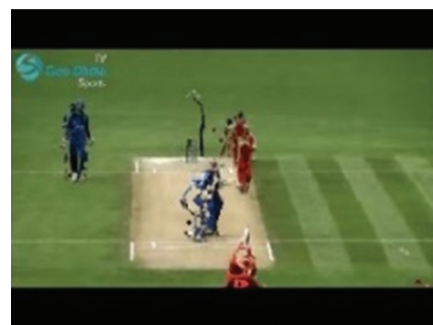


Fig. 7. Summarized Frame output

3.8. MULTI-FEATURE EXTRACTION

The proposed VSBD method splits the video stream into scenes or shots, and we select the key frames from each shot, considering the middle frame of a shot as a key frame. Apply the VS Algorithm to get a summarized frame. The multi-features of the summarized frame are extracted using equations (5-10), and a feature vector for that video stream is formed. This procedure of extracting a feature vector for all videos in the UCF da-

tabase [28] is done in offline mode. In online mode, this procedure is applied to the query video. Next, we perform a similarity measure by calculating correlation coefficients. The top 10 videos with high correlation coefficients are retrieved and displayed.

4. EXPERIMENT RESULTS

The performance of the CBVR System is tested on the UCF database [28], consisting of human action videos. We considered 20 classes, each with 10 videos, totalling 200 videos. Table 2 shows the properties of the database. A few examples of retrieved videos for the given query are shown in Table 3. The performance is evaluated using Precision (Pr), Loss, Compression Ratio (CR) [29], and online Execution Time (ET). The precision, loss, and CR are intended to be used with the equations (19-21). The superior precision values enhance the performance of the CBVR system. The average precision and loss for the UCF dataset are shown in Fig. 8. Table 4 discusses the comparison of our proposed CBVR system with other systems.

$$Precision (P_r) = \frac{\text{Correct no. of videos retrieved}}{\text{Total no. of videos retrieved}} \quad (19)$$










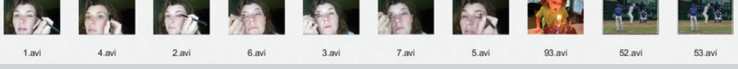

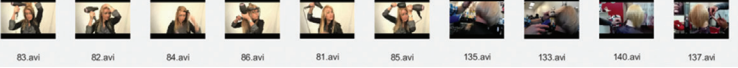

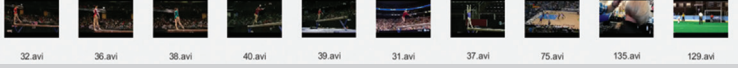
$$Loss = \frac{\text{Incorrect no. of videos retrieved}}{\text{Total no. of videos retrieved}} \quad (20)$$

$$Compression Ratio (CR) = 1 - \left(\frac{\text{No. of Key frames}}{\text{Total no. of frames}} \right) \quad (21)$$

Table 2. UCF Video Database Properties and Description

Description	Quantity
No. of Videos in Database	200
Average Duration	8.7513 seconds
Frame Rate	30 fps
Resolution	240x320
Average No. of Frames	153
Video Format	avi
Bits per pixel	24
Total Implementation time to extract features	1154.79832 seconds

Table 3. Examples of Retrieved Video streams for the given query

Class	Query	Retrieved videos	Pr	Loss	CR	ET (Sec)
Cricket Shot		 102.avi 104.avi 101.avi 110.avi 105.avi 107.avi 106.avi 109.avi 108.avi 151.avi	0.9	0.1	0.89	29.83
Field Hockey Penalty		 122.avi 124.avi 121.avi 127.avi 125.avi 126.avi 129.avi 128.avi 154.avi 151.avi	0.8	0.2	0.95	29.24
Hammer Throw		 143.avi 146.avi 141.avi 147.avi 144.avi 148.avi 145.avi 149.avi 142.avi 73.avi	0.9	0.1	0.92	21.18
Baseball Pitch		 52.avi 53.avi 54.avi 55.avi 51.avi 57.avi 56.avi 74.avi 149.avi 143.avi	0.7	0.3	0.95	20.55
Apply Eye Makeup		 1.avi 4.avi 2.avi 6.avi 3.avi 7.avi 5.avi 93.avi 52.avi 53.avi	0.7	0.3	0.92	20.27
Blow Dry Hair		 83.avi 82.avi 84.avi 86.avi 81.avi 85.avi 135.avi 133.avi 140.avi 137.avi	0.6	0.4	0.94	23.97
Balance Beam		 32.avi 36.avi 38.avi 40.avi 39.avi 31.avi 37.avi 75.avi 135.avi 129.avi	0.7	0.3	0.88	21.09

Plot of Average Precision and Loss of Proposed CBVR System

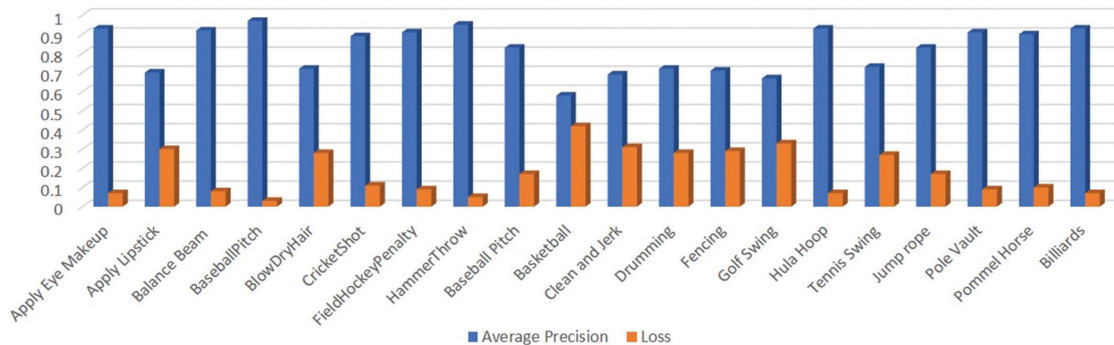


Fig. 8. Plot of Average Precision and Loss of different Video Classes

Table 4. Comparison of the other CBVR Systems to our Proposed work

Methods	Average Precision	Loss
CBVR using DCT [12]	0.6475	0.3525
CBVR using optimized perceptual video Summarization [17]	0.71	0.29
Proposed CBVR System	0.821	0.179

5. CONCLUSION

This paper proposes a novel method for CBVR, utilizing Multi-Feature DWHT and VS with the GMM Algorithm. From a video sequence, we first calculate the DWHT Multi-feature vector, and the correlation between successive frames is plotted. A procedure-based VSBD algorithm is used to divide the video into shots. Secondly, key frames are extracted, and the foreground is detected from them. A summarized frame is then stitched using GMM. From the summarized frame, multi-features are extracted and correlation coefficients between query and dataset videos are computed to retrieve similar videos. Experiments are performed on the UCF dataset, and the proposed CBVR system is evaluated. The proposed CBVR system has an average precision of 0.821 and a loss of 0.179, showing the performance of our work. In the future, we can improve the performance by making the system robust to camera motions and illumination variations.

6. REFERENCES

- [1] A. Moutaoukkel, A. Idarrou, I. Belahyane, "Information retrieval approaches: A comparative study", *International Journal of Electrical and Computer Engineering Systems*, Vol. 13, No. 10, 2022, pp. 961-970.
- [2] W. V. Ramos, A. P. Pumaleque, J. G. Torres, "Bibliometric Analysis of Scientific Production of Intelligent Video Surveillance", *International Journal of Electrical and Computer Engineering Systems*, Vol. 16, No. 6, 2025, pp. 461-471.
- [3] A. S. Adly, I. Hegazy, T. Elarif, M. S. Abdelwahab, "Development of an Effective Bootleg Videos Retrieval System as a Part of Content-Based Video Search Engine", *International Journal of Computing*, Vol. 21, No. 2, 2022, pp. 214-227.
- [4] G. S. N. Kumar, V. S. K. Reddy, L. K. Balivada, "Content-Based Video Retrieval Using Deep Learning Algorithms", *Intelligent Systems and Sustainable Computing*, Vol. 363, Springer, 2023, pp. 557-568.
- [5] W. Hu, N. Xie, L. Li, X. Zeng, S. Maybank, "A survey on Visual Content Based Video Indexing and Retrieval", *IEEE Transactions, On System, Man, And Cybernetics-Part C: Applications and reviews*, Vol. 41, No. 6, 2011, pp. 797-819.
- [6] A. Hussain, M. Ahmad, T. Hussain, I. Ullah, "Efficient content-based video retrieval system by applying AlexNet on key frames", *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, Vol. 11, No. 2, 2022, pp. 207-235.
- [7] D. Asha, Y. M. Latha, "Content-Based Video Shot Boundary Detection Using Multiple Haar Transform Features", *Advances in Intelligent Systems and Computing*, Vol. 900, Springer, 2019, pp. 703-713.
- [8] A. S. Adly, M. S. Abdelwahab, I. Hegazy, T. Elarif, "Issues and Challenges for Content-Based Video Search Engines A Survey", *Proceedings of the 21st International Arab Conference on Information Technology*, Giza, Egypt, 28-30 November 2020, pp. 1-18.
- [9] A. Araujo, B. Girod, "Large-Scale Video Retrieval Using Image Queries", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 28, No. 6, 2018, pp. 1406-1420.
- [10] L. Wang, X. Qian, X. Zhang, X. Hou, "Sketch-Based Image Retrieval With Multi-Clustering Re-Ranking", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 30, No. 12, 2020, pp. 4929-4943.
- [11] P. Nitish, J. Nishita, D. Nishith, J. Bharati, "Content Based Video Retrieval using Deep Learning", *Research Square*, 2024, pp. 1-20.
- [12] S. Hamad, A. S. Farhan, D. Y. Khudhur, "Content based video retrieval using discrete cosine transform", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 21, No. 2, 2021, pp. 839-845.
- [13] B. Sathiyaprasad, K. Seetharaman, B. S. Kumar, "Content based Video Retrieval using Improved Gray Level Co-Occurrence Matrix with Region-based Pre-Convolutional Neural Network-RPCNN", *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems*, Thoothukudi, India, 3-5 December 2020, pp. 558-563.
- [14] A. Dyana, S. Das, "MST-CSS (Multi-Spectro-Temporal Curvature Scale Space) a Novel Spatio-Temporal Representation for Content-Based Video Retrieval", *IEEE Transactions on Circuits and Sys-*

- tems for Video Technology, Vol. 20, No. 8, 2010, pp. 1080-1094.
- [15] S. S. Gornale, A. K. Babaleshwar, P. L. Yannawar, "Analysis and Detection of Content based Video Retrieval", *International Journal of Image, Graphics and Signal Processing*, Vol. 11, No. 3, 2019, pp. 43-57.
- [16] A. K. Mallick, S. Mukhopadhyay, "Video Retrieval Based on Motion Vector Key Frame Extraction and Spatial Pyramid Matching", *Proceedings of the 6th International Conference on Signal Processing and Integrated Networks*, Noida, India, 7-8 March 2019, pp. 687-692.
- [17] S. S. Thomas, S. Gupta, V. K. Subramanian, "Context Driven Optimized Perceptual Video Summarization and Retrieval", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 29, No. 10, 2019, pp. 3132-3145.
- [18] D. Asha, Y. M. Latha, V. S. K. Reddy, "Content Based Video Retrieval system using Multiple Features", *International Journal of Pure and Applied Mathematics*, Vol. 118, No. 14, 2018, pp. 287-294.
- [19] R. C. Gonzalez, R. E. Woods, "Digital Image Processing", Second edition, Pearson, 2019.
- [20] Q. Zhong, Y. Zhang, J. Zhang, K. Shi, Y. Yu, C. Liu, "Key Frame Extraction Algorithm of Motion Video Based on Prior", *IEEE Access*, Vol. 8, 2020, pp. 174424-174436.
- [21] Z.-M. Lu, Y. Shi, "Fast Video Shot Boundary Detection Based on SVD and Pattern Matching", *IEEE Transactions on Image Processing*, Vol. 22, No. 12, 2013, pp. 5136-5145.
- [22] G. L. Priya, S. Domnic, "Walsh-Hadamard Transform Kernel-Based Feature Vector for Shot Boundary Detection", *IEEE Transactions on Image Processing*, Vol. 23, No. 12, 2014, pp. 5187-5197.
- [23] G. L. Priya, S. Domnic, "Video Cut Detection using block-based Histogram Differences in RGB Color Space", *Proceedings of the International Conference on Signal and Image Processing*, Chennai, India, 15-17 December 2010, pp. 29-33.
- [24] F. Joy, V. Vijayakumar, "An improved Gaussian Mixture Model with post-processing for multiple object detection in surveillance video analytics", *International Journal of Electrical and Computer Engineering Systems*, Vol. 13, No. 8, 2022, pp. 653-660.
- [25] A. Lajari, R. Sachin, "Dealing Background Issues in Object Detection using GMM: A Survey", *International Journal of Computer Applications*, Vol. 150, No. 5, 2016, pp. 50-55.
- [26] A. Nurhadiyatna, W. Jatmiko, B. Hardjono, A. Wibisono, I. Sina, P. Mursanto, "Background Subtraction Using Gaussian Mixture Model Enhanced by Hole Filling Algorithm (GMMHF)", *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Manchester, UK, 13-16 October 2013, pp. 4006-4011.
- [27] K. M. Angelo, "A novel approach on object detection and tracking using adaptive background subtraction method", *Proceedings of the Second International Conference on Computing Methodologies and Communication*, Erode, India, 15-16 February 2018, pp. 1055-1059.
- [28] K. Soomro, A. R. Zamir, M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild", *CRCV-TR-12-01*, November, 2012.
- [29] D. Rajeshwari, V. Priscilla C, "An Enhanced Spatio-Temporal Human Detected Keyframe Extraction", *International Journal of Electrical and Computer Engineering Systems*, Vol. 14, No. 9, 2023, pp. 985-992.