

A Novel Content-Based Video Retrieval System Using MBH and BCHOA Optimized Multiple Features

Asha D.¹, Madhavee Latha Y.²

¹Research Scholar, Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Hyderabad, Telangana, India.

²Professor, Department of Electronics and Communication Engineering, Malla Reddy Engineering College for Women, Affiliated to JNT University, Hyderabad, Telangana, India.

E-mail: ¹ashamanickrao@gmail.com, ²madhaveelatha2009@gmail.com

Abstract

Usage of multimedia data is increasing drastically every day, so being able to manage and effectively retrieve information based on its content has become important. A content-based video retrieval (CBVR) system retrieves video data based on visual content. This paper presents a CBVR system that uses visual descriptors, including Motion Boundary Histogram (MBH) temporal features, along with RGB codes and GLCM spatial features. By analysing the optical flow derivatives, MBH features are extracted from each frame to detect shot transitions and identify key frames that summarize the video content. From the keyframes, multiple features are extracted for dataset videos in the offline process and for the query video in the online process. To reduce redundancy and improve retrieval accuracy, feature selection is performed on the feature vector using various optimization algorithms, such as BPSO, BGWO, BWOA, and BCHOA. The performance of the system is tested on sample videos from the UCF50 dataset. The comparison between query and dataset videos is evaluated by computing cosine similarity. The retrieval performance and computational efficiency of the proposed CBVR framework are compared with existing methods by achieving a compression ratio of 0.9701, an F1 score of 0.859, a precision of 0.972 and a recall rate of 0.770.

Keywords: Content-Based Video Retrieval (CBVR), Video Shot Boundary Detection (VSBD), Motion Boundary Histogram (MBH), Feature Selection (FS), Binary Particle Swarm Optimization (BPSO), Binary Grey Wolf Optimization (BGWO), Binary Whale Optimization Algorithm (BWOA), Binary Chimp Optimization Algorithm (BCHOA).

1. Introduction

Multimedia information has become a part of human lifestyle, playing an important role. It comprises videos, text, audio and images [1]. The volume of widely available multimedia information is rapidly expanding due to developments in information technology and advances in multimedia strategies [2]. Video is a significant form of multimedia, containing a huge amount of information and rich in content, created by playing still images sequentially at a faster rate. Since video is large in volume because of redundancy between frames, managing its size is difficult, and processing it is a complex task that creates demand and research scope for video retrieval systems.

The primary approaches for video retrieval are text-based and content-based. Text-based retrieval uses the metadata of a video file, including the file name, titles, tags, annotations or descriptions of the video file. It is an age-old method, and it is less efficient if descriptions or annotations are not given correctly; they are unreliable and more vulnerable to manipulation. Next is a content-based retrieval system, which uses the content information of the video by extracting its features and then comparing them. Presently, content-based retrieval has gained a lot of importance because of its efficiency in segregating information and overcoming all the drawbacks of text-based retrieval systems.

Content-based retrieval enables the application to access video clips using automatically extracted visual features such as color, texture, edges, and shape, rather than unrelated attributes. Text-based retrieval methods are effective when multimedia information is well-annotated. If annotations are incomplete or missing, the information cannot be retrieved. Content-based video retrieval (CBVR) has gained popularity as the use of multimedia data has increased significantly in daily life, surpassing the use of text-based search.

2. Literature Review

A literature review was conducted on various CBVR systems, VSBD techniques, and dimensionality reduction methods. Maojin et al. [3] study introduces a CBVR system that applies a color histogram for keyframe detection and the AlexNet convolutional neural network for feature extraction to the UCF101 dataset. It is suggested for improvements by exploring different color features further. The system achieved an average accuracy of 90%. Processing a 164-frame video on a GPU took 1,186 seconds. Hyeok et al. [4] present a novel CBVR method, Prototypical Category Approximation Content-Based Video Retrieval (PCA-CBVR), evaluated on the UCF101, HMDB51, and ActivityNet datasets. The approach utilises prototypes of deep features, few-shot learning for domain adaptation, and salient frame sampling to mitigate information loss and minimize the need for frequent classifier retraining. Thomas et al. [5] discuss the progression of video summarization from key frame extraction to object-based dynamic synopses. It addresses challenges in video browsing and retrieval, emphasizing the need for efficient methods to manage large video databases, and proposes a single-frame indexing approach. Nitish et al. [6] focus on VGG-16, Inception V3, and ResNet CNN models and explore opportunities to optimize these models and integrate different feature extraction methods, yet are restricted by high computational time. Shivanand et al. [7] recommend using Regions of Interest, especially for signboards, with an ACF detector.

Liu et al. [8] address feature dimensionality reduction, as high-dimensional content features can hinder retrieval efficiency. The authors introduce a method that integrates deep learning for feature extraction with Principal Component Analysis for dimensionality reduction. Shami et al. [9] work on surveying Particle Swarm Optimization (PSO), an efficient algorithm for various optimization problems. Nevertheless, PSO encounters problems like premature convergence, parameter control deficiency and velocity adjustment issues. A new grey wolf optimization algorithm for EMG Pattern recognition is introduced, known as the opposition-based competitive grey wolf optimizer (OBCGWO), in Too et al. [10]. It results in more efficiency for rehabilitation applications with a penalty of higher computational time. A novel Competitive Binary Grey Wolf Optimizer (CBGWO) is described in Too et al. [11], addressing the feature selection problem applied to the EMG signal. Its competition and leader enhancement strategies help avoid local optima and efficiently identify global optima. Too et al. [12] address high-dimensional feature selection using the Spatial Bound Whale

Optimization Algorithm (SBWOA). This approach enhances the search capability of WOA, resulting in improved accuracy, smaller feature sets, and faster computation by excluding irrelevant, redundant, and noisy features. Ayeche et al. [13] present a hybrid feature selection method that combines enhanced binary Chimp Optimization Algorithms with standard classifiers. This approach addresses the curse of dimensionality by selecting optimal features, leading to a higher classification accuracy of 82.02% and faster convergence across multiple high-dimensional datasets.

Shrikant et al. [14] introduce a robust shot boundary detection (SBD) method that integrates FAPG filtering with DTCWT-WHT feature fusion, evaluated on TRECVID datasets. Kar et al. [15] offer a comprehensive review of methodologies and algorithms used in shot boundary detection over the past two decades. They discuss the challenges in varying conditions in video capturing and suggest the integration of multiple features with strategies for feature reduction.

The following research gaps were identified: identifying abrupt and gradual shot transitions using rule-based algorithms and setting the threshold to divide the video into shots by employing optimization techniques for automatic threshold calculation. Choosing features for feature extraction using feature fusion and multiple features. Utilize dimensionality reduction methods, such as feature selection. The paper is ordered as follows: Section 2 presents the literature review conducted; Section 3 provides a detailed explanation of the proposed CBVR system; Sections 4 and 5 illustrate the results, including discussions and conclusions.

3. Proposed CBVR System Using Optimized Multiple Features

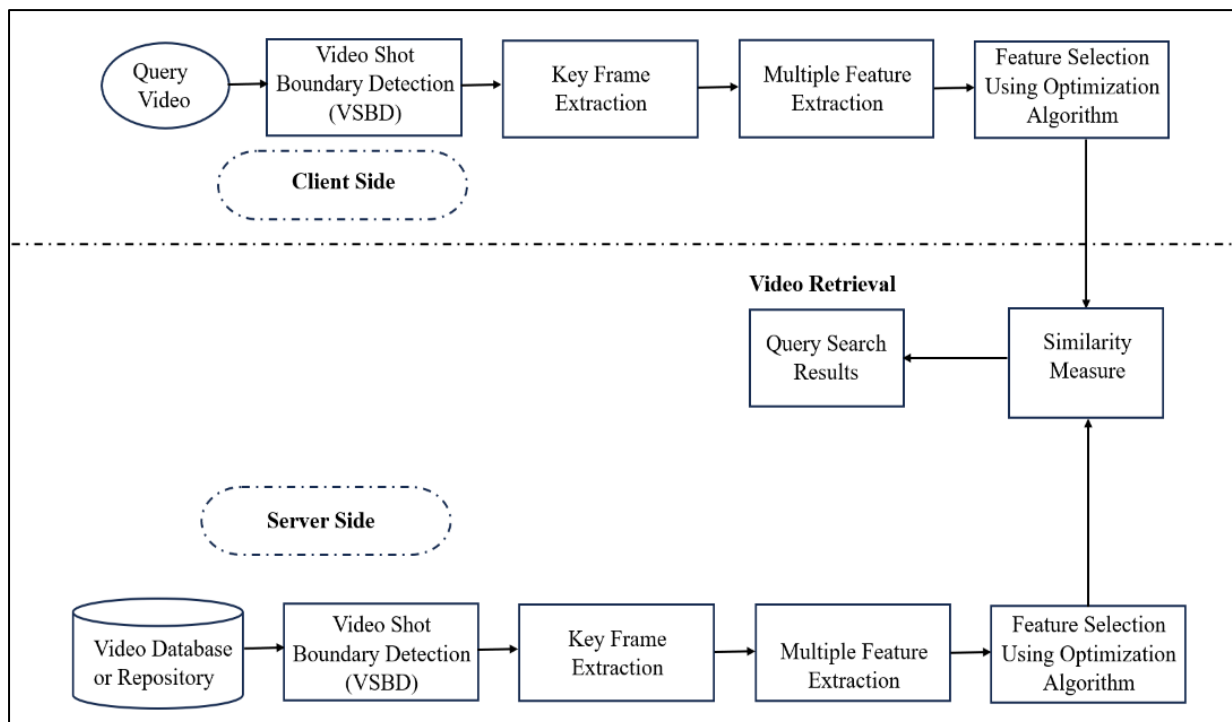


Figure 1. Block Diagram of Proposed CBVR System Using Optimized Multiple Features

The proposed CBVR system is illustrated in Figure 1. The entire process is divided into server-side (offline) and client-side (online) components. The CBVR system consists of five steps, beginning with VSBD, where the video is divided into shots Benoughidence et al. [16].

There are two types of shot transitions: abrupt shot transitions and gradual shot transitions Asha et al. [17]. From the detected shots, key frames are extracted, and then multiple features are extracted. To reduce the dimensionality, an optimization algorithm for feature selection is applied. On the server side, the video repository is considered as input, and all steps are applied, extracting optimized feature vectors for all videos in the database. On the client side, the process is applied to the query video, and then the optimized multiple feature vector for the query video is compared with the database feature vector to measure similarity.

3.1 Proposed VSBD Method

Shot boundaries, such as cuts, fades, and dissolves, typically cause abrupt or systematic motion changes Asha et al. [17]. MBH identifies motion boundaries by analysing optical flow gradients Fan et al. [18]. When a shot boundary occurs, the flow consistency between consecutive frames is disrupted, resulting in abrupt changes in MBH histograms. It remains robust to camera or global motion. Since global motion produces nearly uniform flow, the gradients effectively disappear. As a result, MBH emphasizes local motion differences, which are essential for detecting genuine content changes between shots. The proposed VSBD is shown in Figure. 2.

VSBD Algorithm

Step 1: Read the video and convert the RGB frame to grayscale.

Step 2: Compute Optical Flow using the Farneback method as in (1-2).

$$I(x, y) = [u(x, y), v(x, y)] \quad (1)$$

where $u(x, y), v(x, y)$ are horizontal and vertical displacements respectively

$$I(x) \approx x^T A_x + b^T x + c \quad (2)$$

where $x = (x, y)^T$ is the pixel location, $A =$ Symmetric 2×2 matrix, $b =$ vector
(gradient term, $c =$ Scalar (offset))

Step 3: Compute spatial gradients of optical flow to focus on motion boundaries as in (3-5)

$$u_x = \frac{\partial u}{\partial x}, u_y = \frac{\partial u}{\partial y}, v_x = \frac{\partial v}{\partial x}, v_y = \frac{\partial v}{\partial y} \quad (3)$$

$$u_x(x, y) \cong u(x + 1, y) - u(x - 1, y), u_y(x, y) \cong u(x, y + 1) - u(x, y - 1) \quad (4)$$

$$v_x(x, y) \cong v(x + 1, y) - v(x - 1, y), v_y(x, y) \cong v(x, y + 1) - v(x, y - 1) \quad (5)$$

Step 4: Compute gradient magnitude and orientation as in (6-7)

$$\text{for horizontal flow}(u): M_u(x, y) = \sqrt{u_x^2 + u_y^2}, \theta_u(x, y) = \arctan2(u_y, u_x) \quad (6)$$

$$\text{for vertical flow}(v): M_v(x, y) = \sqrt{v_x^2 + v_y^2}, \theta_v(x, y) = \arctan2(v_y, v_x) \quad (7)$$

Step 5: Build a Histogram per cell by dividing the frame into 16×16 cells. For each cell, a histogram of orientation is computed separately for u and v , as shown in (8).

$$h_i(bin) = \sum_{(x,y) \in cell} M(x,y) \cdot \delta(\theta(x,y) \in bin) \quad (8)$$

$$\text{where } \delta = \text{indicator function } \delta = \begin{cases} 1 & \text{if } \theta \text{ in bin} \\ 0 & \text{else} \end{cases}$$

Step 6: To form MBH descriptors compute histogram normalization and concatenate u and v components.

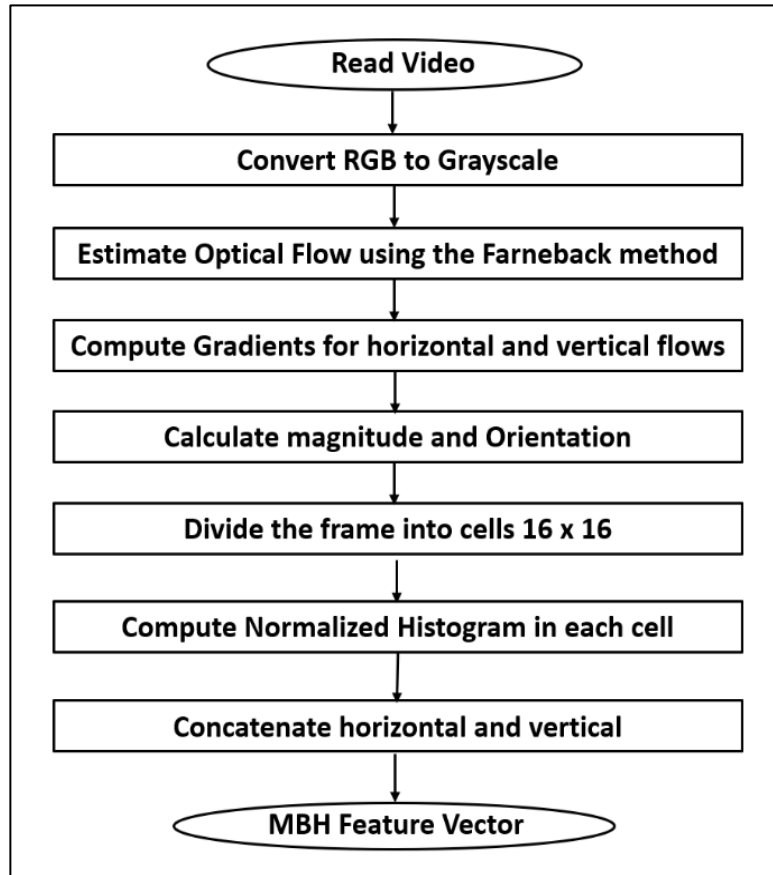


Figure 2. Proposed VSBD Method Using MBH

3.1.1 Motion Boundary Histogram (MBH)

MBH computes spatial derivatives of optical flow fields on a frame in both vertical and horizontal directions. These gradient descriptors identify motion boundaries to capture local motion patterns as they are robust to global movements of the camera, reduce the effect of uniform motion, and highlight relative motion changes. To estimate dense optical flow, the Farneback method Ma et al. [19] is used, where each neighbourhood of pixels is modelled with a quadratic polynomial, and the displacement is determined. Optical flow refers to the perceived motion of objects, surfaces, and edges in a video, caused by object or camera movement. It measures motion between consecutive frames. The Farneback algorithm is a dense optical flow method that estimates motion for every pixel, unlike feature-based methods that track specific points. The algorithm builds a multi-level image pyramid, with each level at a lower resolution than the previous one, as shown in Figure 3.

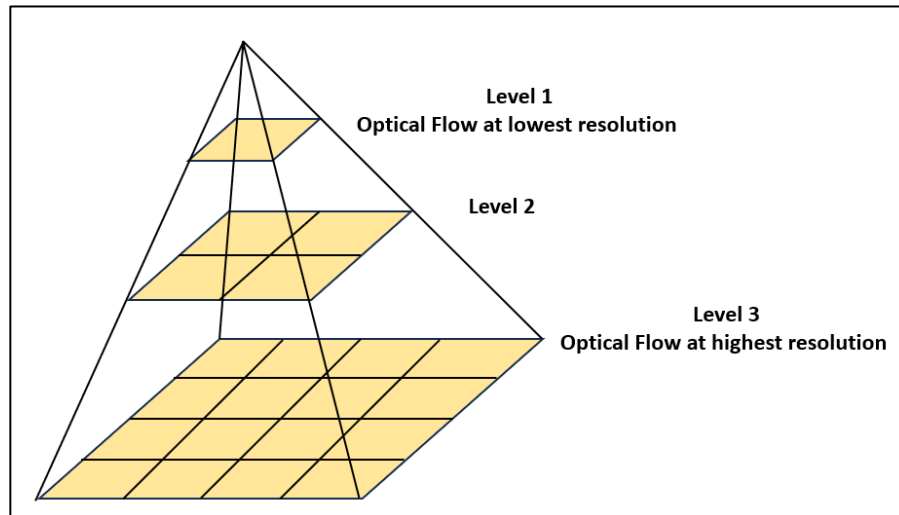


Figure 3. Image Pyramid of the Farneback Optical Flow Method

MBH features are computed for each frame, and the similarity between successive frames is measured, yielding a similarity signal. A threshold is set to detect the shot transitions in the video. The optimum threshold is determined by searching for valley points in the similarity signal and calculating the average of these points, Asha et al. [20]. A rule-based algorithm is run to classify the type of shot transition. If the values in the similarity signal are less than the threshold for more than one frame, then it is categorized as a gradual transition; else, it is considered an abrupt shot transition Asha et al. [20]. The key frames are extracted by considering the middle frame of a shot along with the first and last frames.

3.2 Multiple Feature Extraction

Feature extraction translates raw information into representative values while reducing dimensionality, protecting crucial information, and eradicating redundant and unrelated data. We extracted multiple features from the same frame, as each feature captures a unique aspect. Combining these multiple features improves accuracy and robustness. The feature extraction methods used are RGB code-based features, texture descriptors like the Gray Level Co-occurrence Matrix (GLCM), Benco et al. [21], and motion descriptors like MBH features.

3.2.1 RGB Code-based Features

The RGB code-based feature uses 14 distinct colors by analyzing RGB value relationships and ranges as described in the Table 1. It preserves the spatial coordinate information and makes it less robust to changes in lighting. This method, however, completely ignores the texture features. The conditional equations for assigning an RGB code are prescribed in (9).

```

if (r > g && r > b && g == b) then A(i,j)="R";
    elseif (r > g && r > b && g > b) then A(i,j)="O";
    elseif (r == g && r > b && g > b) then A(i,j)="Y";
    elseif (g > r && g > b && r > b) then A(i,j)="LG";
    elseif (r == b && g > r && g > b) then A(i,j)="G";

```

```

elseif (g > b && g>r && g>b) then A(i,j)="CY";
elseif (g == b && g>r && b>r) then A(i,j)="LB";
elseif (b > g && b>r && g>r) then A(i,j)="B";
elseif (r == g && b>g && b>r) then A(i,j)="I";
elseif (b > r && b>g && r>g) then A(i,j)="V";
elseif (r == b && r>g && b>g) then A(i,j)="M";
elseif (r > g && r>b && b>g) then A(i,j)="P";
elseif ((r == g) &&(r == b) && g<128) then A(i,j)="BK";
elseif ((r == g) &&(r == b) && g>128) then A(i,j)="W";

```

end (9)

Table 1. Color Code Analysis and RGB Relationships

COLOR	R	G	B	RELATION	ASSIGNED VALUE
RED	51	0	0	R>G>B AND G=B	255,0,0
	255	204	204		
ORANGE	51	25	0	R>G>B AND G>B	255,128,0
	255	229	204		
YELLOW	51	51	0	R=G AND R>B AND G>B	255,255,0
	255	255	204		
LIGHT GREEN	25	51	0	G>R>B AND R>B	128,255,0
	229	255	204		
GREEN	0	51	0	R=B AND G>R AND G>B	0,255,0
	204	255	204		
CYAN	0	51	25	G>B>R AND G>B	0,255,128
	204	255	229		
SKY BLUE	0	51	51	G=B AND G>R AND B>R	0,255,255
	204	255	255		
BLUE	0	25	51	B>G>R AND G>R	0,128,255
	204	229	255		
INDIGO	0	0	51	R=G AND B>G AND B>R	0,0,255
	204	204	255		
VILOT	25	0	51	B>R>G AND R>G	128,0,255
	229	204	255		
MAGENTA	51	0	51	R=B AND R>G AND B>G	255,0,255
	255	204	255		
PINK	51	0	25	R>G>B AND B>G	255,0,128
	255	204	229		
BLACK	0	0	0	R=G=B AND R,G,B<128	0,0,0
WHITE	255	255	255	R=G=B AND R,G,B>128	255, 255, 255

3.2.2 GLCM

GLCM is a texture feature descriptor that uses statistical techniques—contrast, correlation, energy, homogeneity, and entropy, as in (10-14), to evaluate the relationship between neighbourhood pixels. It extracts variation, dependency, uniformity, smoothness, and randomness in gray levels. We considered pixel distance as 1 and angle as 00.

$$\text{Contrast} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i-j)^2 P(i,j) \quad (10)$$

$$\text{Correlation} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{(i-\mu_i)(i-\mu_j)P(i,j)}{\sigma_i \sigma_j} \quad (11)$$

$$\text{Energy} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P(i,j)^2 \quad (12)$$

$$\text{Homogeneity} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{P(i,j)}{1+|i-j|} \quad (13)$$

$$\text{Entropy} = - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P(i,j) \log P(i,j) \quad (14)$$

where i, j are rows and columns of Co

– occurrence matrix, $P(i, j)$ is Probability matrix,

L is No. of gray levels ($L = 8$), μ and σ are mean and standard deviation of rows and columns

3.3 Feature Selection (FS)

FS reduces the dimensionality of the feature vector by retaining the relevant features and eliminating the redundant features Jia et al. [22], preventing overfitting and simplifying the data. The procedure of FS includes initialization, generation, evaluation, stopping criteria, and validation. The main factors that affect the performance are the choice of search algorithm, the evaluation measures and the number of objectives. FS methods Girish et al. [23] are classified into four classes: Filter, Wrapper, Embedded, and Metaheuristic. Filter methods use statistical or information-based criteria to assess each feature independent of the learning algorithm. In the wrapper method, a predictive model is used by selecting a subset of features with the best performance. Embedded methods perform feature selection during model training by assigning importance or reducing the weights of irrelevant features. Metaheuristic methods use optimization algorithms inspired by nature or mathematics to identify optimal or near-optimal feature subsets. These approaches are well-suited for managing complex, high-dimensional, and non-linear feature interactions. We implemented four nature-inspired optimization algorithms: BPSO, BGWO, BWOA, and BCHOA.

3.3.1 Binary Particle Swarm Optimization (BPSO)

BPSO is an extensively used optimization method in many applications due to its fast convergence, minimal parameter requirements, short execution time and simplicity of implementation. However, it is sensitive to parameter settings. BPSO is computed using the following equations (15-16).

$$V_k(t+1) = wV_k(t) + c_1 r_1 (P_{best\ k}(t) - x_k(t)) + c_2 r_2 (G_{best}(t) - x_k(t)) \quad (15)$$

where w, c_1, c_2 are PSO parameters ($c_1 = c_2 = 2$), k^{th} element, x is position and V is velocity

$$w = w_{max} - (w_{max} - w_{min}) * \left(\frac{t}{T}\right) \quad (16)$$

where t^{th} , T is maximum No. of iterations ($w_{max} = 0.9, w_{min} = 0.4$)

3.3.2 Binary Grey Wolf Optimization (BGWO)

In the BGWO, the population is grouped into four: alpha, beta, delta, and omega. Grey wolves live in packs of 5 to 12 members. Decision-making is performed by the alpha, who acts as the main leader. The supporter of the alpha is the beta, who acts as the second leader, while the delta is the third leader and manages the omega grey wolves. Too et al. [10]. The three fittest solutions X1, X2, and X3 are designated for alpha, beta, and delta, respectively, who are involved in the hunting process, whereas omega follows their guidelines, Too et al. [11] as in (17-19).

$$X(t + 1) = X_p(A) - A.D \tag{17}$$

$$D = |C.X_p(t) - X(t)| \tag{18}$$

Where X_p is the position of the prey, A and C are Coefficient Vectors, t – iterations, $A = 2a.r_1 - a$ and $C = 2.r_2$

Where r_1, r_2 are two random vectors in interval [0 1], a is parameter that linearly

$$\text{decrease from 2 to 0, and } a = 2 - 2\left(\frac{t}{T}\right)$$

Where T is the maximum number of iterations

$$X(t + 1) = \frac{X_1 + X_2 + X_3}{3} \tag{19}$$

3.3.3 BWOA (Binary Whale Optimization Algorithm)

BWOA is inspired by the humpback whale's bubble-net feeding, an encircling technique used to capture prey. It employs two strategies for exploration: The Shrinking Encircling (Se) Strategy and the Spiral Updating (SU) Strategy, as noted by Fan et al. [18] in (20-24).

$$\text{Shrinking encircling (Se) } X(t + 1) = X_{gb}(t) - A.D \tag{20}$$

$$D = |C.X_{gb}(t) - X(t)| \tag{21}$$

Where X is the position of the whale, X_{gb} is the position of the best hunting agent, A and C are Coefficient Vectors, t – iterations, $A = 2a_1.r_1 - a_1$, $C = 2.r_2$

Where

r_1, r_2 are two random vectors in interval [0 1] a_1 is parameter that linearly decrease

$$\text{from 2 to 0, } a_1 = 2 - 2\left(\frac{t}{T}\right)$$

$$\text{Spiral updating (Su) } X(t + 1) = E.\exp(b_t).(2\pi l) + X_{gb}(t) \tag{22}$$

$$E = |X_{gb}(t) - X(t)| \tag{23}$$

where b is the constant to determine shape of logarithmic Spiral usually ($b = 1$)

l is random vector $[-1 \ 1]$

$$X(t+1) = \begin{cases} X_{gb}(t) - A \cdot D & p < 0.5 \\ E \cdot \exp(b_t) \cdot (2\pi l) + X_{gb}(t) & p \geq 0.5 \end{cases} \quad (24)$$

Where p is a random vector in the interval $[0,1]$

3.3.4 Binary Chimp Optimization Algorithm (BCHOA)

Chimpanzees form social groups. The BCHOA model categorizes populations into four types: Attackers, Barriers, Chasers, and Drivers. Attackers block escape routes, capturing prey or directing them toward hunters. Barriers restrict movement by creating obstacles. Chasers pursue prey quickly to induce panic and close the gap. Drivers encircle prey without initiating an attack. The formulation is as in (25-31).

$$X(t+1) = \frac{X_1 + X_2 + X_3 + X_4}{4} \quad (25)$$

$$X_1 = |X_a - A_1 \cdot D_a|, X_2 = |X_b - A_2 \cdot D_b|, X_3 = |X_c - A_3 \cdot D_c|, X_4 = |X_d - A_4 \cdot D_d| \quad (26)$$

$$D_a = |C_1 \cdot X_a - M_1 \cdot X|, D_b = |C_2 \cdot X_b - M_2 \cdot X|$$

$$D_c = |C_3 \cdot X_c - M_3 \cdot X|, D_d = |C_4 \cdot X_d - M_4 \cdot X| \quad (27)$$

Where M represents a chaotic function which generates a sequence of values in range of 0 to 1

$$M_{n+1} = \cos(6 * \arccos(M_n)) \quad (28)$$

Where X_p is the position of the prey, A , M , and C are Coefficient Vectors, t – iterations

$$A_1 = 2 \cdot f \cdot r_{11} - f, A_2 = 2 \cdot f \cdot r_{21} - f, A_3 = 2 \cdot f \cdot r_{31} - f, A_4 = 2 \cdot f \cdot r_{41} - f \quad (29)$$

$$C_1 = 2 \cdot r_{12}, C_2 = 2 \cdot r_{22}, C_3 = 2 \cdot r_{32}, C_4 = 2 \cdot r_{42} \quad (30)$$

Where r is two random vectors in interval $[0 \ 1]$, f decreases nonlinearly from 2 to 0

$$f = 2 - 2 \left(\frac{t}{T} \right) \quad (31)$$

Where T is the maximum No. of iterations

Algorithm for Binary Feature Selection

Inputs: Total number of Multiple Features (TMF)=1584

Population size (N) = 10

Number of iterations (T) = 50 or 100

Multiple Feature Vector of dataset =MF_{2919 x 1584}

Labels= Y_{2919 x 1}

Classifier=KNN (k=5)

Optimization methods: BPSO, BGWO, BWOA, BCHOA

Outputs: Selected features (SF)

Convergence curve (CC)

Step 1: Initialization $X_i = [x_{i1}, x_{i2} \dots x_{iTMF}] \in \{0,1\}$
 where $x_{ij} = 1$ (feature Selected) or 0 (feature Not Selected)

Step 2: Compute the fitness function (FitF) for all solutions as in (32).

$$Error = 1 - \left[\frac{Correctly\ predicted}{Total\ Valid} \right] \quad (32)$$

Step 3: Update solutions using binary optimization algorithms.

Step 4: Generate the updated population with new feature subsets.

Step 5: Update the best individuals based on velocity, leader, or encircling prey criteria.

Step 6: Check the stopping criteria. If met, stop; otherwise, repeat the process.

3.4 Similarity Measure

In a CBVR system, a similarity measure compares the feature vector of a query video (FV_Q) to those of videos in the database (FV_{DB}), indicating how closely they match. The cosine similarity is used as it is effective for comparing high-dimensional vectors Hyeok et al. [4]. The cosine similarity is defined as in (33).

$$Cosine_{sim}(i) = \sum_{i=1}^K \frac{FV_Q \cdot FV_{DBi}}{\|FV_Q\| \|FV_{DBi}\|} \quad (33)$$

where K is No. of Key frames in database

4. Results and Discussions

The proposed CBVR system using the UCF50 dataset is evaluated in Kishore et al. [24], which contains 500 videos across 50 classes. Experiments were performed on an Intel Core i5 processor using MATLAB 2021a. The metrics used to evaluate the performance of our system are Compression Rate (CR), Precision (PR), Recall Rate (RR), F1 score, and Feature Reduction Rate (FRR), as shown in (34-38). The UCF50 sample dataset descriptions are provided in Table. 2.

$$PR = \frac{No.of\ Videos\ Correctly\ Retrieved}{Total\ No.of\ Videos\ retrieved} \quad (34)$$

$$RR = \frac{No.of\ Videos\ Correctly\ Retrieved}{Total\ No.of\ Videos\ in\ each\ class} \quad (35)$$

$$F1\ Score = \frac{N2 * PR * RR}{(PR + RR)} \quad (36)$$

$$CR = 1 - \frac{No.of\ Extracted\ Key\ Frames}{Total\ No.of\ Frames} \quad (37)$$

$$FRR = 1 - \frac{|Selected\ Features|}{|Total\ No.\ of\ Features|} \quad (38)$$

Table 2. Video Dataset Properties

Description	Properties
Total No. of Videos in the dataset	500
No. of Video Classes	50
Frame Rate	25 or 30 fps
Total Duration of the dataset	3674.97 seconds
Resolution of the frame	240 x 320
Video Format	.avi

Total No. of frames in the dataset	97802
Bits per pixel	RGB 24
Execution Time for key frames extraction	9598.23 seconds
Total No. of Key Frames Extracted	2919
Compression Ratio	0.9701
Execution time for Multiple Feature Extraction	146.13 seconds
Size of Multiple Feature Vector	2919 x 1584

The initial step VSBD, when applied, extracted 2919 key frames out of 97802 total frames, with a CR of 0.9701. The VSBD plot for the Golf Swing video is shown in Figure 4, where two shots are identified and two key frames are extracted from each shot, as shown in Figure. 5. The next feature extraction stage outputs RGB codes, GLCM, and MBH, which are illustrated in Figures 6, 7 and 8. Feature selection optimization algorithms are applied. The optimization algorithms applied are BPSO, BGWO, BWOA, and BCHOA, their results are summarized in Table.3, and the convergence curves are presented in Figure 9. The BCHOA outperforms BPSO, BGWO, and BWOA with a 0.768 FRR and a 0.496 average convergence curve, with an average computation time of 1205.9 seconds. The proposed CBVR system is compared with the CBVR system using AlexNet Maojin et al. [3], as shown in Table. 4. Video retrieval analysis for the selected examples is presented in Table. 5. Figure 10 shows the graphical results for different video classes. The confidence intervals for PR, RR and F1 score are (0.9525, 0.9914), (0.7481, 0.7918) and (0.83987, 0.87232), respectively.

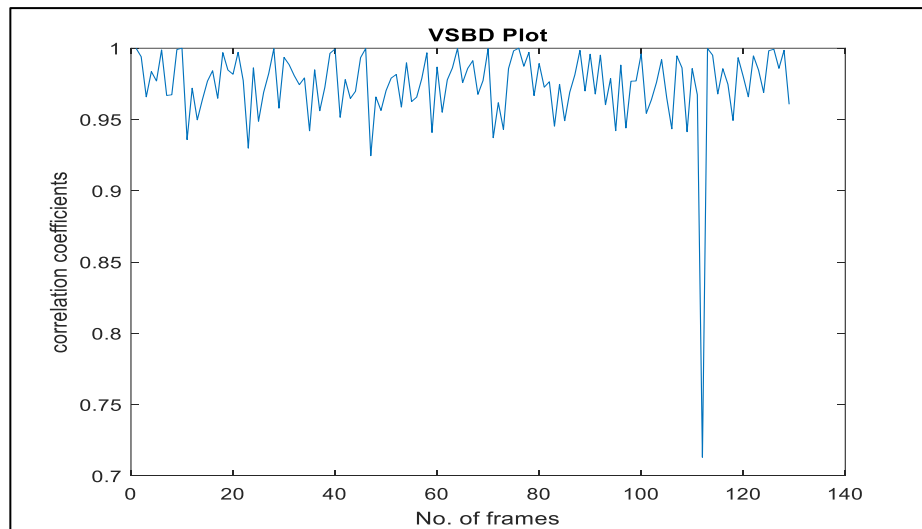


Figure 4. VSBD Plot for Golf Swing Video



Figure 5. Extracted Key Frames for Golf Swing Video

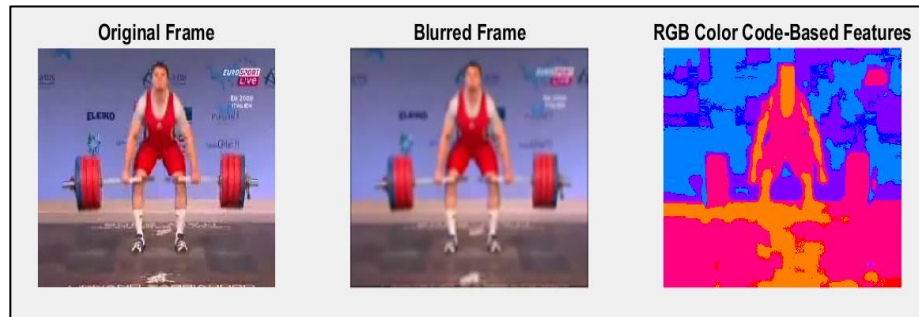


Figure 6. RGB Code- Based Extracted Features

8x8 double									Field ^	Value
1	26	38	6	2	2	2	1	0	Contrast	0.1866
2	40	2153	455	75	32	8	3	1	Correlation	0.9322
3	8	460	21328	1215	142	36	19	3	Energy	0.2000
4	2	88	1186	19948	853	125	34	5	Homogen...	0.9423
5	1	21	156	864	15951	682	89	11	Entropy	0.3955
6	0	5	46	79	678	7259	282	21		
7	0	6	14	30	70	298	1364	75		
8	0	0	5	6	17	17	71	146		

Figure 7. GLCM Co-Matrix and Features

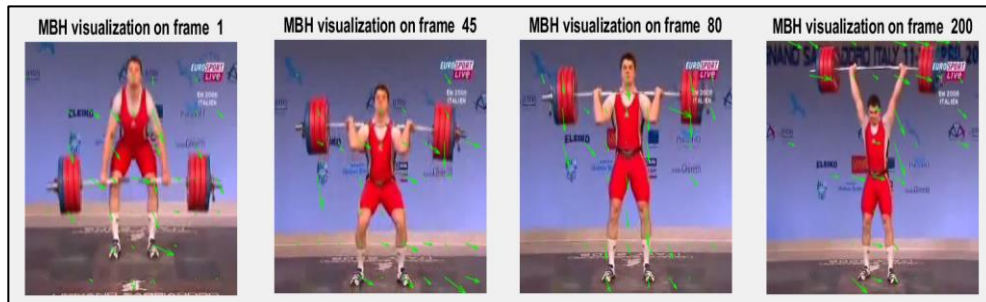


Figure 8. MBH Features for a Few Frames

Table 3. Summarized Optimization Analysis

Optimization Algorithm	Training %	No. of Iteration	Selected Features	Execution time (Seconds)	Convergence Curve	FRR	PR	RR	F1 Score
BCHOA	20	50	448	141.769	0.492	0.717	0.959	0.755	0.845
	40	50	461	249.019	0.504	0.709	0.968	0.761	0.852
	20	100	298	198.598	0.489	0.812	0.978	0.781	0.868
	40	100	263	234.497	0.499	0.834	0.981	0.784	0.872
Average			367.5	205.971	0.496	0.768	0.972	0.770	0.859
WOA	20	50	756	693.678	0.507	0.523	0.910	0.730	0.810
	40	50	735	1867.761	0.510	0.536	0.920	0.750	0.826
	20	100	643	1359.907	0.496	0.650	0.930	0.730	0.818
	40	100	554	2967.990	0.502	0.594	0.972	0.760	0.853
Average			672	1722.334	0.504	0.576	0.933	0.743	0.827
BGWO	20	50	1214	407.027	0.456	0.234	0.930	0.698	0.797
	40	50	1150	510.434	0.544	0.274	0.950	0.710	0.813
	20	100	1202	775.281	0.479	0.241	0.940	0.710	0.809
	40	100	1178	1006.201	0.552	0.256	0.930	0.730	0.818
Average			1186	674.736	0.508	0.251	0.938	0.712	0.809
BPSO	20	50	813	271.140	0.510	0.487	0.880	0.680	0.767
	40	50	777	357.050	0.545	0.509	0.910	0.670	0.772
	20	100	817	476.030	0.501	0.484	0.890	0.680	0.771
	40	100	790	672.030	0.541	0.501	0.910	0.700	0.791
Average			799.25	444.063	0.524	0.495	0.898	0.683	0.775

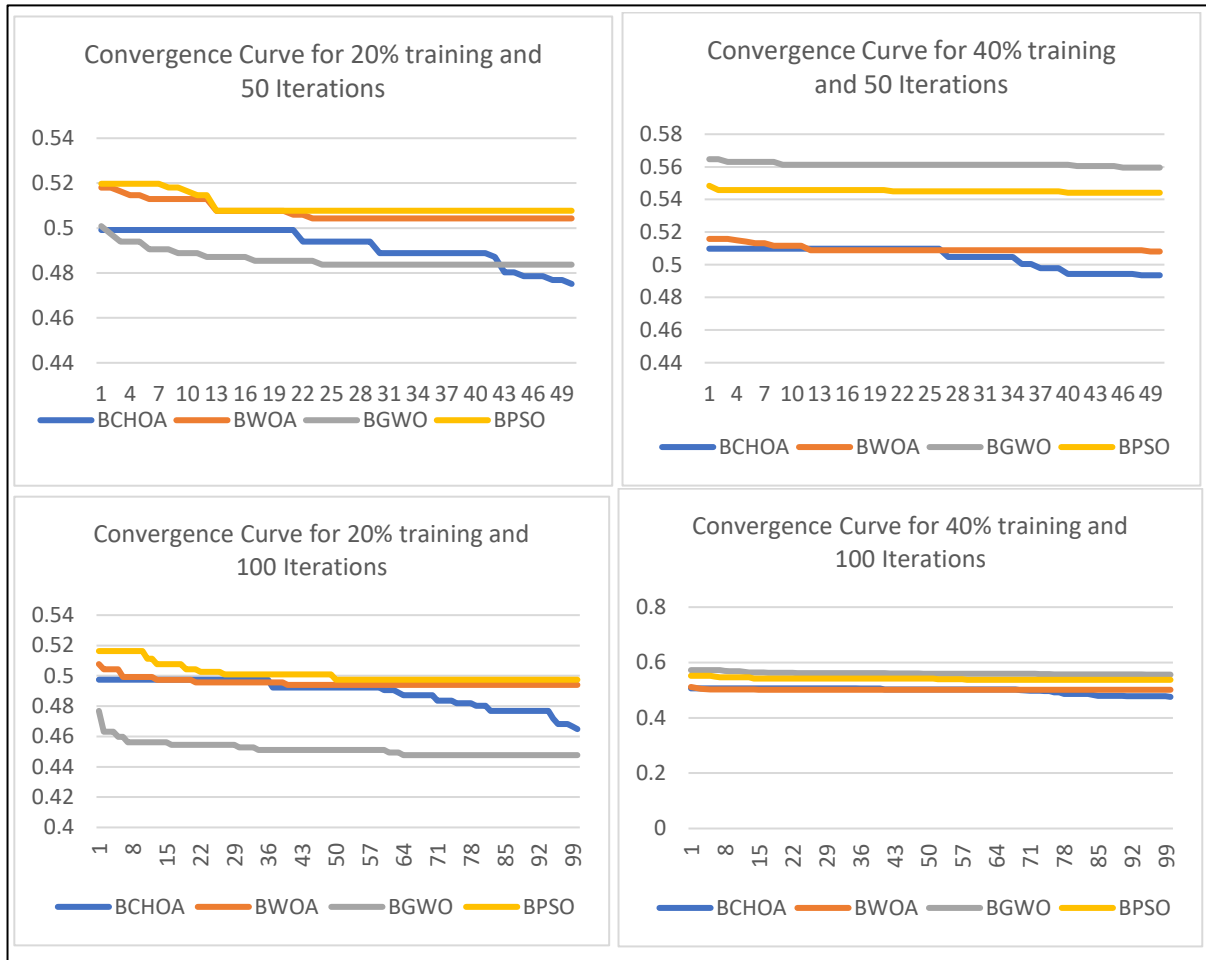





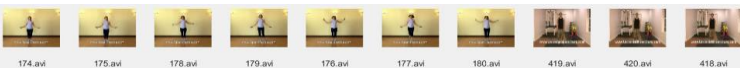



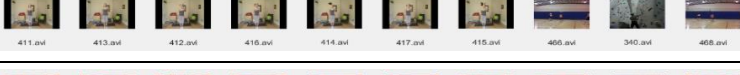


Figure 9. Convergence Curve Graphs for Different Training % and Iterations

Table 4. Comparison of Proposed CBVR System with Other Systems

Method	PR	RR	F1 Score	Average Computation time	Compression Ratio
CBVR using AlexNet [3]	0.96	0.48	0.714	1320.6 seconds on GPU	--
Intelligent Retrieval system [1]	--	--	--	--	0.9360
Proposed CBVR system	0.972	0.77	0.859	1205.9 seconds on i5 Processor	0.9701

Table 5. Video Retrieval Analysis for a few Example Videos

Query	Videos Retrieved	PR	RR	F1 Score	CR
	291.avi, 293.avi, 295.avi, 296.avi, 300.avi, 298.avi, 299.avi, 294.avi, 297.avi, 292.avi	1.00	1.00	1.00	0.98
	64.avi, 63.avi, 62.avi, 67.avi, 69.avi, 66.avi, 438.avi, 68.avi, 70.avi, 61.avi	1.00	0.90	0.95	0.99
	83.avi, 85.avi, 90.avi, 86.avi, 82.avi, 87.avi, 89.avi, 88.avi, 84.avi, 256.avi	1.00	0.90	0.95	0.93
	92.avi, 95.avi, 98.avi, 97.avi, 94.avi, 91.avi, 100.avi, 440.avi, 96.avi, 205.avi	1.00	0.80	0.89	0.91
	151.avi, 153.avi, 154.avi, 157.avi, 156.avi, 152.avi, 155.avi, 343.avi, 344.avi, 346.avi	1.00	0.70	0.82	0.86

		1.00	0.70	0.82	0.97
		1.00	0.70	0.82	0.96
		1.00	0.70	0.82	0.99
		1.00	0.70	0.82	0.91
		1.00	0.90	0.95	0.97

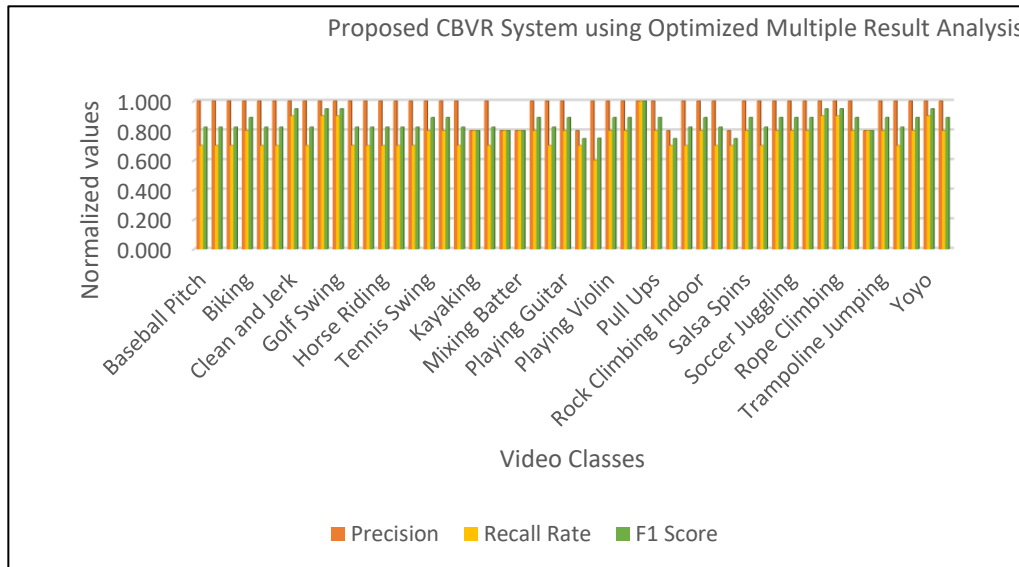


Figure 10. Graphical Analysis of the Proposed CBVR System for Different Video Classes

5. Conclusion

We demonstrated a novel Content-Based Video Retrieval (CBVR) system using Motion Boundary Histogram (MBH), RGB code, and GLCM texture features, along with feature selection optimization algorithms. We considered the MBH feature for VSBD and extracted key frames. From these key frames, we extracted multiple features that integrated both spatial and temporal information. Multiple feature extraction will, in turn, increase the feature vector size; here, feature selection optimization algorithms help reduce redundancy and enhance discrimination. We applied four metaheuristic optimization algorithms—BPSO, BGWO, BWOA, and BCHOA—to feature selection. The experimentation on a sample of the UCF50 dataset demonstrates that the proposed CBVR system using MBH and the BCHOA algorithm performs with higher computational efficiency and retrieval accuracy compared to other conventional works, with a CR of 0.9701, an F1 Score of 0.859, a PR of 0.972, an RR of 0.77, and an average computational time of 1205.9 seconds. These results validate the significance of combining multiple feature extraction techniques and optimization algorithms for feature selection. Future work will explore hybrid optimization approaches and multi-objective fitness functions to further improve the recall rate.

References

- [1] Sun, Maojin. "An Intelligent Retrieval Method for Audio and Video Content: Deep Learning Technology Based on Artificial Intelligence." *IEEE Access* 12 (2024): 123430-123446.
- [2] Adly, Ahmad Sedky, Islam Hegazy, Taha ElArif, and M. S. Abdelwahab. "Development of an Effective Bootleg Videos Retrieval System as a Part of Content-Based Video Search Engine." *Int. J. Comput.* 21, no. 2 (2022): 214-227.
- [3] Hussain, Altaf, Mehtab Ahmad, Tariq Hussain, and Ijaz Ullah. "Efficient Content Based Video Retrieval System by Applying AlexNet on Key Frames." *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* 11, no. 2 (2022): 207.
- [4] Yoon, Hyeok, and Ji-Hyeong Han. "Content-based Video Retrieval with Prototypes of Deep Features." *IEEE Access* 10 (2022): 30730-30742.
- [5] Thomas, Sinnu Susan, Sumana Gupta, and Venkatesh K. Subramanian. "Context Driven Optimized Perceptual Video Summarization and Retrieval." *IEEE transactions on circuits and systems for video technology* 29, no. 10 (2018): 3132-3145.
- [6] Palanivelu, Nitish, Nishita Jagdale, Nishith Dubey, and Bharati Jagtap. "Content Based Video Retrieval Using Deep Learning." (2024). 1-20,
- [7] Gornale, Shivanand S., Ashvini K. Babaleshwar, and Pravin L. Yannawar. "Analysis and Detection of Content Based Video Retrieval." *Int. J. Image, Graph. Signal Process* 11, no. 3 (2019): 43.
- [8] Liu, Yumeng, and Aina Sui. "Research on Feature Dimensionality Reduction in Content Based Public Cultural Video Retrieval." In *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, IEEE, 2018, 718-722.
- [9] TShami, Tareq M., Ayman A. El-Saleh, Mohammed Alswaitti, Qasem Al-Tashi, Mhd Amen Summakieh, and Seyedali Mirjalili. "Particle Swarm Optimization: A Comprehensive Survey." *Ieee Access* 10 (2022): 10031-10061.
- [10] Too, Jingwei, and Abdul Rahim Abdullah. "Opposition Based Competitive Grey Wolf Optimizer for EMG Feature Selection." *Evolutionary Intelligence* 14, no. 4 (2021): 1691-1705.
- [11] Too, Jingwei, Abdul Rahim Abdullah, Norhashimah Mohd Saad, Nursabillilah Mohd Ali, and Weihown Tee. "A New Competitive Binary Grey Wolf Optimizer to Solve the Feature Selection Problem in EMG Signals Classification." *Computers* 7, no. 4 (2018): 58.
- [12] Too, Jingwei, Majdi Mafarja, and Seyedali Mirjalili. "Spatial Bound Whale Optimization Algorithm: An Efficient High-Dimensional Feature Selection Approach." *Neural Computing and Applications* 33, no. 23 (2021): 16229-16250.
- [13] Ayeche, Farid, and Adel Alti. "Efficient Feature Selection in High Dimensional Data Based on Enhanced Binary Chimp Optimization Algorithms and Machine Learning." *Human-Centric Intelligent Systems* 3, no. 4 (2023): 558-587.

- [14] Chavate, Shrikant, and Ravi Mishra. "An Efficient Approach for Shot Boundary Detection in Presence of Illumination Effects using Fusion of Transforms." *International Journal of Engineering Trends and Technology* 70, no. 4 (2022): 418-432.
- [15] Kar, Tejaswini, Priyadarshi Kanungo, Sachi Nandan Mohanty, Sven Groppe, and Jinghua Groppe. "Video Shot-Boundary Detection: Issues, Challenges and Solutions." *Artificial Intelligence Review* 57, no. 4 (2024): 104.
- [16] Halim, Benoughidene Abdel and Titouna Faiza. "Shot Boundary Detection: Fundamental Concepts and Survey." *Conference on Innovative Trends in Computer Science* (2019).
- [17] Asha, D., and Y. Madhavee Latha. "Content-Based Video Shot Boundary Detection Using Multiple Haar Transform Features." In *Soft Computing and Signal Processing: Proceedings of ICSCSP 2018, Volume 1*, Singapore: Springer Singapore, 2019, 703-713.
- [18] Fan, Min, Qi Han, Xi Zhang, Yaling Liu, Huan Chen, and Yaqian Hu. "Human Action Recognition Based on Dense Sampling of Motion Boundary and Histogram of Motion Gradient." In *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, 2018, 1033-1038.
- [19] Ma, Zhihao, Tianyu Wang, Sheng Xu, Xinxing Mu, Qiang Wang, and Qingqiang Guo. "Moving Object Detection Based on Farneback Optical Flow." In *2023 42nd Chinese Control Conference (CCC)*, IEEE, 2023, 7350-7355.
- [20] Asha. D and Y. Madhavee Latha., "Content-Based VSBD Technique Using DWT-HOG Features and Particle Swarm Optimization for Thresholding," *2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT)*, Bidar, India, 2025, 1-5.
- [21] Benco, Miroslav, Patrik Kamencay, Martina Radilova, Robert Hudec, and Martin Sinko. "The Comparison of Color Texture Features Extraction Based on 1D GLCM with Deep Learning Methods." In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, 2020, 285-289.
- [22] Jia, Weikuan, Meili Sun, Jian Lian, and Sujuan Hou. "Feature Dimensionality Reduction: A Review." *Complex & Intelligent Systems* 8, no. 3 (2022): 2663-2693.
- [23] Chandrashekar, Girish, and Ferat Sahin. "A Survey on Feature Selection Methods." *Computers & electrical engineering* 40, no. 1 (2014): 16-28.
- [24] Reddy, Kishore K., and Mubarak Shah. "Recognizing 50 Human Action Categories of Web Videos." *Machine vision and applications* 24, no. 5 (2013): 971-981.