International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Deep Learning And Meta-Heuristic Optimization Approach For Content-Based Video Retrieval

Princy Matlani[1]*, Manish Shrivastava[2]

Submitted: 01/07/2024 **Revised:** 15/08/2024 **Accepted:** 25/08/2024

Abstract : Deep learning-based feature extraction has become a popular approach for Content-Based Video Retrieval (CBVR) due to its ability to capture the complex and discriminative features of the video data.Existing works in CBVR using deep learning-based feature extraction have focused on developing various deep learning models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), etc. for feature extraction from video frames. These models have been trained on large datasets to learn the underlying patterns and features in the video data.Deep learning models can be computationally expensive, especially for large and complex video datasets. This can limit the scalability and real-time performance of CBVR systems. Therefore, in this reserch work a novel CBVR model is intrdoced with the assistance acquired from the deep learning and meta-heuristic optimization model. The proposed model includes five major phases: Video pre-processing, Feature extraction, Feature selection, Video representation, Similarity computation. In the Video pre-processing phase, the collected raw video is pre-processed via video-to-frame conversion, color correction via histogram equalization and denoising via median filtering. Then, visual and temporal features are extracted from the video frames using the new AlexNet and Recurrent Neural Network (RNN), respectively. Principal components of features are selected optimally via the new Self-Improved Snow Leopard Optimization Algorithm (SI-SLO). The proposed SI-SLO model is an extended version of the standard Snow Leopard Optimization (SLO). In the Video representation stage, the selected features are used to represent the video. Then, in the Similarity computation phase, the similarity between two videos is computed using the video representations.The videos can be ranked based on their similarity scores and displayed to the user. Theproposed model has been implemented in MATLAB. The evaluation has been made in terms of accuracy, precision, sensitivity, specificity as well.

Keywords-CBVR (Content-Based Video Retrieval); Deep Learning; Meta-heuristic Optimization; Feature Selection; Similarity Computation

Introduction

The term "content" is not limited to text alone; it can also refer to the hues, textures, and shapes of objects [1]. The segmentation of moving objects into shots is the first stage of " CBVR" In its simplest form, a shot is an indexing unit made up of an image sequence. Key-frames are taken from this sequence of images and give an abstract idea of the incredibly informative visual content that is present in the video, facilitating faster video browsing [2]. The use of CBVR [3] allows for the extraction of features from the videos and the retrieval of desired videos from a large collection of videos. Color and texture are characteristics of a video. The contextual information in the videos is better-described thanks to these features [4]. When the database is small and constrained, text-based searches and retrievals are beneficial. However, content-based retrieval is crucial when it comes to large databases or data warehouses. The CBVR system, therefore, requires the system to be made more reliable, robust, and automated. The CBVR uses visual elements like color, texture, edge, shape, and motion to retrieve related videos from the database [5]. A significant area of research and development now centers on the search for solutions to issues relating to video retrieval. One of the most significant and current areas of image processing research is the retrieval and analysis of content-based videos [6]. For multiuser video search and browsing systems that are helpful in web applications, CBVR is used. [7]. As multimedia technologies and information highways have developed, more and more video data have been recorded, created, and stored. All of these data, however, are hardly usable without the proper methods that can increase the video content's accessibility. However, due to the continuity differences between multimedia and textual data, traditional database technology is unable to handle multimedia information. In light of this, content-based access and retrieval are suitable solutions [8]. CBVR systems have become effective resources for locating particular content in vast video collections that are constantly expanding [9]. The core of most video retrieval tools is a retrieval engine, which returns a ranked video list based on a variety of features [10].

*1*Department of Computer Science & Engineering, Guru Ghasidas University, Bilaspur, CG, India*

2Department of Computer Science & Engineering, Guru Ghasidas University, Bilaspur, CG, India

The major contribution of this reserch work is:

- The use of AlexNet and Recurrent Neural Network (RNN) for visual and temporal feature extraction.
- The use of the Self-Improved Snow Leopard Optimization Algorithm (SI-SLO) for optimizing the selection of principal components in the feature selection phase using PCA.
- The propsoed SI-SLO model is an extended version of the standrd Sea Lion Optimization Algorithm (SLO).

Literature Review

In 2022, Hussain *et al.* [11] suggested using the AlexNet model of the CNN on the keyframes system to create an effective content-based video retrieval system. Choose the keyframes from the video first. After that, the color histogram was computed. The characteristics of the color histogram are then contrasted and examined for CBVR. The extracted features from the frames are combined to store in the feature vector of the proposed system, which was based on the CNN AlexNet model and color histogram. Based on MATLAB simulation results, the suggested method had tested on the benchmark dataset UCF101, which contains 13320 videos from 101 action categories.

In 2019, Shen *et al.* [12] proposed a powerful technique for content-based video retrieval called similarity-preserving deep temporal hashing. Through the use of stacked GRU, it can extract the spatial-temporal characteristics of videos and produce binary codes. It integrates video temporal modeling and learning to hash into one stage to enable maximum retention of video information. The network was effectively trained to maintain intraclass similarity and interclass separability through the introduction of a discriminative objective function. Three video datasets were used in experiments to see how well the method performed in comparison to other video hashing techniques.

In 2020, Bekhet*et al.* [13] analyzed the most popular video retrieval similarity measures. A multifaceted analysis was used to consolidate the results, including confusion matrices, retrieval curves, and multiple difficult video datasets. Also looked into the usefulness of the standard similarity metrics from the perspective of video retrieval.

In 2021, Saoudi*et al.* [14] suggested using a CBVR system that pulls similar videos from a sizable multimedia dataset. That method describes the visual content with vector motion-based signatures and extracts key frames for quick browsing and effective video indexing using machine learning techniques. The proposed method had tested on both a single machine and a real-time distributed cluster to assess its real-time performance, particularly when the quantity and size of videos are high. Different benchmark actions were used in the experiments.

In 2018, Tarigan*et al.* [15] suggested implementing SURF for CBVR. The application searches through a collection of videos using an image as a query and selects any that have frames that correspond to the image. The performance of the algorithm was what to evaluate. Recall, precision, and running time are the three factors that make up the performance measurement.

3. Research Gaps

Author	Process/ Aim	Research Gaps
Sajjad *et al.* [1]	integrating rotationally invariant texture features with salient colors for image representation	• Analyzing predefined structures in other directions is not done.
Brindha*et al.* [2]	Using a multi-stage ESN-SVM classifier, bridging the semantic divide between high-level and low-level features in content-based video retrieval.	• There are no additional optimization methods for ESN.
Kaliaperumal*et al.* [3]	An Indexing and Combinational Features-Based Content-Based Retrieval Model for Distributed Video Objects	• The analysis of the visual imagery using DL is not included.
Gornale*et al.* [5]	Investigation and CBVR Detection	• The retrieval and recognition of the sign boards are not tested.
Münzer*et al.* [9]	combine a straightforward mobile tool that is designed for quick human perception in a sequential manner	• It is not incorporated to use multiple tablet users in the collaboration team and distribute the reranked video list across multiple instances of the tablet tool
Hussain *et al.* [11]	Utilizing AlexNet on Key Frames to Create an Effective Content-Based Video Retrieval System	• To achieve better results, better feature analysis methodologies can be used.
Shen *et al.* [12]	Retrieval of Videos Using Deep Temporal Hashing with Similarity Preservation	• Event prediction and multi-model retrieval are not expanded.
Saoudi*et al.* [14]	An extensive dataset distributed CBVR system	• The performance of processing time is not enhanced.

3.Proposed Methodology

The proposed methodology of Content-Based Video Retrieval (CBVR) The specific implementation of this methodology can vary depending on the requirements and applications, but the overall goal is to retrieve videos that are most relevant to the query based on their content.

This paper proposes a Content based video Retrieval. The projected model includes six major phases:
(a) pre-processing, (b)Feature Extraction(c)Feature Selection(d)Video representation(e) Similarity computation(f)Video Retrieval.

The proposed CBVR model can be summarized as follows:
Video Pre-processing: The raw video is converted into individual frames, and color correction and denoising are performed on the frames.
Feature Extraction: Visual and temporal features are extracted from the video frames using the new AlexNet and Recurrent Neural Network (RNN) models, respectively.
Feature Selection: The extracted features are further processed to identify the most relevant and discriminative features using optimized PCA, whose principal components are selected optimally via the new SI-SLO algorithm.

Video Representation: The selected features are used to represent the video.
Similarity Computation: The similarity between two videos is computed using the video representations, using a cosine similarity-based approach.
Video Retrieval: The video retrieval system uses the computed similarity scores to retrieve relevant videos based on the query, and the videos are ranked based on their similarity scores and displayed to the user.
Evaluation: The proposed model has been implemented in MATLAB and evaluated in terms of F1-score, mAp as well.

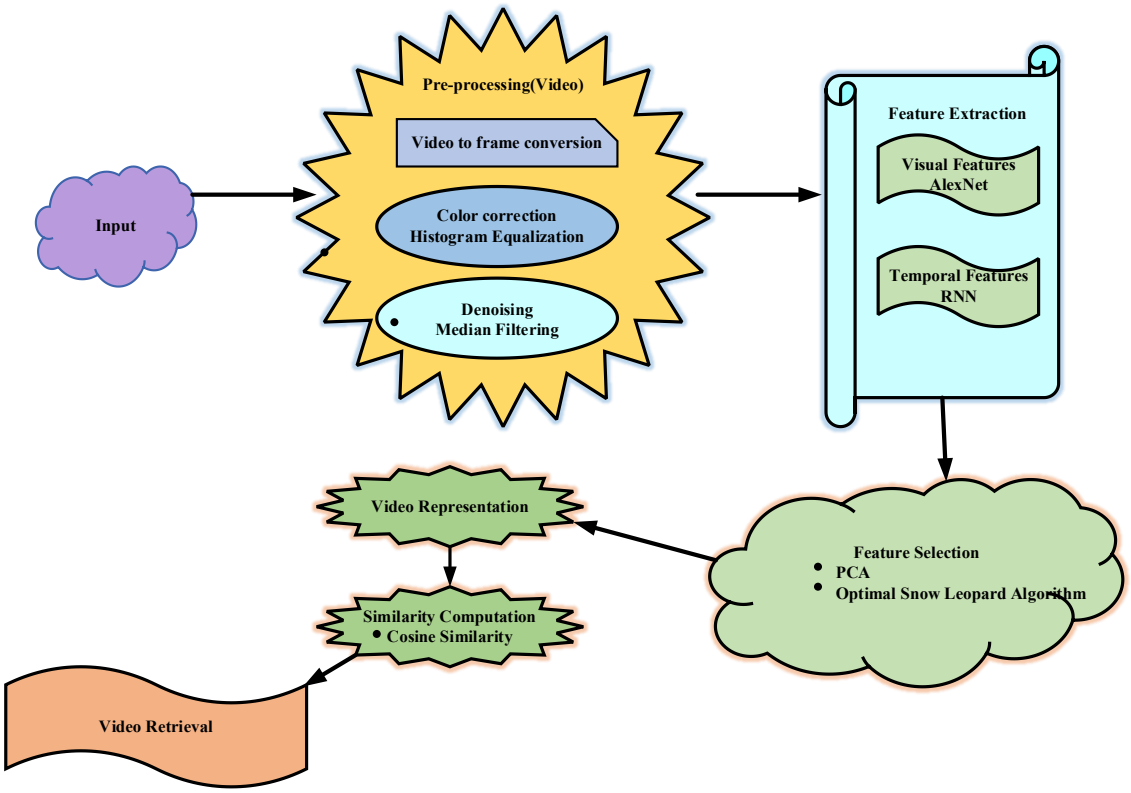


Figure 1: Proposed Model

3.1. Video Pre-processing

In this research work, Video Pre-processing is done using Video-to- frame Conversion, Color Correction via Histogram Equalization, and Denoising via Medium Filtering. Content-based video retrieval (CBVR) is a field of computer vision that deals with searching and retrieving video clips based on their visual content. Pre-processing plays an important role in CBVR as it can significantly affect the accuracy and efficiency of the retrieval process.

3.1.1. Video-to-frame Conversion

Video-to-frame conversion is a process in CBVR where a video clip is divided into individual frames, which can then be processed and analyzed separately. The goal of this process is to simplify the video data, making it easier to extract visual features and representations that can be used for comparison and similarity matching in the retrieval process. During video-to-frame conversion, the frames are typically sampled at regular intervals or keyframes to ensure that important information is captured. The selected frames can then be processed and analyzed using computer vision algorithms to extract features such as color

histograms, edge maps, or deep learning-based features that can be used to represent the visual content of the video. The extracted features can then be compared to a database of features from other video clips to find the most similar videos based on their visual content. This process is a key component of CBVR, as it allows for fast and efficient search and retrieval of relevant video clips based on their visual content.

3.1.2. Color Correction via Histogram Equalization

Color correction via histogram equalization is a technique used in CBVR to adjust the color distribution of a video frame to make it more visually appealing or to remove color casts. The goal of histogram equalization is to adjust the intensity levels of the colors in the video so that they are spread evenly across the color spectrum. Look at a digital image that has a variety of grayscale tones $[0, A-1]$. The Probability Distribution function of the image can be calculated using an Eqn. (1).

$$B(a_c) = \frac{m_c}{Z} \qquad c = 0, \ldots., A-1 \qquad (1)$$

Where$a_c$is the cth gray level and $m_c$is the numberof pixels in the image having a gray level q_l.Cumulative Distribution Function (CDF) can also becomputed in Eqn. (2):

$$D(a_c) = \sum_{n=0}^{n=c} B(a_n) \qquad (2)$$

$$c = 0, \ldots., A-1, 0 \leq D(a_c) \leq 1$$

HE appropriates gray level $Y_c$to gray level $a_c$of the input image using Eqn. (2).

$$Y_c = (A-1) * D(a_c) \qquad (3)$$

With the standard histogram equalization method, the grey level Y_c changes can be calculated in Eq. (4):

$$\Delta Y_c = (A-1) * B(a_c) \qquad (4)$$

Eq. (4) means that the distance between $Y_c$and $Y_c + 1$has a direct relation with the PDF of the input image at the gray level q_l.Eq. (2) quantization operation and summarising characteristics result in adverse HE effects.

3.1.3. Denoising via Median Filtering

A non-linear denoising technique called the median filter is used to take out salt-and-pepper noise from an image or a signal. It functions by substituting a pixel's value with the median value of the pixels in its defined neighbourhood. The median, which is robust to outliers or extreme values, is the middle value among a group of values. The median filter in an image replaces a pixel's value with the median value of the pixels next to it in a sliding window. Each pixel in the image goes through this procedure once more to produce a smoothed-out, denoised image.

$$\widehat{pp}(z, bn) = mdn\{fh(s, a)\} \quad (z, bn) \epsilon EF_{zb} \ (5)$$

3.2 Feature Extraction

In this research, the features extracted are Visual features and Temporal features. Visual and temporal features are extracted from the video frames using the new AlexNet and Recurrent Neural Network (RNN) models, respectively. They are explained as follows.

3.2.1 Visual features

Visual features are a set of attributes extracted from an image that help computers understand the content of an image and describe it using numerical values. Examples of visual features include color histograms, edge detection, texture analysis, and object detection. These features are critical components in computer vision and image processing applications, such as object recognition and image classification.

3.2.1.1 AlexNet

AlexNet is a deep learning convolutional neural network that won the ImageNet 2012 classification competition and revolutionized computer vision.Eight layers make up the architecture: three fully-connected layers and five convolutional layers. Fig. 2 illustrates the AlexNet architecture. In the first convolutional layer, which uses 96 different 1111-sized receptive filters, convolution, and max pooling are performed along with local response normalization (LRN). 33 filters with a stride size of 2 are used to perform the max pooling operations. In the second layer, identical operations are carried out using 55 filters. In the third, fourth, and fifth convolutional layers, which have 384, 384, and 296 feature maps, respectively, 33 filters are applied.

A Softmax layer is used at the end after two fully connected (FC) layers with dropout are used. For this model, parallel training is performed on two networks with comparable architectures and the same number of feature maps. Local Response Normalization is one of two novel ideas (LRN).With dropout and a Softmax layer at the very end, two fully connected (FC) layers are utilized. For this model, two networks with comparable structural similarities and an equal number of feature maps are trained concurrently. In this network, two novel ideas—Local Response Normalization (LRN) and dropout—are presented. Two methods of applying LRN are possible: first, on single channels or feature maps, where a NN patch is chosen from the same feature map and normalized based on the neighborhood values Second, LRN can be used with feature maps or channels.

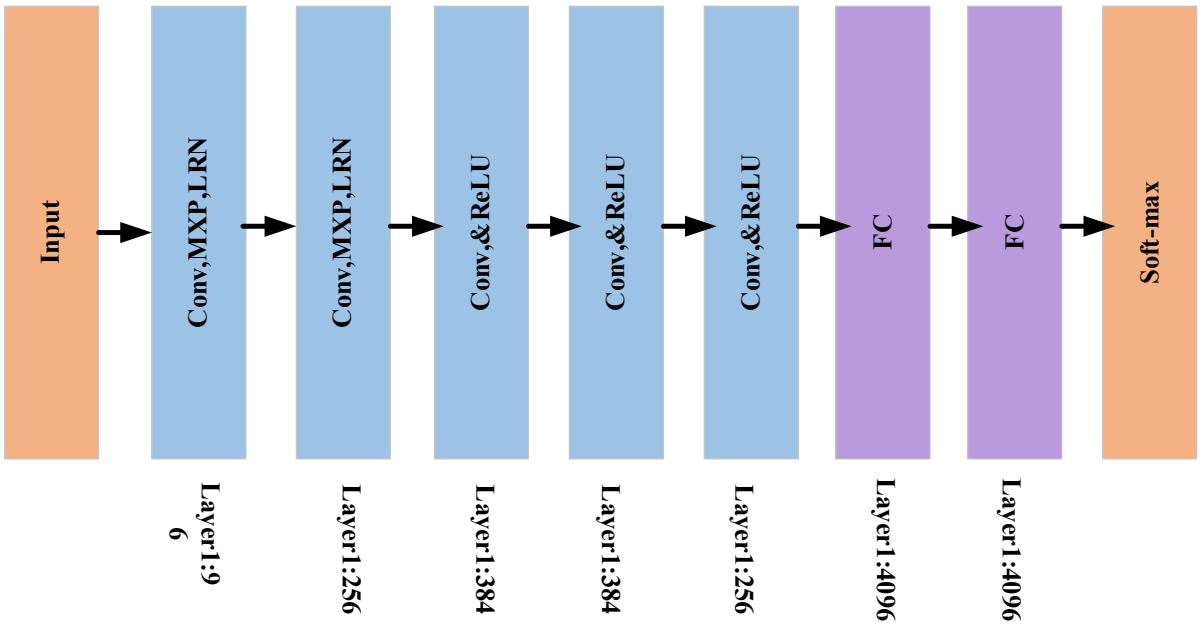


Figure2: AlexNet Architecture

3.2.2 Temporal Features

Temporal features are attributes that describe the dynamic aspects of a video, including motion, changes in appearance, and motion patterns. They help computers analyze and understand the motion in a video and detect events or actions. Examples of temporal features include optical flow, feature tracking, motion estimation, and action recognition. These features are important in video analysis and processing applications, such as video surveillance, sports analysis, and human-computer interaction.

3.2.2.1 Recurrent Neural Network

Recurrent neural networks (RNNs) are a type of neural network in which the results of one step are fed into the current step as input. Traditional neural networks have inputs and outputs that are independent of one another, but there is a need to remember the previous words in situations where it is necessary to predict the next word in a sentence. As a result, RNN was developed, which utilized a Hidden Layer to resolve this problem. The Hidden state, which retains some information about a sequence, is the primary and most significant characteristic of RNNs.

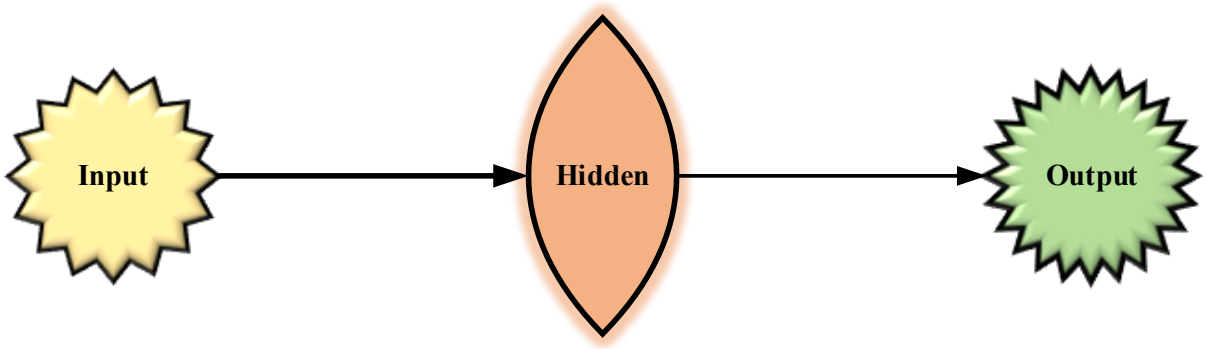


Figure3: Recurrent Neural Network Architecture

RNNs have a "memory" that retains all data related to calculations. It performs the same task on all of the inputs or hidden layers to produce the output, using the same parameters for each input. In contrast to other neural networks, this reduces the complexity of the parameter set.

$$p_s = e(p_{s-1}, y_s) \quad (6)$$

As per Eqn. (6)p_s is the current state, p_{s-1} is the previous state and y_s is the input state.To calculate the output the following Eqn.(7) is implemented.

$$z_s = U_{pz} p_s \quad (7)$$

As per Eqn.(7) z_s is the output and U_{pz} is the weight at the output layer.

3.3. Feature Selection

In this research Work, Feature selection is carried out using Optimized PCA and optimized SI-SLO. The principal components of PCA is optimized via the new SI-SLO algorithm.

3.3.1 Principal Component Analysis (PCA)

The variance maximization concept is applied in PCA to minimize the number of variables, eliminate irrelevant information, and maximize the useful information by substituting the rows or columns of the

original data matrix with a new set of linearly independent and orthogonal vectors. The PCA is resolved in this manner.

1) Averaging the original data's processing values $a_c^m(z)$, then calculate its covariance matrix V_{o*o}. The mathematical model is shown in Eq. (8).

$$cov(a,b) = \frac{1}{l}\sum_{c=1}^{l}(a_c - \bar{a})(b_c - \bar{b}), 1 \le a \le o, 1 \le b \le o \quad (8)$$

where l is the number of features, a and b are specific features, and o is the number of samples.

2) The covariance matrix V is decomposed into its characteristic roots, yielding characteristic roots Λ_{o*o} and U_{o*o}. The size of each characteristic root corresponds to the amount of information that each principal component contains. These principal components are selected optimally via the new SI-SLO algorithm. The mathematical model is shown in Eq. (9)

$$V_{o*o} = U_{o*o}\Lambda_{o*o}U_{o*o} \quad (9)$$

3)Find the projection M_{l*o} of the original data $a_c^m(z)$ in the new coordinate system. The mathematical model is shown in Eq. (10).

$$M_{l*o} = X_{l*o}U_{l*o} \quad (10)$$

4) Contribution over time. Each main component's characteristic root size and the cumulative contribution rate of the previous fk main components show how much information is included in each component, respectively. The mathematical model is shown in Eq. (11).

$$pre_{fk} = \sum_{c=1}^{fk}\lambda_c / \sum_{c=1}^{o}\lambda_c \quad (11)$$

λ_c is the cth characteristic root of the solution.

5)Pick the appropriate cumulative contribution prior, then transform the prior d main component M_{l*d} into new data, and use this alternative raw data $a_c^m(z)$ to classify patterns. (In normal: $d < o$).

3.3.2 SI-SLO

Snow Leopard optimization is a meta-heuristic optimization algorithm inspired by the behavior of snow leopards. It is a population-based algorithm that can be used to solve optimization problems in various fields, including machine learning, engineering, and computer science. In the context of the proposed CBVR model, Snow Leopard optimization can be used to optimize the number of principal components selected in the feature selection phase using PCA. The Self-Improved Snow Leopard Optimization (SI-SLO) algorithm is an enhanced version of the standard Snow Leopard Optimization (SLO) algorithm. The major advantage of SI-SLO over SLO is that it has been improved to better handle the optimization problem and to produce better optimization results. This improvement is achieved by incorporating additional components such as self-improvement mechanism, dynamic parameters adaptation, and randomness. These components allow the SI-SLO algorithm to better explore the search space and avoid getting trapped in local optima. The result is a more efficient optimization algorithm that can produce better results in less time compared to the standard SLO algorithm.

- **Mathematical modeling**

Each snow leopard is a part of the algorithm population in the proposed SI-SLO. Members of the SI-SLO act as search agents, including a certain number of snow leopards. Members of the population are identified using a matrix known as the population matrix in population-based optimization algorithms. The number of columns in the population matrix corresponds to the number of variables in the optimization problem, and the number of rows in this matrix corresponds to the number of population members. Eqn. (12) specifies the population matrix as a matrix representation.

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_j \\ \vdots \\ Y_K \end{bmatrix} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,q} & \cdots & y_{1,r} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{j,1} & \cdots & y_{j,q} & \cdots & y_{j,r} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{K,1} & \cdots & y_{K,q} & \cdots & y_{k,r} \end{bmatrix}_{K\times r} \quad (12)$$

As per Eqn.(12)$Y$is the snow leopard population,Y_j is the jth snow leopard. $y_{jp}$is the value for pth problem variable suggested, K is the number of snow leopards in the algorithm population and r is the number of problem variables? Using Eqn. (13), a vector specifies the values of the objective function.

$$E = \begin{bmatrix} e_1 \\ \vdots \\ e_j \\ \vdots \\ e_K \end{bmatrix}_{K\times 1} = \begin{bmatrix} e(Y_1) \\ \vdots \\ e(Y_j) \\ \vdots \\ e(Y_K) \end{bmatrix}_{K\times 1} \quad (13)$$

E is the objective function vector and Ej is the objective function value.

Phase1: **Travel routes and movement**

Equations (14) to (16)are used in the mathematical modeling of this stage of the proposed SI-SLO...

$$y_{j,q}^{C1} = y_{j,q} + d \times \left(y_{b,q} - J \times y_{j,q}\right) \times sign(E_j - E_b), b \epsilon 1,2,3,\ldots\ldots K, q = 1,2,3\ldots . r \quad (14)$$

$$Y_j = \begin{cases} Y_j^{C1}, & E_j^{C1} < E_j \\ Y_j, & else \end{cases} \quad (15)$$

$$J = round(1 + d) \quad (16)$$

The new value for qth problem variable is$y_{j,q}^{C1}$ and the random number in an interval of [0,1] is $m.b$ is the row number of selected snow leopards and the objective function value is Y_j^{C1}.

Phase 2: Levy Filgit based Hunting (proposed)

Equations (17)– (19) are used to simulate the snow leopards' natural hunting behavior (19). The location of the prey for it with the snow leopard is given by equation (17). Equation (18) represents the approach of a snow leopard to its prey. In this reserch work, the levy flight meachnism$Ley(d)$is newly introduced

within SLO model. In the Snow Leopard Optimization (SLO) algorithm, the Levy flight based hunting mechanism is one of the key components. The Levy flight based hunting mechanism is used to search for the global optimum of the optimization problem. The basic idea behind the Levy flight based hunting mechanism is to imitate the random hunting behavior of the snow leopard in its natural habitat. The snow leopard is known to move randomly in its habitat, searching for its prey. In the SLO algorithm, this random search behavior is mimicked to search for the global optimum of the optimization problem. The Levy flight-based hunting mechanism in SLO algorithm uses the Levy distribution to generate random search points in the solution space. These search points are used to update the current best solution, leading to convergence towards the global optimum.

$$C_{j,q} = y_{j,q} d = 1,2,3 \dots r \quad (17)$$

$$y_{j,q}^{C1} = y_{j,q} + d \times \Big(\big(c_{j,q} - y_{j,q}\big) \times C + \big(c_{j,q} - 2 \times y_{j,q}\big) \times (1 - C) \Big) * Ley(d) \times sign(E_j - E_c) \quad (18)$$

$$Y_j = \begin{cases} Y_j^{C2}, E_j^{C2} < E_j \\ Y_j, \ else \end{cases} \quad (19)$$

$c_{j,q}$ is the qth dimension of the location of prey considered for the jth snow leopard, E_c is the objective function value based on the location of prey, $y_{j,q}^{C1}$ is the new value for the qth problem variable obtained by the jth snow leopard based on phase 2.

Phase 3: Reproduction

In this stage, the algorithm's population is increased by a factor of half based on the natural reproduction patterns of snow leopards.The Eqn.(20) is used to create a mathematical model of the snow leopard reproduction process based on the aforementioned ideas .

$$G_o = \frac{y_o + Y_{K-o+1}}{2} \quad , o = 1,2,3, \dots \dots, \frac{K}{2} \quad (20)$$

Phase 4: Mortality

Everything alive is constantly in danger of dying. Even though reproduction boosts the population of snow leopards, mortality and losses cause the population to remain constant throughout the algorithm's replication. The value of the objective function serves as the SI-SLO's snow leopard mortality criterion. As a result, snow leopards with weaker objective functions are more likely to perish. Additionally, due to their poor objective functionality, some newborn cubs may perish.

3.4 Video Representation

Video representation refers to the process of transforming raw video data into a compact, meaningful, and computationally manageable form that can be used for further processing and analysis. This may involve converting the video into a sequence of still images, extracting relevant features such as color, texture, and motion, or encoding the video into a compact format such as H.264. The goal of video representation is to preserve the essential information in the video while reducing the size and complexity of the data, making it more suitable for storage, transmission, and analysis.

3.5 Similarity Computation

Similarity computation refers to the process of determining the degree of similarity between two or more objects. This can be applied to various types of data, including text, images, audio, or numerical values. The result of similarity computation is typically a numerical value that indicates the level of similarity between the objects being compared, with higher values indicating greater similarity. This is a fundamental task in many fields, including computer vision, natural language processing, and recommendation systems.

3.5.1 Cosine similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It is calculated as the cosine of the angle between the vectors, which can range from -1 to 1. A cosine similarity of 1 means that the vectors are identical, a similarity of 0 means that the vectors are orthogonal (perpendicular) to each other, and a similarity of -1 means that the vectors are opposed. Cosine similarity is commonly used in natural language processing and information retrieval to compare the similarity of documents or text based on their term frequency vectors.

3.6 Video Retrieval

Video retrieval refers to the process of searching and retrieving video content from a large database of videos based on certain criteria. This can be performed based on low-level features such as color, texture, and motion, or high-level features such as the content of the video, such as people, objects, and scenes. The goal of video retrieval is to find the most relevant videos for a given query, such as a keyword, image, or audio clip, and to present them to the user in an ordered list based on their similarity or relevance. Video retrieval is a key task in many applications, such as video search engines, surveillance systems, and multimedia content management.

4. Result and Discussion

4.1 Experimental Setup

The proposed model has been implemented in MATLAB. The proposed model is the data collected from: UCF101. UCF101 is a dataset of videos that covers a wide range of human actions and activities and is widely used for video classification tasks. The proposed model is validated over the existing models in tersm of F-measure, mAP as well.

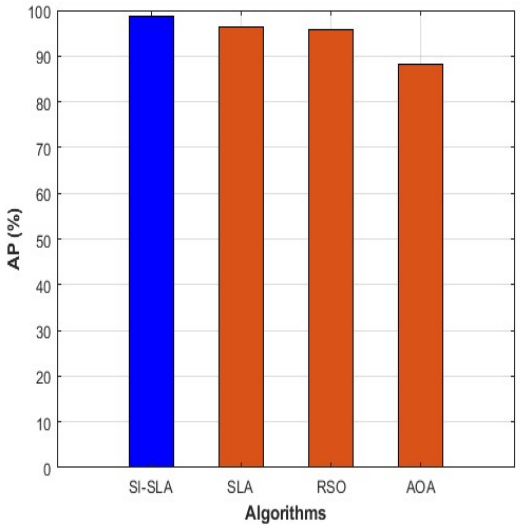
4.2 Overall Performance Analysis of the Proposed Model

According to the table I, all of the current models with varying training percentages are compared to the proposed model.

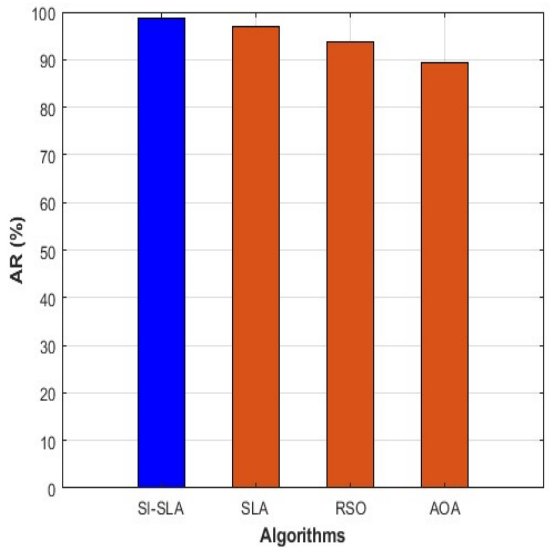
Table I: Overall performance analysis of the proposed model

	AP	AR	F-measure	mAP	Retrieval Time(sec)
Proposed	98.76	98.77	98.76	0.69	131
SLA	96.36	96.9	95.63	0.52	429
RSO	95.83	93.75	94.78	0.49	419
AOA	88.14	89.39	88.76	0.47	553

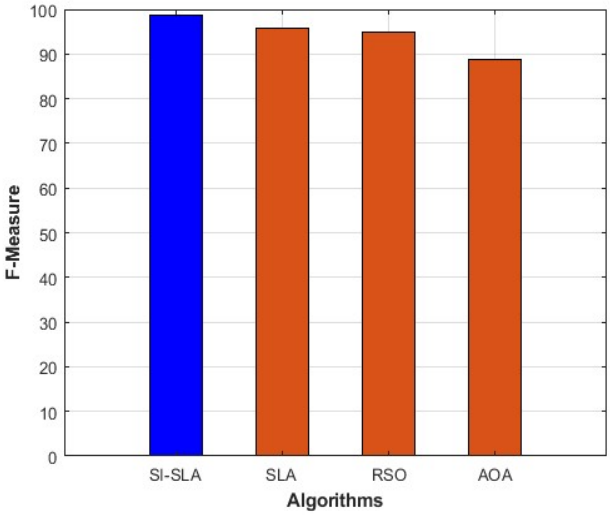
The proposed method achieved 98.76% accuracy in AP, 98.77% in AR, and 98.76% F-measure with anmAP of 0.69. It took 131 seconds for retrieval time.The SLA method had an accuracy of 96.36% in AP, 96.9% in AR, and 95.63% F-measure with anmAP of 0.52 and took 429 seconds for retrieval time.The RSO method had 95.83% accuracy in AP, 93.75% in AR, and 94.78% F-measure with anmAP of 0.49 and took 419 seconds for retrieval time.The AOA method had 88.14% accuracy in AP, 89.39% in AR, and 88.76% F-measure with anmAP of 0.47 and took 553 seconds for retrieval time.



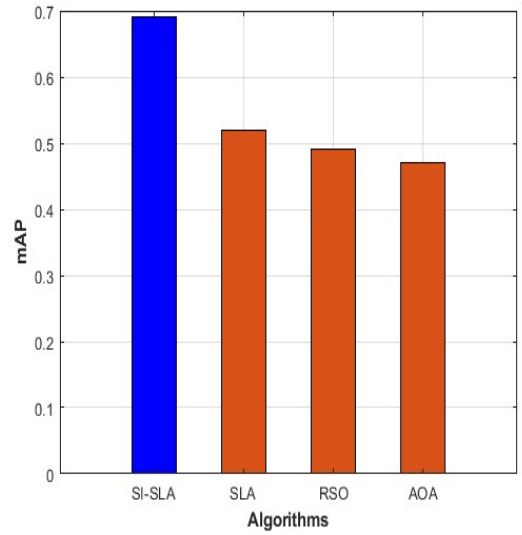
(a)



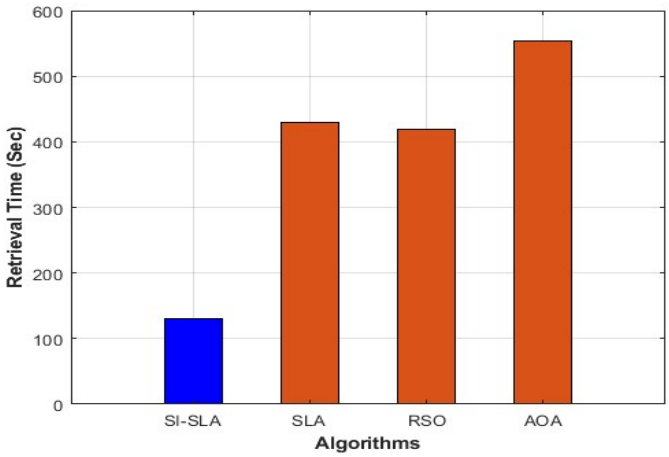
(b)



(c)

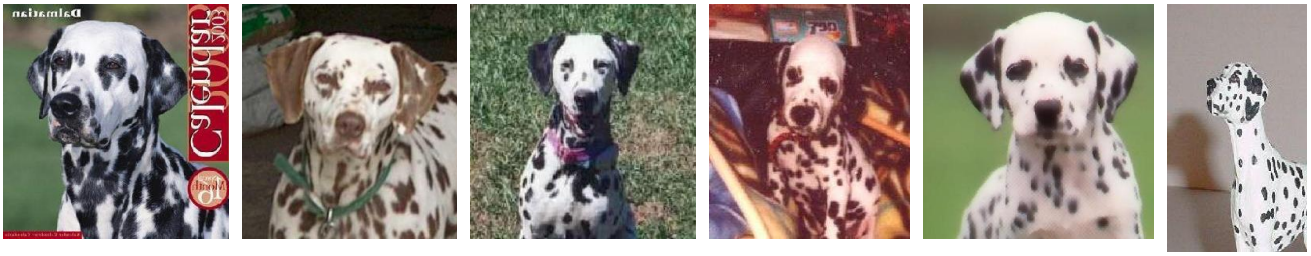


(d)



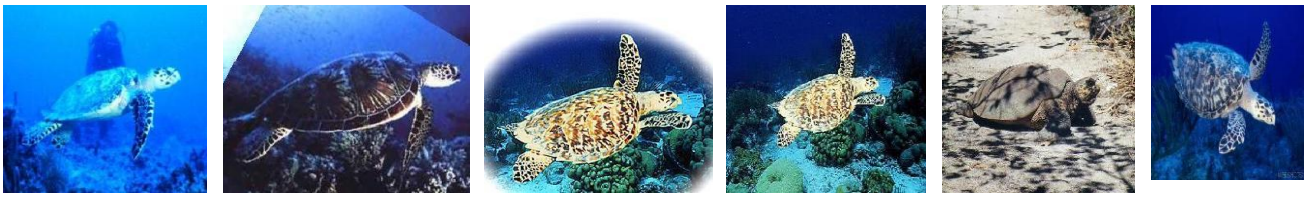
(d)

Figure 4: Overall performance analysis of (a) AP,(b) AR,(c) F-Measure, (d) mAP, (e) Retrieval time



(i) (a) (b) (c) (d) (e)

Figure 5: Query image(i) and (a)(b)(c)(d) are sample retrieval images



(ii) (a) (b) (c) (d) (e)

Figure 6: Query images (ii) and (a), (b), (c), (d) and (e) are sample retrieval images

(iii) (a) (b) (c) (d) (e)

Figure 7: (iii) Query image and (a),(b), (c),(d) and (e) are sample retrieval images

(iv) (a) (b) (c) (d) (e)

Figure 8: (iv) Query image and (a),(b), (c), (d) and (e) are sample retrieval images

Figures 5,6,7,8 are the query image and the sample retrieval images of the proposed model.

5. Consluion

In conclusion, the research presented a novel Content-Based Video Retrieval (CBVR) model that leveraged the advancements in deep learning and meta-heuristic optimization. The proposed model went through five main phases, starting with Video pre-processing, where the raw video was transformed into frames, underwent color correction and denoising. Then, the visual and temporal features were extracted from the video frames using AlexNet and Recurrent Neural Network (RNN) respectively. The features were further processed to select the most relevant and discriminative features through optimized Principal Component Analysis (PCA), with the help of the Self-Improved Snow Leopard Optimization Algorithm (SI-SLO). The video representations were then created using the selected features, and the similarity between two videos was computed using cosine similarity. Finally, the video retrieval system retrieved relevant videos based on the query and ranked them based on their similarity scores. The proposed model was implemented in MATLAB and evaluated in terms of F1-score, mAp as well.

Nomenclature

ABBREVIATION	DESCRIPTION
CBVR	Content-based Video Retrieval
CNN	Convolutional Neural Network
GRU	Gated Recurrent Units
SURF	Speeded-Up Robust Features
ANN	Artificial Neural Networks
DL	Deep Learning
CBVIR	Content-Based Video Indexing and Retrieval

References

[1] Sajjad, M., Ullah, A., Ahmad, J., Abbas, N., Rho, S. and Baik, S.W., 2018. Integrating salient colors with rotational invariant texture features for image representation in retrieval systems. Multimedia Tools and Applications, 77(4), pp.4769-4789.

[2] Brindha, N. and Visalakshi, P., 2017. Bridging semantic gap between high-level and low-level features in content-based video retrieval using multi-stage ESN–SVM classifier. Sādhanā, 42(1), pp.1-10.

[3] Kaliaperumal, N., Das, A. and Balakrishnan, V., A Content-Based Retrieval Model with Combinational Features and Indexing for Distributed Video Objects.

[4] Patel, B.V. and Meshram, B.B., 2012. Content-based video retrieval systems. arXiv preprint arXiv:1205.1641.

[5] Gornale, S.S., Babaleshwar, A.K. and Yannawar, P.L., 2019. Analysis and detection of content-based video retrieval. Int. J. Image, Graph. Signal Process, 11(3), p.43.

[6] Chun, Y.D., Kim, N.C. and Jang, I.H., 2008. Content-based image retrieval using multiresolution color and texture features. IEEE Transactions on Multimedia, 10(6), pp.1073-1084.

[7] Wankhede, V.A. and Mohod, P.S., 2012. A Review on Content-Based Image Retrieval from Videos using Self Learning Object Dictionary. International Journal of Science and Research.

[8] Raj, B.V. and Kandoi, C., 2020. Content-based Video Retrieval.

[9] Münzer, B., Primus, M.J., Hudelist, M., Beecks, C., Hürst, W. and Schoeffmann, K., 2017, July. When content-based video retrieval and human computation unite: Towards effective collaborative video search. In 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (pp. 214-219). IEEE.

[10] Lokoč, J., Bailer, W., Schoeffmann, K., Münzer, B. and Awad, G., 2018. On influential trends in interactive video retrieval: video browser showdown 2015–2017. IEEE Transactions on Multimedia, 20(12), pp.3361-3376.

[11] Hussain, A., Ahmad, M., Hussain, T. and Ullah, I., 2022. Efficient Content-Based Video Retrieval System by Applying AlexNet on Key Frames. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 11(2), pp.207-235.

[12] Shen, L., Hong, R., Zhang, H., Tian, X. and Wang, M., 2019. Video retrieval with similarity-preserving deep temporal hashing. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 15(4), pp.1-16.

[13] Bekhet, S. and Ahmed, A., 2020. Evaluation of similarity measures for video retrieval. Multimedia Tools and Applications, 79(9), pp.6265-6278.

[14] Saoudi, E.M. and Jai-Andaloussi, S., 2021. A distributed content-based video retrieval system for large datasets. Journal of Big Data, 8(1), pp.1-26.

[15] Tarigan, J.T., Sihombing, P. and Marpaung, E.P., 2018. Implementing content-based video retrieval using speeded-up robust features.