SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/https://doi.org/10.14445/23488379/IJEEE-V12I12P105

Volume 12 Issue 12, 51-70, December 2025
© 2025Seventh Sense Research Group®

Original Article

An Enhanced Content-Based Image Retrieval with Similarity and ROI Analysis using AD-ES-CNN and EPL-Fuzzy

S. Ravi[1], Kamal Sutaria[2]

[1]*Faculty of Engineering and Technology, Parul University, Waghodia, Vadodara, Gujarat, India.*
[2]*Department of Computer Science and Engineering, Parul Institute of Engineering and Technology, Faculty of Engineering and Technology, Parul University, Waghodia, Vadodara, Gujarat, India.*

[1]*Corresponding Author : 2123004136006@paruluniversity.ac.in.*

Received: 04 October 2025 Revised: 06 November 2025 sAccepted: 05 December 2025 Published: 27 December 2025

Abstract - *To retrieve relevant images from a large database, Content-Based Image Retrieval systems (CBIR) are designed centered on the query image's visual content. Yet, the existing approaches often struggled with cluttered scenes, complex backgrounds, and the Region of Interest (ROI) in the image. Therefore, to address these challenges, this paper introduces You Only Live Once Version 3 (YOLO V3) and Renormalized Entropy - Gaussian Mixture Model (RE-GMM) approaches. Primarily, the image datasets are collected and pre-processed. The foreground and background objects are separated. Next, by using YOLOV3, the separated objects are detected. Further, by using Gram-Graph Cut (G^2C), detected objects are segmented, and saliency mapping is carried out. Afterward, the features are extracted. Then, by using the Gaussian Mutation Tuna Swarm Optimization Algorithm (GM-TSOA), the features are selected from the extracted features. Afterward, by using the AD-ES-CNN algorithm, the object classification is performed. Also, the structural similarity index and semantic feature similarity are carried out by using the SIFT and Jaccard Index, respectively. Lastly, by using the EPL-Fuzzy approach, the object is retrieved and indexed. As per the experimental analysis, the proposed model attained 99.29% accuracy for the Caltech 256 image dataset and 98.5% accuracy for the Corel image dataset.*

Keywords - *Region of Interest (ROI), T-splines Contrast Limited Adaptive Histogram Equalization (T-CLAHE), Renormalized Entropy - Gaussian Mixture Model (RE-GMM), Gram-Graph Cut (G^2C), Gaussian Mutation Tuna Swarm Optimization Algorithm (GM-TSOA), Attention Drop E-Swish Convolutional Neural Networks (AD-ES-CNN), Exponential Piecewise Linear (EPL-Fuzzy).*

1. Introduction

Widespread storage of images across numerous domains like healthcare, education, e-commerce, architecture, and social media is caused by digital imaging technologies' rapid growth [1]. CBIR systems have emerged as a powerful solution to effectively manage and retrieve this huge amount of visual data [2]. CBIR focuses on extracting features directly from the images, like color, texture, and shape, to facilitate the search for a specific image within a large database [3, 4]. Yet, these basic features can sometimes miss important details, making it difficult to achieve the most accurate results [5]. Thus, to enhance retrieval performance, advanced techniques are needed to consider the necessary factors like object recognition, spatial relationships, and regional content [6]. The Conventional image retrieval methods typically focus on feature extraction techniques, such as edge detection, pattern recognition, and statistical analysis [7, 8]. Yet, these methods often struggle to handle complex or cluttered scenes effectively, where multiple objects with varying backgrounds can obscure the key elements [9]. To address these challenges, recent advancements in Deep Learning (DL) and Machine Learning (ML) have been integrated into CBIR systems and offer significant improvements in feature extraction and similarity analysis [10, 11].

Most of the existing research methodologies used DL methods like Convolutional Neural Networks (CNN) since they are known for their ability to automatically learn hierarchical image representations and become integral to modern CBIR frameworks [12, 13]. Similarly, to learn the inherent patterns within the training data, conventional models of ML methodologies like K-Nearest Neighbor (KNN), along with Support Vector Machine (SVM), were used [14, 15]. Nevertheless, none of the prevailing works concentrated on cluttered scenes, occlusions, complex backgrounds, and the presence of multiple objects in the image for image retrieval.

Thus, this research methodology proposes the similarity and ROI analysis of CBIR employing AD-ES-CNN and EPL-Fuzzy approach.

1.1. Problem Statement

Some drawbacks in the existing research methodologies are,

- None of the existing works aimed at the cluttered scenes, occlusions, complex backgrounds, and the ROI in the image. If the system failed to precisely identify the ROI in an image, then it resulted in inaccurate retrieval results.
- The existing work [16] did not focus on the entire region of an image in ROI-based CBIR systems.
- The existing work [17] did not focus on addressing the semantic gap between lower-level visual features and higher-level semantic concepts in ROI-centric CBIR systems.
- The existing work [18] did not focus on efficient indexing and retrieval techniques for handling large-scale image collections, which led to issues with the scalability of CBIR systems.
- Most of the existing works required users to manually annotate the ROI, which resulted in time consumption and inaccuracies.

1.2. Objectives

To overcome the above problems, the proposed work's objectives are given below:

- To concentrate on cluttered scenes, occlusions, complex backgrounds, and the ROI, the RE-GMM and YOLO V3 approaches are used.
- To consider the entire region of an image in ROI-based CBIR systems, the (G^2C) algorithm is used.
- The semantic gap between lower-level visual features and higher-level semantic concepts, in ROI-centric CBIR systems, is addressed by using saliency mapping.
- The efficient indexing and retrieval techniques for handling large-scale image collections are done by using the EPL-Fuzzy approach.
- The AD-ES-CNN classifier is used as a DL classification model to annotate and categorize the objects.

The remaining part is organized as follows: the existing research works are elucidated in Section 2, the proposed approach is elucidated in Section 3, the proposed approach's performance is assessed with the prevailing research works in Section 4, and in Section 5, the paper is concluded with future enhancement.

2. Literature Survey

[16] established a DL model for image retrieval based on color objects through ROI segmentation. To classify the data with class labels and effectively retrieve images, pictures, and color regions, the Feed Forward Neural Network was used. As per the experimental analysis, the presented approach attained a higher accuracy of 87.33% when weighed against the existing methods. Nevertheless, the network used in this method required a larger number of parameters to achieve accurate results.

[17] advanced a framework for a CBIR system for disease diagnosis. To assess contextual similarity between images, the Local Ternary Pattern, Local Phase Quantization, and Discrete Wavelet Transform were wielded by constructing a balanced graph. The experimental results revealed that it achieved a mean average precision of 70%, outperforming existing methods. Yet, the background region of the image was not concentrated, causing a loss of image information.

[18] An advanced, effective framework for a CBIR system with high retrieval performance. For classifying semantically similar images and retrieving images from a broader range of image classes, the Support Vector Machine (SVM) was employed. As per the experimental results, the presented approach provided superior retrieval performance when compared to the existing approaches. Yet, SVM had a long training time, particularly when dealing with large datasets.

[19] developed a DL method for CBIR, specifically for traditional woven fabric images. This method employed a CNN to extract features. As per the experimental analysis, the presented approach achieved a higher accuracy of 99.96%. Nevertheless, it was slower because of the presence of max pool operations in the architecture.

[20] An advanced, effective framework for CBIR employing the DL method. To improve image retrieval accuracy, the framework utilized DL Enhanced CNN (DLECNN) by focusing on relevant and similar features. Thus, the presented approach achieved high accuracy, precision, recall, and f-measure, along with reduced complexity. Nevertheless, the system did not incorporate an optimization-based feature selection algorithm.

[21] developed an approach for implementing a CBIR system employing artificial neural networks and DL methods. To predict class labels for each feature vector, the SVM, K-Nearest Neighbours (KNN), and CNN approaches were utilized. As per the experimental results, the presented approach achieved high accuracy and efficiency when compared to the other methods. Yet, the system was limited by a lack of lower-level features, which affected its performance.

[22] established a DL approach for image classification and retrieval utilizing an attention mechanism. To effectively identify and classify, along with retrieving several architectural images, the CNN method was used. As per the

experimental analysis, the presented approach outperformed existing methods regarding retrieval accuracy. Yet, the image quality in the datasets needed improvement as the model's generalization ability was currently limited.

[23] advanced a framework for the CBIR system based on transfer learning to retrieve images using ML and DL methods. To compute image similarity, the KNN algorithm was used. As per the experimental analysis, the presented approach achieved the highest precision in identifying the similarities when compared to the conventional methods. Still, when querying similar images in the database at different orientation angles, the system's retrieval performance might degrade.

[24] advanced an ML model for effective CBIR employing features like color, grayscale, advanced textures, and shape. To retrieve the input query image from the image database, the Random Forest classifier was used by classifying the training images. As per the results, the presented approach provided superior image retrieval performance when compared to the existing works. Yet, the system was not able to handle remote sensing images or multispectral images.

[25] offered a framework for a CBIR system for disease diagnosis. To extract deep features from medical images, the Visual Geometry Group (VGG) 16 architecture was used, thereby providing a strong representation. As per the experimental analysis, the presented approach achieved higher precision and mean average precision when compared to the other existing models. Yet, the system lacked real-time feature extraction along with unique neural designs.

3. Proposed Roi and Similarity Analysis for Content-Based Image Retrieval using AD-ES-CNN and EPL-Fuzzy Approach

Here, ROI and similarity analysis for CBIR are presented. Here, the important phases of the research are foreground and background separation, image segmentation, Object classification, structural similarity index, and semantic feature similarity. In Figure 1, the proposed framework's structural diagram is presented.

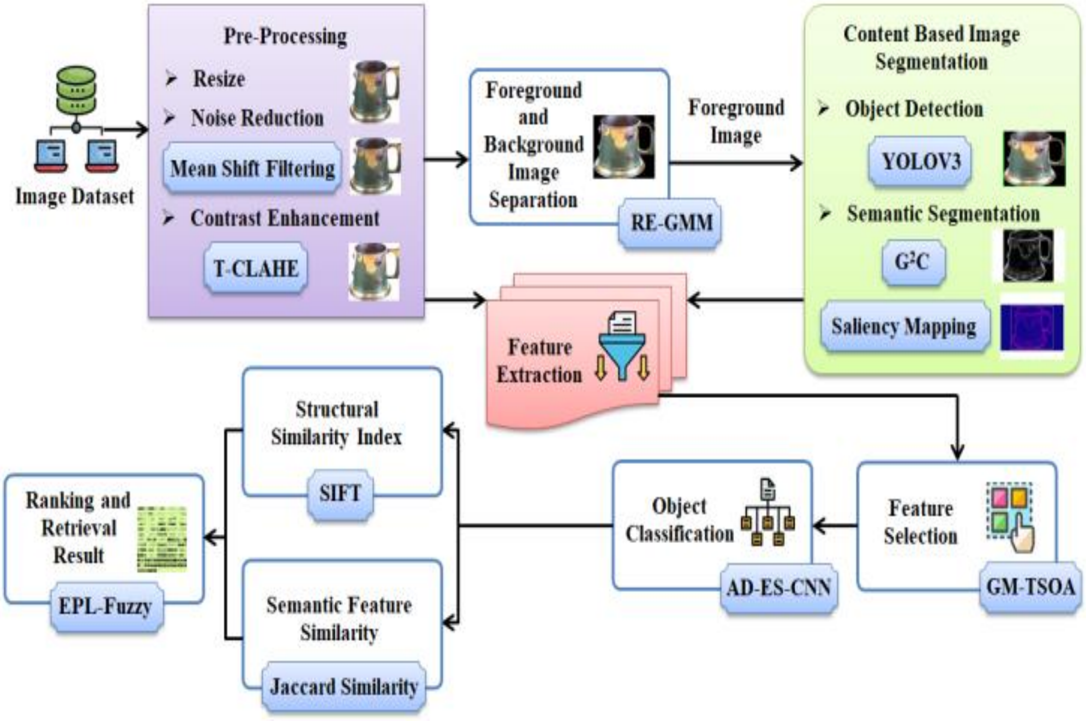


Fig. 1 Structure diagram for the proposed methodology

3.1. Image Dataset

Primarily, from the dataset, the input images of various categories, such as airplanes, grasshoppers, golf balls, etc., are collected. These images are represented as follows,

$$\exists = \{\exists_1, \exists_2, \dots \dots \exists_G\} \tag{1}$$

Here, $(\exists)$ signifies the collected dataset, and G -specifies the number of data points present in$(\exists)$.

3.2. Pre-Processing

Here, to improve the image quality, $(\exists)$ undergoes pre-processing. Here, the pre-processing stage involves three processes, which are described below:

3.2.1. Image Resizing

Initially, $(\exists)$ is resized. This process is essential for a consistent process and analysis. The resized images $(R)$are given as,

$$R = r_e\big(W(\exists), H(\exists)\big) \tag{2}$$

Here, (r_e) signifies the resize operation and (W)along with (H) signifies the new width and height of the image, respectively.

3.2.2. Noise Reduction

After the image resizing, by using the Mean Shift Filtering, the noise reduction of (R) is done. Mean Shift Filtering is a non-parametric technique used as a smoothing method in image processing, particularly for noise removal and edge detection. The noise from $(R)$is reduced by the Mean Shift Filtering and is given in equation (3),

$$n = \frac{\sum_l \forall(R-R_l)\cdot R_l}{\sum_l \forall(R-R_l)} - R \tag{3}$$

Here, (n) signifies the reduced noise of the image(R) and (R_l) embodies the neighboring data points.

3.2.3. Contrast Enhancement

After removing the noise from the image by using the T-CLAHE approach, the contrast level of the image is enhanced for similarity analysis of CBIR. The CLAHE approach enhances the contrast of the image more effectively by preserving the local details. Yet, the use of the bilinear interpolation method results in a loss of detail, especially in areas with high-frequency textures or sharp edges. Likewise, the blurring effect is due to the smoothing nature of the interpolation method, which averages neighboring pixel values. To solve this issue, the T-splines interpolation method is used to accurately capture both smooth and sharp features in the image. Initially, (n) is divided into a number of regions.

$$B = \{B_1, B_2, B_3, \dots, B_s\} \tag{4}$$

Here, (B) implies the separated region from the image and (B_s) specifies (s)-number of image regions. Then, the histogram $(\hbar)$of each region of the image is done. Then, the clip limit (c_l) is computed for the image as follows,

$$c_l = \frac{d}{e}\big(1 + \varepsilon(\hbar - 1)\big) \tag{5}$$

Here, (ε) implies a clip factor, and$(d)$and (e) signify the size of the image region and grayscale values, respectively. Next, the cumulative histogram (T) of the contextual region can be expressed as follows,

$$T = \frac{\sum_{s=0}^{s} \hbar, c_l}{p_i} \tag{6}$$

Here, (p_i) signifies the pixel coordinates. Then, normalization of (T) is performed, which ensures that the pixel intensity values are scaled within the range of the output image. It is expressed as follows,

$$E = \frac{T - T_{\min}}{T_{\max} \;\; - T_{\min}} \tag{7}$$

Here, Esignifies the normalized value of (T), and $T_{\min}$ and $T_{\max}$ are the minimal and maximal values of(T), respectively. Next, to create an output image with improved contrast, the T-splines interpolation method is used and is given as follows,

$$T_s(p_i) = B(p_i) \cdot E(p_i) \tag{8}$$

Here, (T_s) signifies the output images' interpolated value. Therefore, the pre-processed image is denoted as(∂).

3.3. Foreground and Background Separation

Here, the foreground and background objects are separated from the pre-processed image (∂) to detect the objects efficiently by using the RE-GMM algorithm. In the conventional Gaussian mixture model, the background is removed directly by the probability distribution function. Nevertheless, it does not consider the intensities as the missing of a small value might cause the loss of certain noteworthy details. Thus, the probability distribution function's Renormalized Entropy is used regarding the intensity level. Initially, the mean(m_e), covariance(c_v) , and weight(ω) are initialized to separate the foreground and background objects. The probability distribution function of the Renormalized Entropy $(\hat{E})$ is calculated for each point(∂) the following equation,

$$\hat{E} = -\frac{\sum_{i=1}^{p_k} \partial \log(\partial)}{\log(p_k)} \tag{9}$$

Here, (p_k) signifies the number of components in the GMM model. Then, the Maximization step takes place. In the M-step, the parameter (ω) is updated based on the responsibilities and is given as follows,

$$\omega'' = \frac{\sum_{i=1}^{u} \hat{E}}{u} \tag{10}$$

Here, (u) characterizes the number of data points and (ω'') signifies the updated parameter of weight (ω). Next, based on the responsibilities, the parameters $(c_v)$and $(m_e)$are updated and are given as follows,

$$m_e^{update} = \frac{\sum_{i=1}^{u} \hat{E}\cdot\partial}{\sum_{i=1}^{u} \hat{E}} \tag{11}$$

$$c_v^{update} = \frac{\sum_{i=1}^{u} \left(\partial - m_e{}'\right)^2}{\sum_{i=1}^{u} \hat{E}} \tag{12}$$

Here, $\left(c_v^{update}\right)$ and $\left(m_e^{update}\right)$ signifies the updated parameters of covariance (c_v) and mean (m_e), respectively. Until the log-likelihood converges to a maximum, these steps are repeated iteratively. The log-likelihood is calculated by using the following equation,

$$\Upsilon = \sum \log\left(\sum \omega''\left(\partial \middle| m_e^{update}, c_v^{update}\right)\right) \tag{13}$$

Here, (Υ) signifies the maximized log-likelihood function. Therefore, the foreground $\left(f_g\right)$ and background $\left(b_g\right)$ of the images are separated.

3.4. Content-Based Image Segmentation

Here, to analyze the CBIR system's ROI, the foreground image $\left(f_g\right)$ undergoes segmentation. Here, for the ROI analysis, the segmentation stage involves two processes, which are described in a further section.

3.4.1. Object Detection

Here, the objects are detected from$\left(f_g\right)$ within images and localized with Bounding Boxes (BB) (i.e., ROI) using the YOLO V3 algorithm. An object detection algorithm, which divides the image into a grid and predicts BB (i.e., ROI) and class probabilities for each grid cell in a single pass through the neural network, is termed YOLO. The algorithm of YOLO V3 is described below:

Primarily, the input $\left(f_g\right)$ is divided into grid cells. The image coordinates $(y, z)$of the center of the grid cell $\left(g_{grid}, h_{grid}\right)$can be expressed for each grid cell as,

$$g_{grid} = \left(\frac{y}{f_g}\right) \cdot \nu \quad , \quad h_{grid} = \left(\frac{z}{f_g}\right) \cdot \tau \tag{14}$$

Here, (ν) depicts the width of the image$\left(f_g\right)$, and (τ) depicts the height of the image$\left(f_g\right)$. Next, the BB (i.e., ROI) is predicted based on certain parameters for each grid cell. The parameters for each BB (i.e., ROI) (b)in the cell are given as,

$$\ddot{x} = \sigma(t_x) + g_{grid} \tag{15}$$

$$\ddot{y} = \sigma\left(t_y\right) + h_{grid} \tag{16}$$

$$\widetilde{w} = p^w \exp(t_w) \tag{17}$$

$$\tilde{h} = p^h \exp(t_h) \tag{18}$$

Here, $(\ddot{x}, \ddot{y})$ signifies the BB (i.e., ROI) center relative to the $\left(g_{grid}, h_{grid}\right)$, respectively, $\left(t_x, t_y\right)$ signifies the predicted offsets of $(\ddot{x}, \ddot{y})$, $\left(\widetilde{w}, \tilde{h}\right)$symbolizes the width and height of BB (i.e., ROI) relative to the entire image, (t_w, t_h)represents the predicted dimensions for BB (i.e., ROI), $(p_w, p_h)$are the prior values of width along with height of the BB (i.e., ROI),(exp) represents the exponential terms to ensure that the predicted width and height are positive, and (σ) represents the sigmoidal function, and it is given as,

$$\sigma = \frac{1}{1+e^{\left(g_{grid}, h_{grid}\right)}} \tag{19}$$

Here, (e) signifies the exponential value in(σ). Next, the confidence score (c_s) is predicted and is given as,

$$c_s = \sigma\left(d_{c_s}\right) \tag{20}$$

Here,$\left(d_{c_s}\right)$ signifies the confidence score prediction. Therefore, the overall BB (i.e., ROI) prediction for each grid cell is represented as,

$$B_x = \left(\ddot{x}, \ddot{y}, \widetilde{w}, \tilde{h}, c_s\right) \tag{21}$$

Next, class probabilities (C_P) are predicted for each BB (i.e., ROI), and are given as,

$$C_P = \widetilde{\varpi}\left(q_1, q_2, \dots, q_p\right) \tag{22}$$

Here, $\left(q_p\right)$ signifies the total number of classes as well as$(\widetilde{\varpi})$ represents the softmax activation function. Finally, objects are detected within images along with being localized with BB es (i.e., ROI), and it is expressed as,

$$\lambda = \{\lambda_1, \lambda_2, \lambda_3 \dots \dots \lambda_\nabla\} \tag{23}$$

Where, (λ) signifies the detected objects in the BB (i.e., ROI) and (λ_∇) signifies the detected objects in the BB (i.e., ROI).

3.4.2. Semantic Segmentation

Here**,** the objects are segmented from(λ) to extract the relevant features by using the G^{2}C algorithm. Graph cut segmentation maintains edges and structures, producing high-quality segmentation outcomes with well-defined boundaries.

Yet, covariance matrix regularization imposes constraints on the covariance matrices, which can sometimes lead to a reduction in the model's ability to capture significant information. Strong regularization may overly smooth the estimated covariance matrices, causing the model to underfit the data and miss important variations in the color distributions. Thus, the Gram matrix is used in the Graph Cut segmentation. Let $\left(\mu_{b_p}\right)$ be the label vector at the pixel $\left(b_p\right)$ of(λ). The data term (ζ_{data}) with respect to $\left(\mu_{b_p}\right)$ is given as follows,

$$\zeta_{data}(\mu_{b_p}) = -\ln P_e(In_{b_p} \mid \mu_{b_p}) \qquad (24)$$

Here, (In_{b_p})signifies the intensity value at the pixel (μ_{b_p})as well as(P_e) signifies the likelihood of observing the (In_{b_p}) at the pixel (μ_{b_p}). The smoothness term(ζ_{smo}) is employed to encourage neighboring pixels with similar colors to have the same label and is given as follows,

$$\zeta_{smo}(b_p, c) = \exp\left(-\frac{\left\|In_{b_p} - In_c\right\|^2}{2\eta^2}\right) \qquad (25)$$

Here, (c) depicts the neighboring pixel, (η) signifies the parameter controlling the sensitivity of the smoothness term, and (In_c) represents the intensity value at the pixel (c). The smoothness term is then implemented with a Gram matrix that encodes the similarity between neighboring pixels. The Gram matrix is used based on the pixel value, which encodes the color or intensity of the pixels. Thus, the Gram matrix (G_M) is constructed and is given as follows,

$$G_M = \ell \cdot \ell^{trans} \qquad (26)$$

Here, (ℓ) signifies the matrix of the vector in $(\lambda)$and (ℓ^{trans}) represents the transpose of the matrix (ℓ). The energy function $(\xi)$to be minimized is given as follows,

$$\xi = \aleph \sum_{b_p \in \lambda} \zeta_{data}(\mu_{b_p}) + \sum_{(b_p, c) \in \lambda} \zeta_{smo}(b_p, c) \bullet G_M \qquad (27)$$

Here, $(\aleph)$ depicts the weight scalar value controlling the relative importance of the data term and smoothness term. Once the energy function is minimized, the semantically segmented objects are obtained and are denoted as (S_{seg}).

3.4.3. Saliency Mapping

Here, for further processing in saliency mapping, (S_{seg}) are provided. Saliency mapping helps to identify regions of interest within an image that are *visually* significant or attention-grabbing.

These areas usually include objects, textures, or patterns that are unique or meaningful, making them useful for finding and retrieving information. It is mathematically derived as follows,

$$ƛ = \sum \Psi \times |\xi - In| \qquad (28)$$

Here, $(ƛ)$ depicts the output of the saliency mapping process, (In)signifies the image's intensity value, and (Ψ) signifies the frequency of (In).

3.5. Feature Extraction

Then, the features are extracted for the effective classification of images from (∂) and $(ƛ)$. The features of (∂), namely Gabour, Corner, Interest point, Entropy, Dissimilarity, Histogram of Local Ternary Patterns, Histogram of Oriented Gradients (HOG), Brightness, Speeded-Up Robust Features (SURF), Energy, Correlation, Shape, Rotation, and Local Phase Quantization (LPQ), and the features of $(ƛ)$ like SIFT, Contrast, Homogeneity, skewness, kurtosis, mean, area value, size, and Standard Deviation are extracted and represented as,

$$V = \{V_1, V_2, \ldots\ldots, V_{tp}\} \qquad (29)$$

Here, (V) specifies the extracted feature set and (V_{tp}) denotes (tp)-number of extracted features.

3.6. Feature Selection

Here, from the extracted features (V), the optimal features are chosen to reduce the classifier's complexity and increase its interpretability in high-dimensional attribute spaces by using GM-TSOA. The existing TSOA is used to explore the entire search space efficiently, making it suitable for global optimization problems. However, this approach suffers from limited exploration, especially in high-dimensional or complex search spaces beyond local optima. In order to solve this issue, Gaussian Mutation is used in the exploration phase. Initially, the populations are initialized. Here, the extracted features are considered as the population. Primarily, the position is randomly initialized with the number of tuna in the population (S_n) and the number of iterations(ϕ) using equation (30).

$$Z_r = \Theta \times (up_b - lo_b) + lo_b \qquad (30)$$

Here, (Z_r) represents the initial position of the (r^{th}) individual, (Θ) is expressed as a random number, and (lo_b) and (up_b) are the search spaces' lower as well as upper bounds, respectively. Then, the fitness value is estimated for the population. Here, the maximal accuracy is the fitness function, which is given in equation (31),

$$N_F = M_{ax}(\gamma_{acc}) \qquad (31)$$

Here, $M_{ax}(\gamma_{acc})$ signifies the maximum accuracy and (N_F) represents the fitness function of the derived population.

3.6.1. Phase 1: Spiral Foraging Strategy

The tuna group chases the prey by forming a tight spiral formation in this spiral foraging phase. Here, the Gaussian mutation process is carried out to enhance the searching behavior, particularly in high-dimensional and complex search spaces. The Gaussian mutation is expressed as follows,

$$\varpi = Z_r + \cup\,(0, \Lambda^2) \qquad (32)$$

Here, (ϖ) signifies the muted value of the solution, $\cup(0, \Lambda^2)$ signifies the normal distribution with mean 0, along with variance (Λ^2). The updated position of the new solution is given as follows,

$$Z_{1new} = \begin{cases} \begin{pmatrix} \delta_1 \cdot \left(Z_{ref} + \rho \cdot \left|Z_{ref} - \varpi\right| + \delta_2 \cdot \varpi\right) \\ \delta_1 \cdot \left(Z_{ref} + \rho \cdot \left|Z_{ref} - \varpi\right| + \delta_2 \cdot \varpi_{t-1}\right) \end{pmatrix}, if\Theta < \dfrac{\phi}{\phi_{\max}} \\ \begin{pmatrix} \delta_1 \cdot (Z_{best} + \rho \cdot |Z_{best} - \varpi| + \delta_2 \cdot \varpi) \\ \delta_1 \cdot (Z_{best} + \rho \cdot |Z_{best} - \varpi| + \delta_2 \cdot \varpi_{t-1}) \end{pmatrix}, if\Theta \geq \dfrac{\phi}{\phi_{\max}} \end{cases} \tag{33}$$

Here, (Z_{1new}) represents the updated position for the tuna swarm's spiral foraging strategy, (δ_1) and (δ_2) are the weight coefficients, which control individual tunas' tendency to move near the optimal individual as well as the preceding individual, (ρ) is the distance parameter that controls the distance between the tuna individual and the optimal individual, $(\phi_{\max})$ signifies the maximal iterations, (Z_{ref}) is the reference point in the random position, and (Z_{best}) represents the present best tuna individual.

3.6.2. Phase 2: Parabolic Foraging Strategy

The mathematical model of the tuna swarm's parabolic foraging is,

$$Z_{2new} = \begin{cases} Z_{best} + \Theta \cdot \left(Z_{best} - Z_r^{\phi}\right) + R_a \cdot \beta^2 \cdot \left(Z_{best} - Z_r^{\phi}\right), if\Theta < 0.5 \\ R_a \cdot \beta^2 \cdot Z_r^{\phi}, if\Theta \geq 0.5 \end{cases} \tag{34}$$

$$\beta = (1 - \phi/\phi\_\mathrm{max}\ )\text{^}(\phi/\phi\_\mathrm{max}\ )$$

Here, (Z_{2new}) depicts the new position updated for the parabolic foraging strategy of the tuna swarm, (Z_r^{ϕ}) depicts the (r^{th})individual tuna's position at the (ϕ^{th}) iteration, and(R_a) is the random value, which is either 1 or -1. Thus, the features are selected and are denoted as (F^{select}).

3.7. Object Classification

Here, to categorize the objects, (F^{select}) is given as the input to the AD-ES-CNN approach. CNNs are highly effective at processing images because they automatically extract the features. This enhances the efficiency of CNNs in performing tasks like image classification. However, CNN is susceptible to overfitting and Vanishing Gradient issues, particularly when trained on small datasets or when model capacity exceeds the complexity of the task. To solve overfitting and improve generalization performance, Attention Drop regularization techniques and E-Swish activation function are used in the CNN architecture. In Figure 2, the AD-ES-CNN classifier is given.

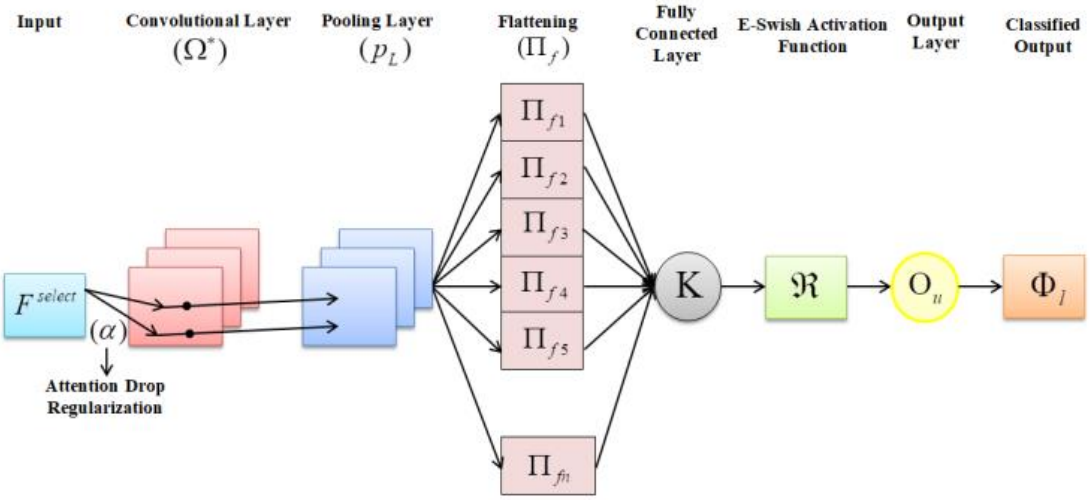


Fig. 2 Structure of the AD-ES-CNN approach

3.7.1. Weight Initialization

By using the Attention Drop regularization, the weight(α) of the classifier is initialized, which is given as,

$$\alpha = \frac{L \otimes m_a}{\rho_{rob}} \tag{35}$$

Here, (L) implies the attention matrix used for training, (m_a)signifies the drop mask applied on the attention matrix, and (ρ_{rob}) signifies the probability of keeping an attention weight in the attention matrix.

3.7.2. Convolution Layer

Here, $(F^{select})$is given as input to the Convolutional Layer, which convolves the input and gets activated using the E-Swish activation function $(\Re)$ as shown below,

$$\Omega^* = \{[(F^{select}) * (\alpha)] + b^s\} \times \Re \tag{36}$$

$$\Re = (F^{select}) \cdot \frac{1}{1+e^{-F^{select}}} \tag{37}$$

Here, (Ω^*)depicts the output of the convolutional layer, as well as (b^s) depicts the bias value.

3.7.3. Pooling Layer

The pooling equation is calculated as follows,

$$p_L = \max(\Omega^*) \tag{38}$$

Where, (p_L) represents the pooling layer's output.

3.7.4. Fully Connected Layer (FCL)

The flattened layer's equation is expressed as follows,

$$\Pi_f = flat(p_L) \tag{39}$$

Here, (Π_f) is the flattened layer, and $flat(\dashv)$is the flattening of the maximum pooling layer. Finally, the obtained output is given to the FCL based on the activation function $(\Re)$ and is derived as follows,

$$K = \left[\Re \sum \Pi_f \times \alpha\right] + b^s \tag{40}$$

Here,(K) signifies the output of an FCL. Later, the output is obtained and is given as,

$$O_u = \frac{e^K}{\sum_{d=1}^{P} e^{K_d}} \tag{41}$$

Here,(O_u) signifies the output layer's outcome, (P) characterizes the total number of classes, and$d$is the index value for overall classes. Therefore, the object is classified and is denoted as(Φ_l). The pseudo-code for AD-ES-CNN is given as follows,

Pseudo-code of the AD-ES-CNN approach

```
Input: Feature Selection, $(F^{select})$
Output: Classified objects, $(\Phi_l)$
Begin
        Initialize weights$(\Delta)$, bias $(b^s)$
        For each $(F^{select})$ do
                Perform activation function, $(\Re)$
                Convolute $\Omega^* = \{[(F^{select}) * (\Delta)] + b^s\} \times \Re$
                Evaluate,$p_L$
                Perform flattening, $\Pi_f$
                Compute fully-connected layer, $K = \left[\Re \sum \Pi_f \times \kappa\right] + b^s$
                Derive output layer, $O_u = (K \times \Re)$
                Obtain classified object, $(\Phi_l)$
                If $(classifiedoutcome == satisfied)${
                        Terminate
                } else {
                        Repeat
                } end if
        End For
        Return$(\Phi_l)$
End
```

Therefore, the classified object (Φ_l) is obtained and given for the similarity analysis. In both the training and testing phases, the above steps are performed. The similarity identification is carried out only during the testing phase as described in the following section.

3.8. Similarity Identification

Here, the similarity identification is carried out between the classified objects of the query images. $(\Phi_l)$and the input query image(q_{im}) to retrieve top images from a database that are visually similar to the query image. CBIR can provide users with relevant content based on their visual preferences and requirements by identifying similar images. The similarity identification is given as follows,

3.9. Structural Similarity Index

Here, by using the SIFT method, the identification of the structural similarity between (Φ_l) and (q_{im}) is done. The SIFT extracts local feature descriptors from images, which represent distinctive visual patterns, such as edges, corners, and textures. The Similarity Index value by query, along with the test image, is estimated based on the count of location-wise matches of points in the query along with the test image. The Similarity index value between (Φ_l) and $(q_{im})$is calculated using the following equation,

$$\tilde{\ell} = \frac{\left(2U_{\Phi_l}U_{q_{im}}+kc_1\right)\left(2g_{\Phi_l q_{im}}+kc_2\right)}{\left(U_{\Phi_l}^2+U_{q_{im}}^2+kc_1\right)\left(g_{\Phi_l}^2+g_{q_{im}}^2+kc_2\right)} \tag{42}$$

Here, U_{Φ_l} and $U_{q_{im}}$ depicts the mean values of images, $g_{\Phi_l q_{im}}$ depicts the covariance of the images,(kc_1) and (kc_2) signify the constant value, and $g_{\Phi_l}^2$and $g_{q_{im}}^2$ depicts the variances of the images. Next, by using SIFT, the feature points of the image $F_P(\Phi_l)$ and $F_P\left(\Phi_{q_{im}}\right)$ are obtained. Then, the feature points of both objects are matched and are denoted as$\left(M\left(\Phi_l, \Phi_{q_{im}}\right)\right)$. Next, the feature scores are calculated by using the following equation,

$$Q = \frac{\left(M\left(\Phi_l, \Phi_{q_{im}}\right)\right)}{F_P(\Phi_l)} \tag{43}$$

Here, (Q) signifies the SIFT score. Therefore, the final structural similarity index is calculated as follows,

$$SS_I = Q \cdot \tilde{\ell} \tag{44}$$

Here, (SS_I) signifies the structural similarity index.

3.10. Semantic Feature Similarity

Here, by using the Jaccard Index, the identification of the semantic feature similarity between (Φ_l) and (q_{im}) is done. The Jaccard Index (J_I) is given for semantic feature similarity by the following equation,

$$J_I = \frac{\left|\Phi_l \cap \Phi_{q_{im}}\right|}{\left|\Phi_l \cup \Phi_{q_{im}}\right|} \qquad (45)$$

Here**,** $(\cap)$ implies the set of common features between the two objects $\Phi_l$and$\Phi_{q_{im}}$, and $(\cup)$ is the set of all unique features from both objects Φ_l and $\Phi_{q_{im}}$.

3.11. Ranking and Retrieval Results

Here, by using the EPL-Fuzzy approach, the calculated similarity is provided as input for object retrieval to rank the objects based on the similarity score. The fuzzy algorithm is very easy as well as understandable; also, it could offer an effective solution to complex issues. Yet, the Fuzzy algorithm has difficulty in tuning the membership function. To solve this issue, the Exponential Piecewise Linear (EPL) Membership function is used in the Fuzzy algorithm. The EPL-Fuzzy method is described below,

3.11.1. Fuzzification

Primarily, the combination of $(SS_I)$and (J_I) is denoted as(Ξ). Here, the crisp inputs (Ξ) are converted into fuzzy data with respect to the membership function. It can be calculated as follows,

$$F_{uzzy} = \hat{\theta} \times \Xi \qquad (46)$$

Here, $\left(F_{uzzy}\right)$ implies the fuzzified data. The Exponential Piecewise Linear (EPL) Membership function of the fuzzy data is expressed as follows,

$$\hat{\theta} = \begin{cases} \ddot{e}^{(q)(\Xi)} - 1, if \Xi < 0 \\ \Xi, if \Xi \geq 0 \end{cases} \qquad (47)$$

Here, $\left(\hat{\theta}\right)$ signifies the EPL Membership function, $(\ddot{e})$ represents the exponential term of the fuzzy data, and (q) is the decay constant.

3.11.2. Rule Base

Here, based on the similarity of the objects, the if-then-based rules are set to rank them in order. The rules are set and are mentioned below,

$$\tilde{\gamma} = \begin{cases} R_h, if \Xi is > 0.5 \\ R_m, if \Xi is = 0.5 \\ R_{lo}, if \Xi is < 0.5 \end{cases} \qquad (48)$$

Here, $(\tilde{\gamma})$ signifies the set of rules for ranking the object based on the similarity level, (R_h)signifies the high-level ranking, (R_m) represents the medium-level ranking, and (R_{lo}) represents the low-level ranking.

3.11.3. Inference Engine

Regarding $\left(\hat{\theta}\right)$, the fuzzy data are inferred in order to generate the fuzzy output and map the fuzzy set, which leads to the defuzzification process. It is represented as follows,

$$I_f = \Xi \times \tilde{\gamma} \qquad (49)$$

Here, $\left(I_f\right)$ signifies the inferred fuzzy output of the given fuzzy set.

3.11.4. Defuzzification

The fuzzy output data $\left(I_f\right)$ obtained from inference is converted to quantifiable crisp output by the following equation,

$$de_{fuzzy} = \frac{\Sigma I_f \otimes \hat{\theta}}{\Sigma \hat{\theta}} \qquad (50)$$

Here, $\left(de_{fuzzy}\right)$ signifies the defuzzified output for the given fuzzy set. Therefore, the retrieved image is obtained and is denoted as (N_r). The pseudo-code for EPL-Fuzzy is given as follows,

Pseudo code for EPL-Fuzzy approach

```
Input: Crisp inputs, $\Xi$
Output: Retrieved images, $N_r$

Begin
    Initialize$v_{sim}$,$q$
    For each $\Xi$ do
        Evaluate fuzzy data, $F_{uzzy} = \hat{\theta} \times \Xi$
        Compute membership function, $\hat{\theta}$
        Set fuzzy rules
            If $(\Xi is > 0.5)${
                $\tilde{\gamma} \to R_h$
            } else if $(\Xi is = 0.5)${
                $\tilde{\gamma} \to R_m$
            } else $(\Xi is < 0.5)${
                $\tilde{\gamma} \to R_{lo}$
            } end if

        Infer the data, $I_f = \Xi \times \tilde{\gamma}$
        Defuzzify, $de_{fuzzy} = \frac{\Sigma I_f \otimes \hat{\theta}}{\Sigma \hat{\theta}}$
    End For
    Obtain, $N_r$
End
```

Lastly, the images similar to the input query image are ranked and retrieved based on the similarity score.

4. Results and Discussions

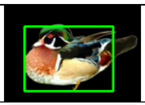
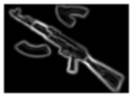
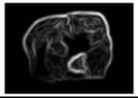
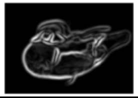
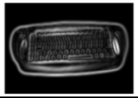
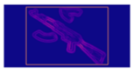
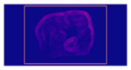
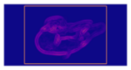
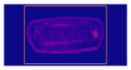
Here, the proposed AD-ES-CNN approach's performance for a CBIR system is assessed with the existing research methodologies. The proposed research work is implemented in the working platform of PYTHON.

4.1. Dataset Description

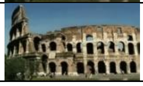
The proposed framework utilizes the Caltech 256 image and Corel image datasets for the performance analysis. The source link for the collected dataset is provided in the reference section. The Caltech 256 image dataset encompasses 30,607 images and 257 object categories. In this research, from the overall data (30,607), 80% data (24485) is wielded for training purposes, along with the remaining 20% data (6122) being deployed for testing purposes. The Corel image dataset encompasses 1000 images. Here, from the overall data (1000), 80% of the data (800) is wielded for training purposes; also, the remaining 20% of the data (200) is deployed for testing purposes.

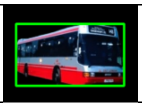
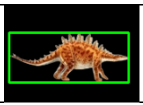
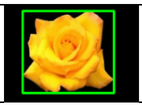
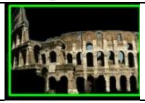
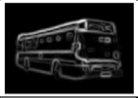
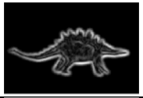
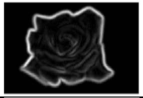
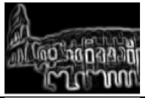
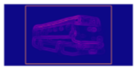
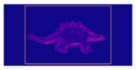
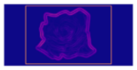
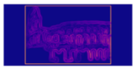
Table 1. Sample image outcomes of the proposed work

(a)

Images	**Image-1**	**Image-2**	**Image-3**	**Image-4**
Input				
Resize				
Noise Reduction				
Contrast Enhancement				
Foreground image				
YOLO detected image				
Semantic segmentation				
Saliency mapping				

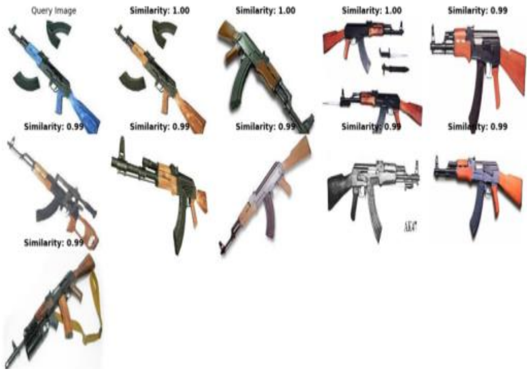
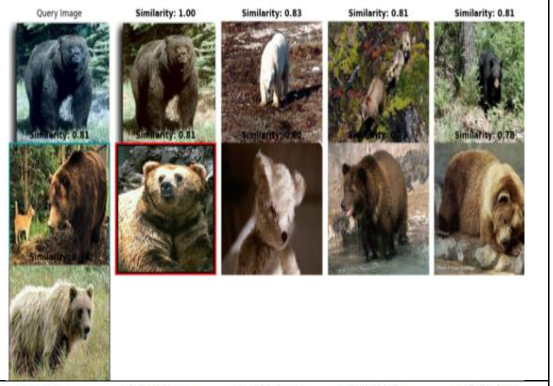
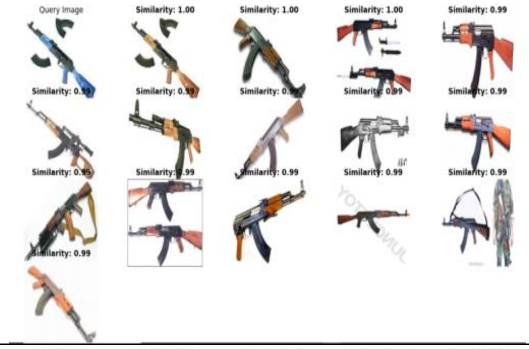
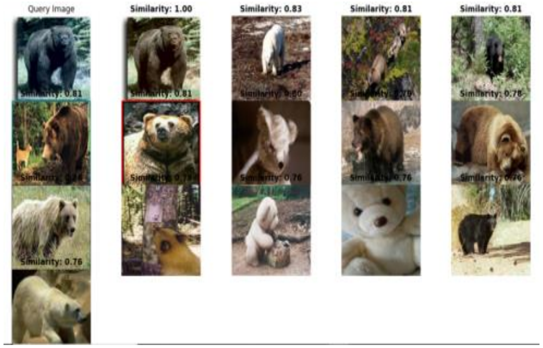
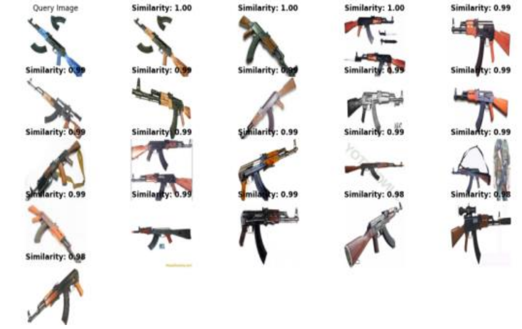
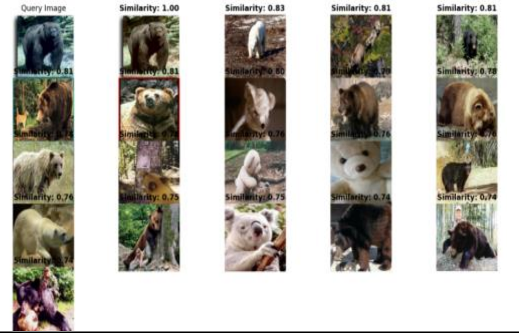
(b)

Images	**Image-4**	**Image-5**	**Image-6**	**Image-7**
Input				
Resize				
Noise Reduction				
Contrast Enhancement				

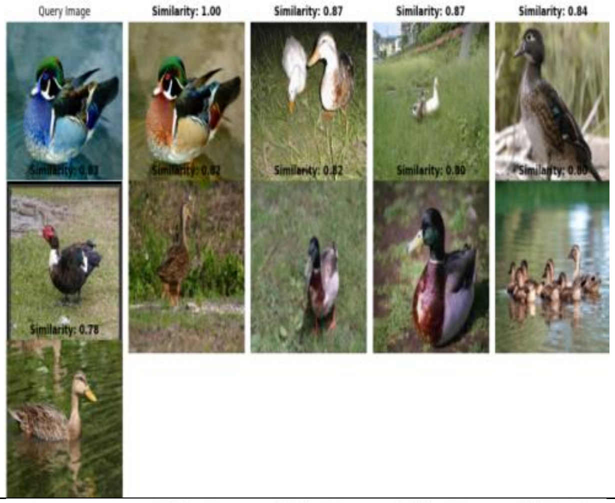
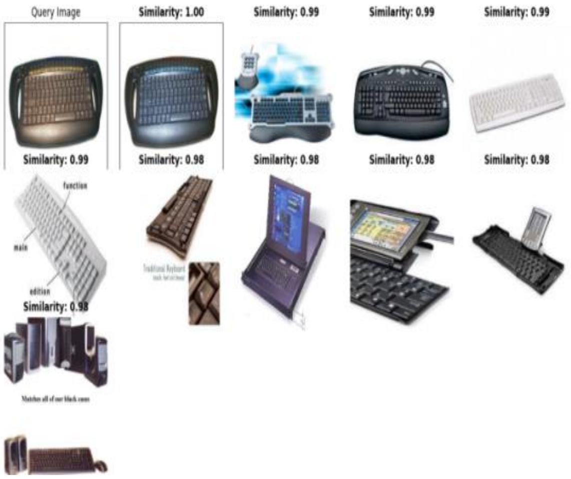
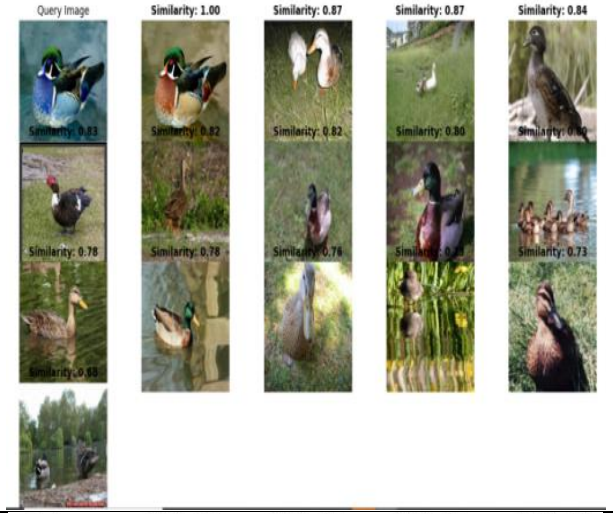
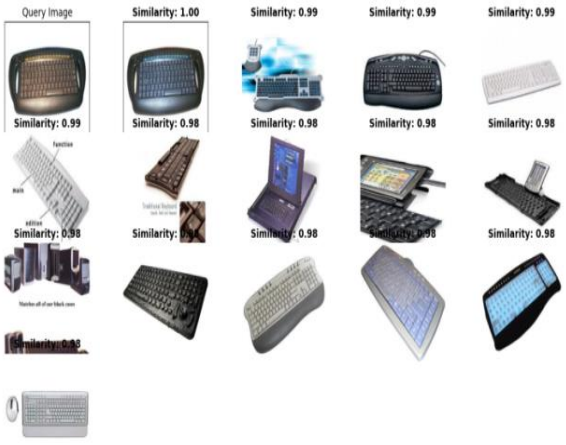
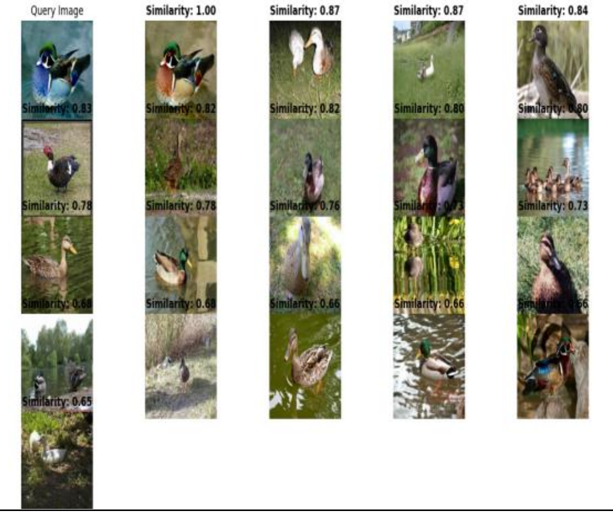
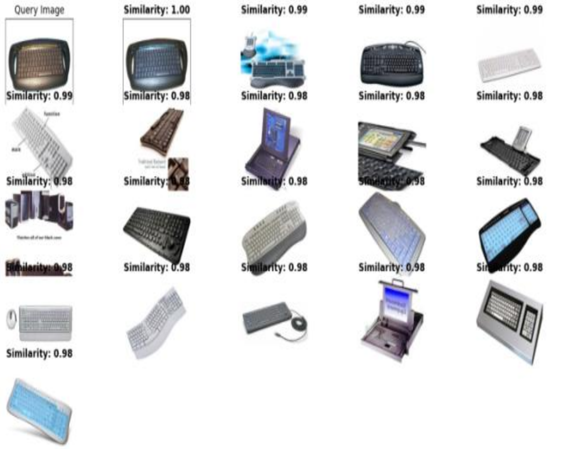
Foreground image				
YOLO detected image				
Semantic segmentation				
Saliency mapping				

Tables 1 (a) and (b) depict the sample image outcomes of the proposed work for the Caltech 256 image dataset and the Corel image dataset, according to the input image, resize, noise reduction, contrast enhancement, foreground image, YOLO detected image, semantic segmentation, and saliency mapping.

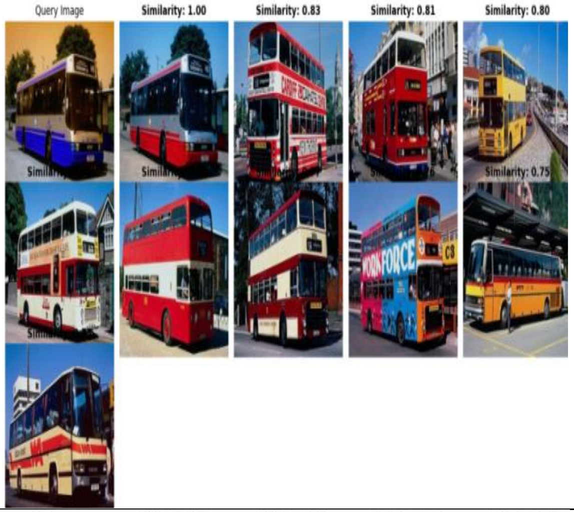
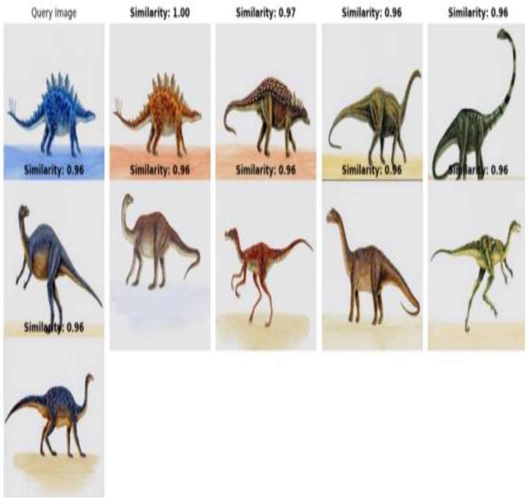
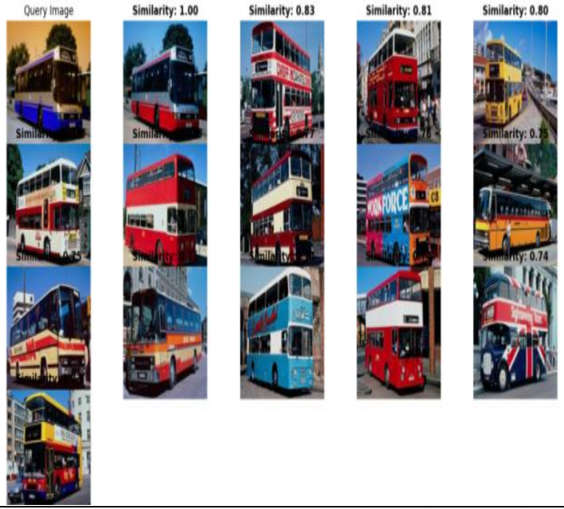
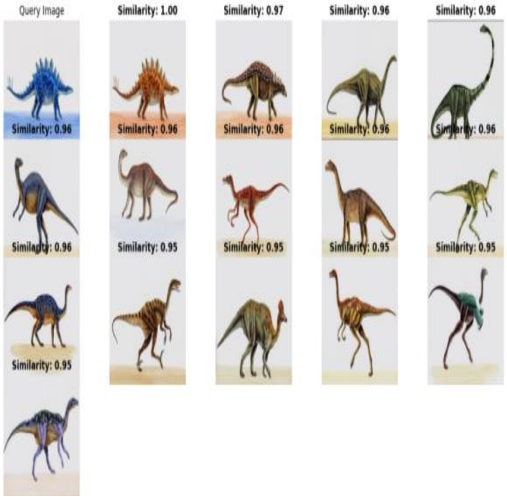
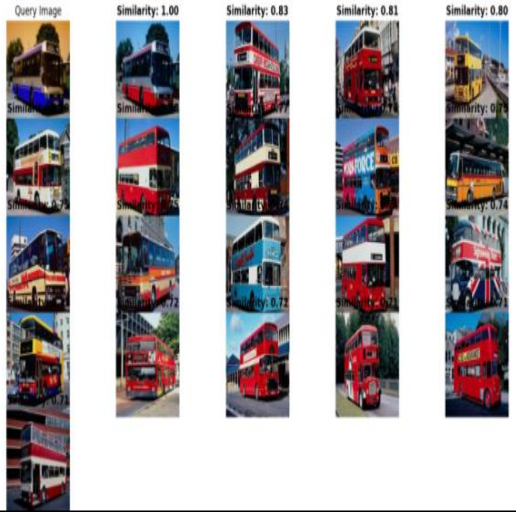
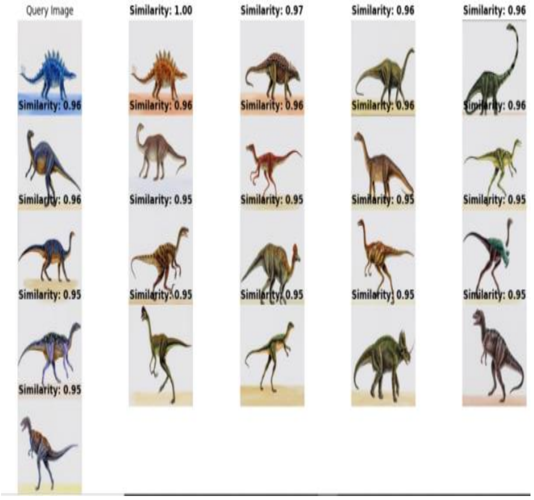
Table 2. Sample retrieval images of the proposed work
(a)

N value	Retrieval Image -1	Retrieval Image -2
N=10		
N=15		
N=20		

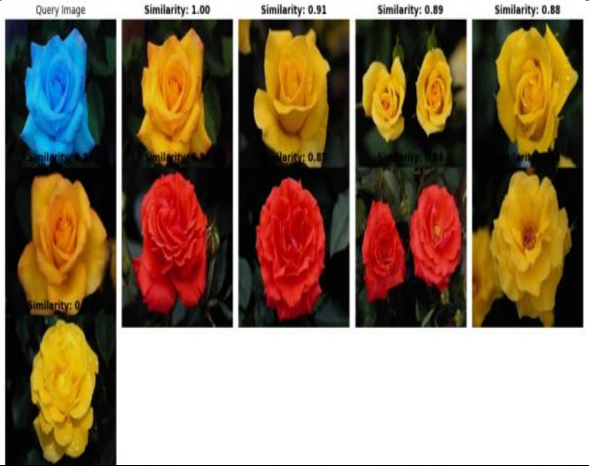
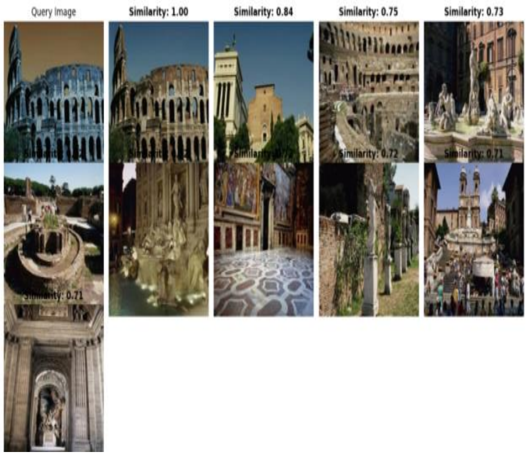
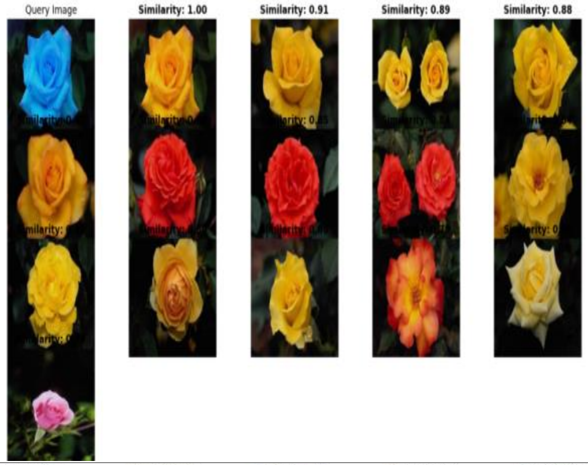
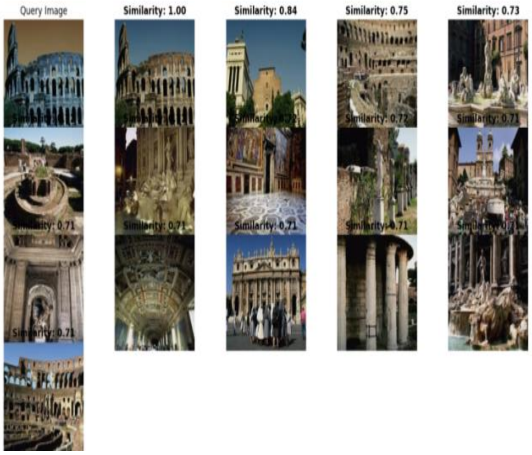
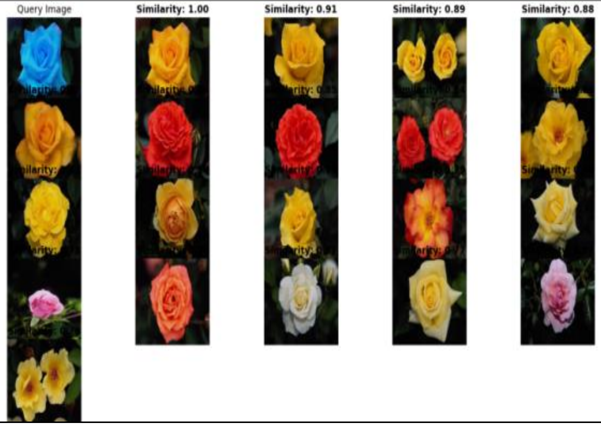
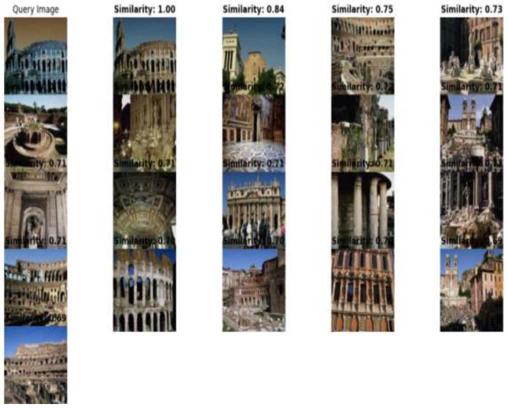
(b)

N value	Retrieval image-3	Retrieval image-4
N=10	 Query Image Similarity: 1.00 Similarity: 0.87 Similarity: 0.87 Similarity: 0.84 Similarity: 0.83 Similarity: 0.82 Similarity: 0.82 Similarity: 0.80 Similarity: 0.80 Similarity: 0.78	 Query Image Similarity: 1.00 Similarity: 0.99 Similarity: 0.99 Similarity: 0.99 Similarity: 0.99 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 function main edition Similarity: 0.98 Traditional Keyboard Matches all of our black cases
N=15	 Query Image Similarity: 1.00 Similarity: 0.87 Similarity: 0.87 Similarity: 0.84 Similarity: 0.83 Similarity: 0.82 Similarity: 0.82 Similarity: 0.80 Similarity: 0.80 Similarity: 0.78 Similarity: 0.78 Similarity: 0.76 Similarity: 0.73 Similarity: 0.68	 Query Image Similarity: 1.00 Similarity: 0.99 Similarity: 0.99 Similarity: 0.99 Similarity: 0.99 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 function main edition Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Matches all of our black cases Similarity: 0.98
N=20	 Query Image Similarity: 1.00 Similarity: 0.87 Similarity: 0.87 Similarity: 0.84 Similarity: 0.83 Similarity: 0.82 Similarity: 0.82 Similarity: 0.80 Similarity: 0.80 Similarity: 0.78 Similarity: 0.78 Similarity: 0.76 Similarity: 0.73 Similarity: 0.73 Similarity: 0.68 Similarity: 0.68 Similarity: 0.66 Similarity: 0.66 Similarity: 0.66 Similarity: 0.65	 Query Image Similarity: 1.00 Similarity: 0.99 Similarity: 0.99 Similarity: 0.99 Similarity: 0.99 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 function main Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98 Similarity: 0.98

(c)

N value	Retrieval image-5	Retrieval image-6
N=10	 Query Image Similarity: 1.00 Similarity: 0.83 Similarity: 0.81 Similarity: 0.80 Similarity: 0.75	 Query Image Similarity: 1.00 Similarity: 0.97 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96
N=15	 Query Image Similarity: 1.00 Similarity: 0.83 Similarity: 0.81 Similarity: 0.80 Similarity: 0.75 Similarity: 0.74	 Query Image Similarity: 1.00 Similarity: 0.97 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95
N=20	 Query Image Similarity: 1.00 Similarity: 0.83 Similarity: 0.81 Similarity: 0.80 Similarity: 0.74 Similarity: 0.72 Similarity: 0.72 Similarity: 0.71 Similarity: 0.71	 Query Image Similarity: 1.00 Similarity: 0.97 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.96 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95 Similarity: 0.95

(d)

N value	Retrieval image-7	Retrieval image-8
N=10	 Query Image Similarity: 1.00 Similarity: 0.91 Similarity: 0.89 Similarity: 0.88	 Query Image Similarity: 1.00 Similarity: 0.84 Similarity: 0.75 Similarity: 0.73 Similarity: 0.72 Similarity: 0.71 Similarity: 0.71
N=15	 Query Image Similarity: 1.00 Similarity: 0.91 Similarity: 0.89 Similarity: 0.88	 Query Image Similarity: 1.00 Similarity: 0.84 Similarity: 0.75 Similarity: 0.73
N=20	 Query Image Similarity: 1.00 Similarity: 0.91 Similarity: 0.89 Similarity: 0.88	 Query Image Similarity: 1.00 Similarity: 0.84 Similarity: 0.75 Similarity: 0.73

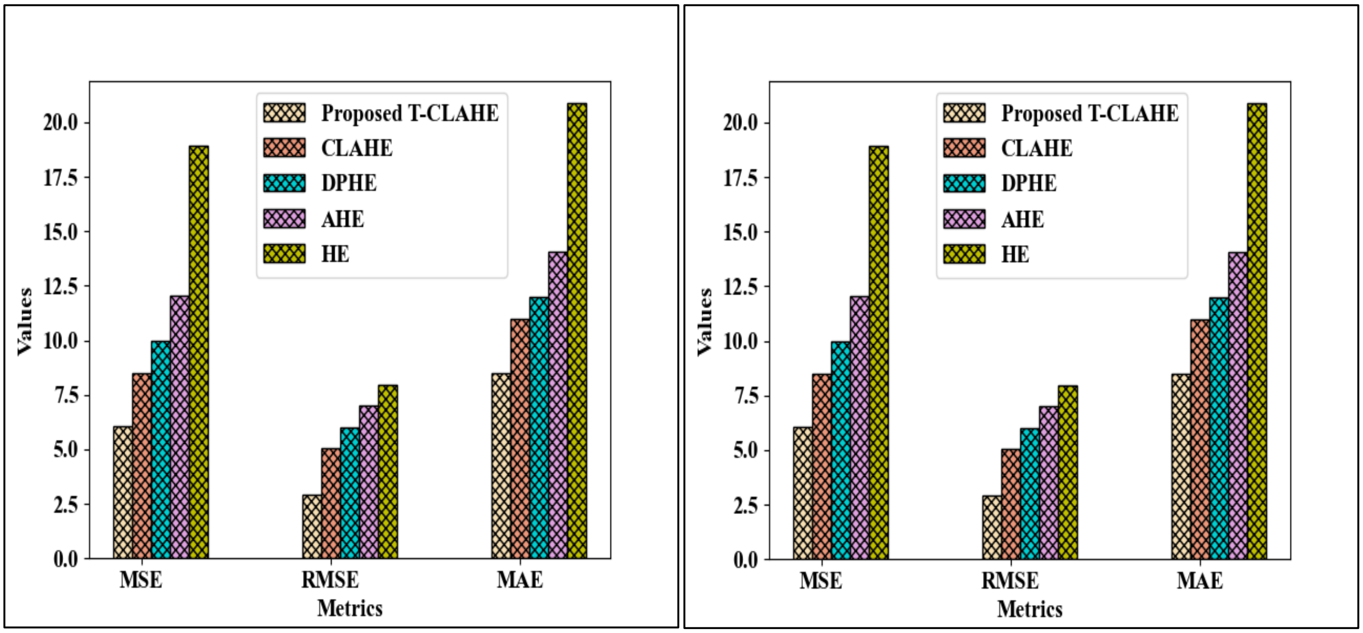
Tables 2 (a), (b), (c), and (d) depict the sample retrieval images of the proposed work according to the N value (N=10, 15, and 20).

4.2. Performance Analysis

Here, the proposed framework's performance is appraised with the existing works using the Caltech 256 image and Corel image datasets.

4.2.1. Performance Analysis for Contrast Enhancement

Here, the proposed T-CLAHE's performance is analyzed with the present Dynamic Probability Histogram Equalization (DPHE), Adaptive Histogram Equalization (AHE), and Histogram Equalization (HE). The performance of contrast enhancement regarding Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) for the Caltech 256 image and Corel image datasets is depicted in Figures 3 (a) and (b).



(a) **(b)**

Fig. 3 Performance analysis of contrast enhancement on (a) Caltech 256 image dataset, and (b) Corel image dataset.

Figure 3 (a) depicts that for the Caltech 256 image dataset, the proposed method achieved an MSE of 6.006229868, RMSE of 2.903755861, and MAE of 8.50164706, which showed better performance than the existing methodologies. In Figure 3 (b), for the Corel image dataset, the proposed method achieved a lower MSE of 6.01373105, RMSE of 2.9960093, and MAE of 8.47519657.

The existing approaches, such as CLAHE, DPHE, AHE, and HE in both datasets, achieved higher MSE, RMSE, and MAE values. The enhancement in the proposed approach was due to the usage of the T-splines interpolation method in the CLAHE approach.

4.2.2. Performance Analysis for Semantic Segmentation

Here, the proposed G^{2}C's performance is analyzed with the fundamental GC, Residual Network (ResNet), Segmented neural Network (SegNet), and Region Growing (RG). The performance of semantic segmentation based on the Jaccard Index (JI) for the Caltech 256 image and Corel image datasets is depicted in Table 3 and Figure 4.

As per Table 2, the proposed G^{2}C achieved a higher JI value of 0.04622 on the Caltech 256 image dataset. Figure 4 displays that for the Corel image dataset, the proposed G^{2}C achieved a higher JI value (0.054879), outperforming existing methodologies. In contrast, the existing approaches, such as GC, ResNet, SegNet, and RG, exhibited lower JI values on both datasets. The higher Jaccard Index of the proposed approach was due to the usage of the Gram matrix in the graph cut algorithm. Therefore, the performance of the proposed approach was superior in semantic segmentation.

Table 3. Ji Analysis on Corel image dataset

Methods	Jaccard Index
Proposed G^{2}C	0.054879
GC	-0.08745
ResNet	-0.06326
SegNet	-0.05456
Region Growing	-0.03475

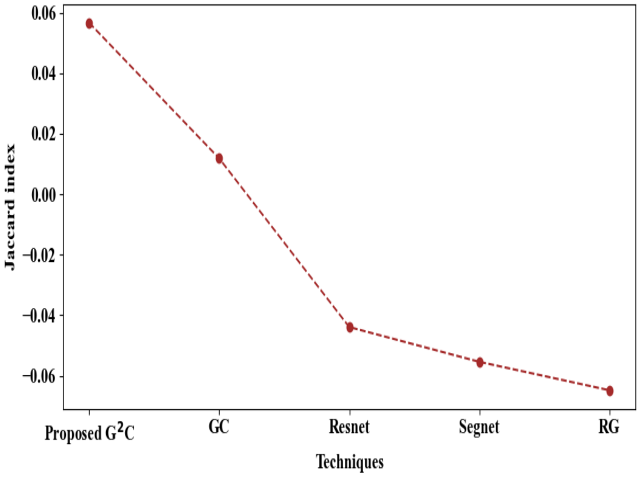


Fig. 4 JI analysis on Caltech 256 image dataset

4.2.3. Performance Analysis for the Ranking and Retrieval Process

The proposed EPL-Fuzzy's performance is weighed against the existing Fuzzy, Triangular Fuzzy Logic (TFL), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Rapid Back Propagation (RBP).

Table 4. Performance analysis for the ranking and retrieval process
(a) Caltech 256 image dataset

Methods	Fuzzification time (ms)	Defuzzification time (ms)	Rule Generation time (ms)
Proposed EPL-Fuzzy	671	647	479
Fuzzy	794	748	520
TFL	807	786	553
ANFIS	849	805	576
RBP	895	843	604

(b) Corel image dataset

Methods	Fuzzification time (ms)	Defuzzification time (ms)	Rule Generation time (ms)
Proposed EPL-Fuzzy	661	641	434
Fuzzy	783	737	503
TFL	801	778	523
ANFIS	841	801	556
RBP	873	833	601

The performance of the proposed and existing methodologies for the ranking and retrieval process centered on fuzzification, defuzzification, and with rule generation time on the Caltech 256 image dataset and the Corel image dataset is depicted in Tables 4 (a) and (b). According to Table 4 (a), for the Caltech 256 image dataset, the proposed method took a lesser fuzzification of 671 ms, a defuzzification of 647 ms, and a rule generation of 479 ms. Table 3 (b) depicts that the proposed EPL-Fuzzy took a lesser fuzzification of 661 ms, a defuzzification of 641 ms, and a rule generation of 434 ms for the Corel image dataset. This was due to the Exponential Piecewise Linear (EPL) Membership function used in the fuzzy algorithm. The existing Fuzzy, TFL, ANFIS, and RBP attained average fuzzification times of 836ms, 795ms, and 563ms in the Caltech 256 image dataset, respectively. They also took average fuzzification times of 824ms, 787ms, and 545ms in the Corel image dataset, respectively. Thus, in the ranking and retrieval process of CBIR, the proposed approach was highly helpful.

4.2.4. Performance Analysis for Feature Selection

The proposed GM-TSOA's performance is weighed against the prevailing TSOA, Butterfly Optimization Algorithm (BOA), Whale Optimization Algorithm (WOA), and Zebras Optimization Algorithm (ZOA). The fitness vs iteration analysis of the proposed and existing methodologies for the Caltech 256 image dataset and the Corel image dataset are depicted in Table 5 and Figure 5.

Table 5. Fitness analysis of the proposed and existing methodologies on the caltech 256 image dataset

Iterations	Proposed GM-TSOA	TSOA	BOA	WOA	ZOA
10	89.17	88.02	86.26	84.13	82.24
20	92.97	90.13	88.36	86.29	85.66
30	94.8	92.43	90.87	87.79	86.75
40	96.59	93.23	91.19	89.16	87.84
50	99.61	95.83	94.63	92.77	88.39

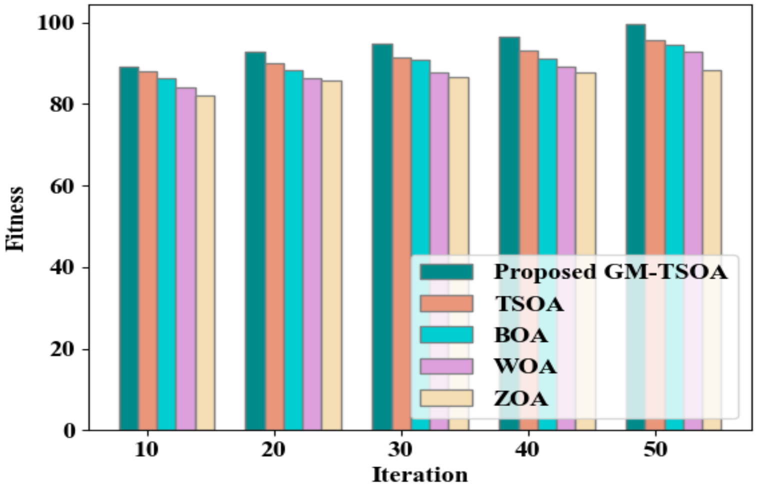


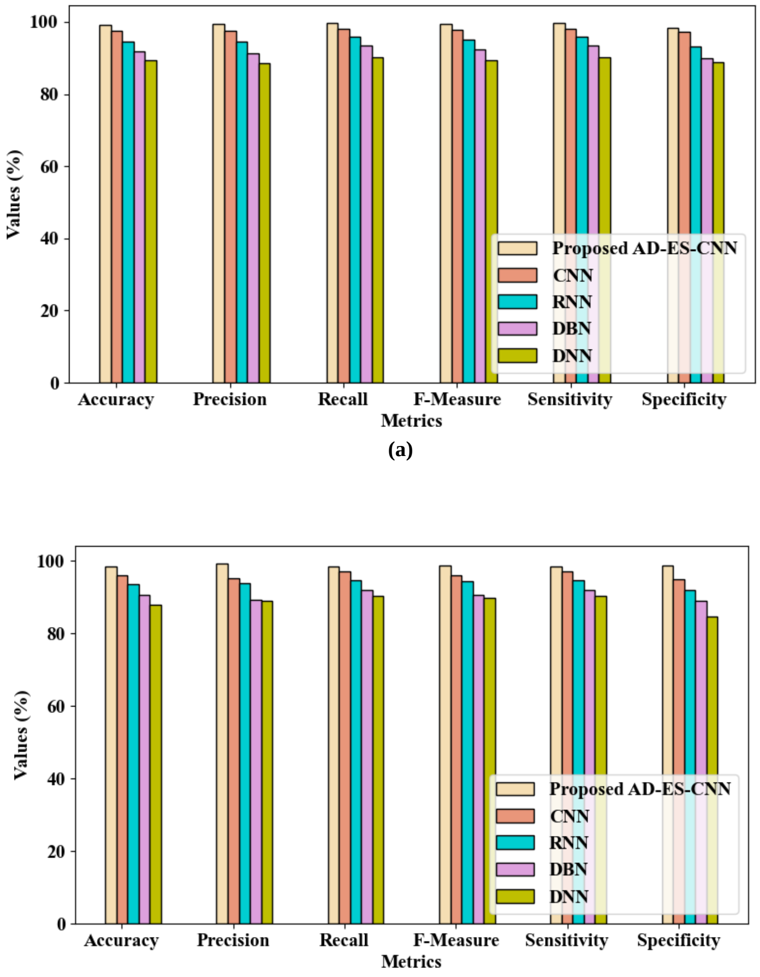
Fig. 5 Fitness vs. Iteration analysis for the Corel image dataset

As shown in Table 4, the proposed GM-TSOA achieved the highest fitness value (99.61%) at an iteration count of 50 for the Caltech 256 image dataset, outperforming existing methods. Figure 5 displays that for the Corel image dataset, the proposed GM-TSOA achieved a fitness of 99.43% at the iteration count of 50. Yet, the existing methods, such as TSOA, BOA, WOA, and ZOA, achieved fitness values of 95.83%, 94.63%, 92.77%, and 88.39% for the Caltech 256 dataset at 50 iterations, respectively. Likewise, the existing methods attained lower fitness values compared to GM-TSOA for the Corel image dataset. The GM-TSOA's higher performance was due to the usage of Gaussian mutation in the optimization algorithm.

4.2.5. Performance Analysis for Object Classification

The proposed AD-ES-CNN's performance is weighed against the prevailing CNN, Recurrent Neural Network (RNN), Deep Belief Network (DBN), and Deep Neural Network (DNN).

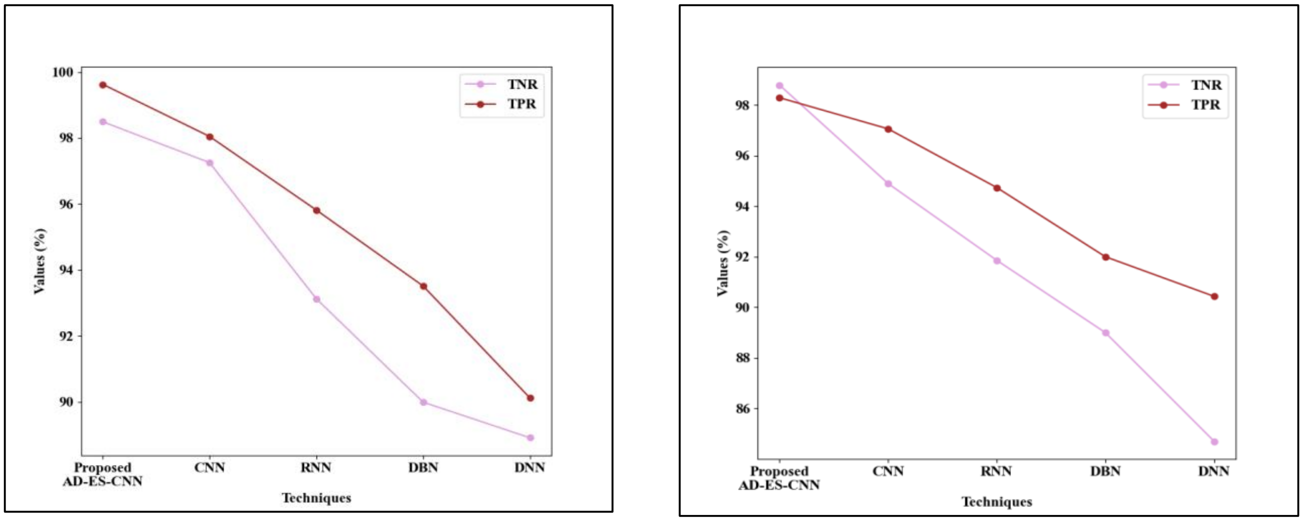
The proposed AD-ES-CNN's performance centered on accuracy, precision, recall, f-measure, specificity, and sensitivity using the Caltech 256 image dataset and the Corel image dataset is depicted in Figures 6 (a) and (b). According to Figure 6(a), for the Caltech 256 image dataset, the proposed AD-ES-CNN approach achieved 99.29% accuracy, 99.37% precision, 99.63% recall, 99.50% f-measure, 98.49% specificity, and 99.63% sensitivity, which were higher than the conventional approaches. As per Figure 6 (b), for the Corel image dataset, the proposed AD-ES-CNN approach achieved a higher accuracy of 98.5%, precision of 99.13%, recall of 98.29%, f-measure of 98.71%, specificity of 98.79%, and sensitivity of 98.29%. However, for both datasets, the existing DNN approach achieved very low performance on these metrics compared to other existing approaches like CNN, RNN, and DBN, and the proposed method. The proposed approach's higher performance was due to the Attention Drop regularization and E-Swish activation function used in the CNN architecture. Thus, the proposed AD-ES-CNN effectively classified the objects in CBIR systems.



(a)

(b)

Fig. 6 Performance analysis for object classification on (a) The Caltech 256 image dataset, and (b) The Corel image dataset.



(a) **(b)**

Fig. 7 Performance analysis of the proposed AD-ES-CNN classifier in terms of TNR and TPR using (a) Caltech 256 image dataset, and (b) Corel image dataset.

The performance of the proposed AD-ES-CNN classifier based on TNR and TPR using the Caltech 256 image dataset and the Corel image dataset is depicted in Figures 7 (a) and (b). As per Figure 7 (a), the proposed approach achieved 99.63% TPR and 98.49% TNR, which were higher than the existing approaches in the Caltech 256 image dataset. Figure 7 (b) displays that the proposed approach attained higher TPR (98.29%) and TNR (98.79%) values in the Corel image dataset. Yet, for both datasets, the TPR and TNR values of the existing CNN, RNN, DBN, and DNN classifiers were lower than those of the proposed approach. Thus, the proposed AD-ES-CNN effectively classified the objects.

4.3. Comparative Analysis

Here, the proposed research's performance is assessed with existing research works.

The proposed approach's performance is weighed against the present works based on accuracy, precision, and recall metrics, as depicted in Table 6.

Table 6. Comparative analysis of the proposed research with the existing research works

Authors' Name	**Methods**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**
[26]	CNN	92.35	-	-
[27]	CNN	81	-	-
[28]	Local Binary Patterns (LBP)	-	0.92	-
[29]	Local Polar Discrete wavelet transforms Feature (LPDF)	-	0.99	0.83
[30]	Local Binary Patterns (LBP)	98.71	-	90.54
Proposed	AD-ES-CNN	Caltech 256 image - 99.29 and Corel image- 98.5	Caltech 256 image - 99.37 and Corel image-99.13	Caltech 256 image -99.63 and Corel image-98.29

Here, the proposed approach achieved 99.29% accuracy, 99.37% precision, and 99.63% recall for the Caltech 256 image dataset. Likewise, the proposed approach achieved an accuracy of 98.5%, precision of 99.13%, and recall of 98.29% for the Corel image dataset, outperforming existing methods. The enhanced performance in CBIR was achieved by considering semantic segmentation using the G^{2}C approach, object classification using the AD-ES-CNN ranking and retrieval process, and the relevant feature selection from the extracted features using GM-TSOA. Overall, the proposed methodology effectively provided the similarity and ROI analysis for CBIR.

5. Conclusion

Here, ROI and similarity analysis for CBIR utilizing the AD-ES-CNN and EPL-Fuzzy approach is proposed. For feature selection, the proposed GM-TSOA achieved 99.61% (Caltech 256 image dataset) fitness at an iteration of 50. For semantic segmentation, the proposed G^{2}C achieved JI values of 0.045033 (Caltech 256 image dataset) and 0.054879 (Corel image dataset).

For object classification, the accuracy and precision of the AD-ES-CNN classifier were 99.29% and 99.37% for the Caltech 256 image dataset, as well as 98.5% and 99.13% for the Corel image dataset, respectively. Henceforth, the proposed methodology effectively provided the similarity and ROI analysis for CBIR. Yet, the research did not concentrate on the user behavior, preferences, and contextual information of the image retrieval systems.

5.1. Future Scope

In the future, the work should focus on user behavior, preferences, and contextual information for image retrieval systems and provide more personalized results.

Data Availability

Data available from the author on reasonable request.

Dataset Link

https://www.kaggle.com/datasets/jessicali9530/caltech256
https://www.kaggle.com/datasets/elkamel/corel-images

References

[1] Antonio M. Rinaldi, and Cristiano Russo, “A Content based Image Retrieval Approach based on Multiple Multimedia Features Descriptors in E-Health Environment,” *2020 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, Bari, Italy, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Lakshmana, P.V. Bhaskar Reddy, and Megha P. Arakeri, “Content based Image Retrieval System using Modified LSTM with Clustering and K-D Tree Indexing Techniques,” *2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS)*, Bangalore, India, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Palwinder Kaur, and Rajesh Kumar Singh, “A Panoramic view of Content-based Medical Image Retrieval System,” *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, London, UK, pp. 187-192, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Socratis Gkelios et al., “Deep Convolutional Features for Image Retrieval,” *Expert Systems with Applications*, vol. 177, pp. 1-47, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Rajesh Yelchuri et al., “Deep Semantic Feature Reduction for Efficient Remote Sensing Image Retrieval,” *IEEE Access*, vol. 11, pp. 112787-112803, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Naushad Varish et al., “Image Retrieval Scheme using Quantized Bins of Color Image Components and Adaptive Tetrolet Transform,” *IEEE Access*, vol. 8, pp. 117639-117665, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Sachendra Singh, and Shalini Batra, “An Efficient Bi-Layer Content based Image Retrieval System,” *Multimedia Tools and Applications*, vol. 79, no. 25-26, pp. 17731-17759, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yougui Ren et al., “A Scene Graph Similarity-based Remote Sensing Image Retrieval Algorithm,” *Applied Sciences*, vol. 14, no. 18, pp. 1-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Guanyuan Feng et al., “Hierarchical Clustering-based Image Retrieval for Indoor Visual Localization,” *Electronics*, vol. 11, no. 21, pp. 1-31, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jianan Bai et al., “Image Retrieval Method based on Visual Map Pre-Sampling Construction in Indoor Positioning,” *ISPRS International Journal of Geo-Information*, vol. 12, no. 4, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Zahra Tabatabaei et al., “WWFedCBMIR: World-Wide Federated Content-based Medical Image Retrieval,” *Bioengineering*, vol. 10, no. 10, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Massimo Salvi et al., “The Impact of Pre- and Post-Image Processing Techniques on Deep Learning Frameworks: A Comprehensive Review for Digital Pathology Image Analysis,” *Computers in Biology and Medicine*, vol. 128, pp. 1-24, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Arpit Kumar Sharma et al., “A Survey on Machine Learning based Brain Retrieval Algorithms in Medical Image Analysis,” *Health and Technology*, vol. 10, no. 6, pp. 1359-1373, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Manal M. Khayyat, and Lamiaa A. Elrefaei, “Manuscripts Image Retrieval using Deep Learning Incorporating a Variety of Fusion Levels,” *IEEE Access*, vol. 8, pp. 136460-136486, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Shiv Ram Dubey, “A Decade Survey of Content based Image Retrieval using Deep Learning,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2687-2704, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Rohit Raja, Sandeep Kumar, and Md Rashid Mahmood, “Color Object Detection based Image Retrieval using ROI Segmentation with Multi-Feature Method,” *Wireless Personal Communications*, vol. 112, no. 1, pp. 169-192, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Muhammad Kashif, Gulistan Raja, and Furqan Shaukat, “An Efficient Content-based Image Retrieval System for the Diagnosis of Lung Diseases,” *Journal of Digital Imaging*, vol. 33, no. 4, pp. 971-987, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Umer Ali Khan, Ali Javed, and Rehan Ashraf, "An Effective Hybrid Framework for Content based Image Retrieval (CBIR)," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26911-26937, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Silvester Tena, Rudy Hartanto, and Igi Ardiyanto, "Content-based Image Retrieval for Traditional Indonesian Woven Fabric Images using a Modified Convolutional Neural Network Method," *Journal of Imaging*, vol. 9, no. 8, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] M. Sivakumar, N.M. Saravana Kumar, and N. Karthikeyan, "An Efficient Deep Learning-based Content-based Image Retrieval Framework," *Computer Systems Science and Engineering*, vol. 43, no. 2, pp. 683-700, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Sarath Chandra Yenigalla, Karumuri Srinivasa Rao, and Phalguni Singh Ngangbam, "Implementation of Content-based Image Retrieval using Artificial Neural Networks," *Engineering Proceedings*, vol. 34, no. 1, pp. 1-6, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Le Gao et al., "Research on Image Classification and Retrieval using Deep Learning with Attention Mechanism on Diaspora Chinese Architectural Heritage in Jiangmen, China," *Buildings*, vol. 13, no. 2, pp. 1-19, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Shahbaz Sikandar, Rabbia Mahum, and AbdulMalik Alsalman, "A Novel Hybrid Approach for a Content-based Image Retrieval using Feature Fusion," *Applied Sciences*, vol. 13, no. 7, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Manoharan Subramanian et al., "Content-based Image Retrieval using Colour, Gray, Advanced Texture, Shape Features, and Random Forest Classifier with Optimized Particle Swarm Optimization," *International Journal of Biomedical Imaging*, vol. 2022, no. 1, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Nitin Arora, Aditya Kakde, and Subhash C. Sharma, "An Optimal Approach for Content-based Image Retrieval using Deep Learning on COVID-19 and Pneumonia X-Ray Images," *International Journal of Systems Assurance Engineering and Management*, vol. 14, no. S1, pp. 246-255, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] K. Karthik, and S. Sowmya Kamath, "A Deep Neural Network Model for Content-based Medical Image Retrieval with Multi-View Classification," *The Visual Computer*, vol. 37, no. 7, pp. 1837-1850, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[27] Shubham Agrawal et al., "Content-based Medical Image Retrieval System for Lung Diseases using Deep CNNs," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3619-3627, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Gabriel Da Silva Vieira, Afonso Ueslei Da Fonseca, and Fabrizzio Soares, "CBIR-ANR: A Content-based Image Retrieval with Accuracy Noise Reduction," *Software Impacts*, vol. 15, pp. 1-7, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[29] T. Sunitha, and T.S. Sivarani, "An Efficient Content-based Satellite Image Retrieval System for Big Data Utilizing Threshold based Checking Method," *Earth Science Informatics*, vol. 14, no. 4, pp. 1847-1859, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] Mona Ghahremani, Hamid Ghadiri, and Mohammad Hamghalam, "Local Features Integration for Content-based Image Retrieval based on Color, Texture, and Shape," *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 28245-28263, 2021. [CrossRef] [Google Scholar] [Publisher Link]