EAI Endorsed Transactions
on Intelligent Systems and Machine Learning Applications Research Article EAI.EU

Quality Aware Content-Based Image Retrieval Using QIM and Deep Learning for Big Visual Data

Amit Phadikar[1,*]

[1]Santal Bidroha Sardha Satabarshiki Mahavidyalaya, Goaltore, Paschim Medinipur, W.B., India

Abstract

Modern technology has made storing, sharing, and organizing huge amounts of data simple through the Internet of Things. Search engines and query-based retrieval databases made access to relevant data easy through ranking and indexing based on stored content. In this paper, a secure CBIR scheme based on watermarking is proposed. Firstly, the image owner embeds the watermark in the image using quantization index modulation (QIM) in the luminance (Y) color space. The watermarked images are then uploaded to the cloud server, which extracts image feature vectors. In this article, features derived from a pre-trained network model from a deep-learning convolutional neural network trained for large image classification have been used for the retrieval of similar images. The image similarity is calculated using Euclidean distance, and the precision (P) is used as the performance measure of the model that achieved nearly 100%. Extensive experiments are carried out, and assessment results reveal the outperforming result of the proposed technique compared to other related schemes. The scheme can be used in many applications that need CBIR, such as digital libraries, historical research, fingerprint identification, and crime prevention.

Keywords: Cloud Server, Content-Based Image Retrieval, Quantization Index Modulation, Deep Learning, Big Visual Data

Received on 17 October 2025, accepted on 29 November 2025, published on 02 December 2025

doi: 10.4108/eetismla.10624

[*]Corresponding author. Email: amitphadikar@rediffmail.com

1. Introduction

The huge amount of data due to digital media, smartphones, and likewise results in ‘Big data’. A major part of this visual data becomes publicly available from blogs, social networking websites, image and video-sharing websites, and wikis, often guided by explicit structured and unstructured text annotations. Since these data are continuously uploaded, the traditional ‘‘compute and storage’’ method is not able to handle the massive amount of data. In addition, most of the data is unstructured. However, it is accompanied by explicit structured data (e.g. geo-location and timestamps). This tendency leads a large number of people to access a large image database [1]. Image retrieval is a well-studied problem of image matching, where similar images are retrieved from a database with respect to a given query image. The similarity between the query image and the database images is used to rank the database images in decreasing order of similarity. Thus, the performance of any image retrieval method depends on the similarity computation between images. Ideally, the similarity score computation method between two images should be discriminative, robust, and efficient [2].

The requirement of CBIR has received extensive attention, and the huge number of solutions have been given in [3-6]. Alsmadi [3] proposed a combination of methods, i.e. clustering algorithm and the Canny edge method, to extract shape features. They also used YCbCr color with discrete wavelet transform and the Canny edge histogram to extract color features, and gray-level co-occurrence matrix to extract texture features. Yosr et al. [4] introduced a fuzzy similarity measure (FSM) motivated by near-sets theory and Type-2 Fuzzy logic. Then, they proposed three new IT-2 FSMs with mathematical justification to demonstrate that the proposed FSMs satisfy proximity properties. Singh et al. [5] proposed a Bi-layer Content-Based Image Retrieval (BiCBIR) method that consists of two modules. The very first module extracts the features of images from the database in terms of texture, color, and shape. The second module contains two sub-methods: in the first method all images are compared with the query image for texture and shape features, and indexes of 'M' most similar images to the given query image are retrieved. Khan et al. [6] used a structured matrix decomposition method to highlight the prominent area and then used two-dimensional principal component analysis (2DPCA) to extract features that result in faster recognition.

After the introduction and evolution of deep-learning neural network (DNN), the performance of CBIR has received a boost. This is because deep models allow us to extract higher-level features, in addition to the low-level features, from the image. This helps reduce the semantic gap mentioned above. Chen et al. [7], provided a survey on deep learning algorithms and techniques for instance retrieval. The survey was organized by deep feature extraction, feature embedding, and aggregation methods, and network fine-tuning strategies. Rani et al. [8] proposed a CBIR method using a Convolutional Neural Network (CNN) of type Residual Network (ResNet-18) architecture to accurately get the similarity score. The scheme offered an accuracy of 93.65% for 84 iterations. Mohammed et al. [9] introduced a deep learning approach for the efficient retrieval of images using robust deep features extracted from the VGG-19 architecture. The scheme involved fine-tuning a pre-trained network to adapt it to test the dataset by replacing the final layers. Finally, features are extracted from the 'fc7' layer. Euclidean distance is applied to calculate the closest distance between the query image and the features database. Xu et al. [10] proposed a scheme based on relevance feedback (RF) for CBIR using deep learning. When the results meet the demand of the user, the RF model stops and returns the final optimal results to users. Authors argued that deep learning, combined with the RF model, significantly outperforms only applying deep learning for CBIR tasks.

It is already mentioned that the number of images generated by all kinds of devices has greatly increased in recent years. Accordingly, content-based image retrieval (CBIR) technology research has generated wide attention and made remarkable advances. Images themselves are storage-consuming, and the CBIR technologies are typically of high computational complexity. Thus, there is a motivation to outsource the CBIR services to the cloud server [11]. The public cloud storage services provide cheap storage space, are computationally convenient, and have multiple access modes. Although the cloud storage service has great advantages, it is worth pondering the privacy/security problem it brings. The user defaults that the cloud service provider is untrustworthy. The urgent need for privacy protection has attracted experts to study secure outsourced CBIR schemes.

To solve the secrecy problems of outsourcing CBIR services, the proposed CBIR scheme mainly takes the following steps. Firstly, the user embeds a watermark in the image and the image is uploaded to the cloud server. In this CBIR scheme, the cloud server only provides storage and retrieval services. In this scheme, the features derived from a pre-trained network model from a deep-learning convolutional neural network trained for large image classification have been used for the retrieval of similar images. The resulting algorithm appears to achieve remarkable success in terms of retrieval accuracy and appears to outperform many contemporary state-of-the-art CBIR methods. Moreover, the algorithm is quite fast.

Contributions: The major contributions are enumerated as follows:

(1) A specially designed image watermarking scheme is proposed for image quality access control in lifting-based discrete wavelet transform (DWT) using QIM.

(2) The scheme combines digital watermarking (QIM) and deep learning-based CBIR—a powerful hybrid technique that resolves the issues of both accuracy and security.

(3) In the proposed scheme, users only need to complete the work of image watermarking. The content-based image retrieval will be completed by the cloud server, which will largely reduce the user's workload.

(4) Large visual datasets are handled by cloud computing off-loading, which makes it scalable for Internet of Things applications.

(5) QIM watermarking offers the advantage of controlling the access through watermark decoding, thus improving the security of data and management of the quality of images.

(6) Feature extraction from the images is performed using pre-trained models ResNet-18 and GoogLeNet to lower the computational cost. The retrieval performance and security are tested and analysed, respectively.

The article is organized as follows: Section 3 presents details of the proposed algorithm. Section 4 contains the results of extensive experiments. In Section 4, we present the concluding remarks and some future directions.

2. Proposed Quality-Aware Content-Based Image Retrieval Scheme

The overall quality-aware content-based image retrieval scheme consists of three stages: (1) the data embedding stage by the image owner, (2) deep learning-based image retrieval at the cloud server, and (3) watermark decoding at the user's side for improved quality images that act as quality access control for retrieved images. The block schematic representation of the proposed quality-aware content-based image retrieval scheme is shown in Fig. 1.

2.1 Image Watermarking by Image Owner

In our scheme, a logo image is used as the watermark (W) and is permuted to obtain a pseudo-random sequence (W') using Eq. (1) & (2):

$$W = \{w\,(i,j), 1 \le i \le n, 1 \le j \le n, w\,(i,j) \in (0,1)\} \qquad (1)$$

The symbol 'K' is the chaotic binary sequence. K is used as a secret key.

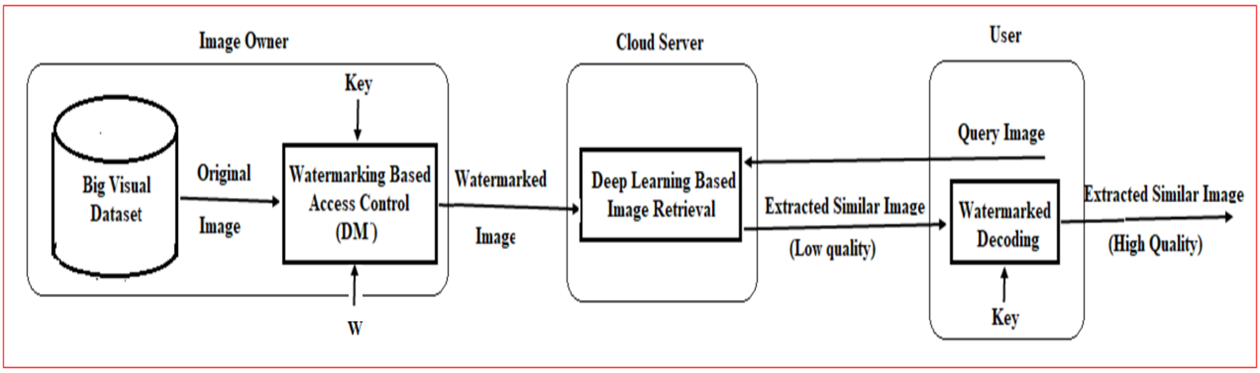


Figure 1. Block diagram of the proposed scheme.

$$K = \{k\,(i,j), 1 \le i \le n, 1 \le j \le n, k\,(i,j) \in (0,1)\} \qquad (2)$$

$W' = W \oplus K$ where $\oplus$ denotes the XOR operation. Then image watermarking is done in different steps as follows [12]:

Step 1: Color Space Transformation: The host color image is first transformed from red, green, and blue (RGB) into luminance/chrominance color space such as luminance(Y), chrominance-blue (C_b), and chrominance-red (C_r).

Step 2: Image Transformation: Lifting-based *n*-level 2D-DWT is performed on the luminance(Y) component of the image.

Step 3: Coefficients Selection Criteria for One Bit of Watermark Embedding: One bit of permuted watermark (W') is inserted into different energy levels. For the 3-level decomposition of an image total of 24 coefficients (four coefficients from low-low(LL), high-low(HL), low-high(LH), and high-high(HH_3), four coefficients from high-high(HH_2), 16 coefficients from high-high(HH_1) are selected for one bit of watermark insertion.

Step 4: Steps Size Selection for QIM-based Watermarking: Different types of step sizes (*Δ*) are selected for the tiles of different energy levels. We select a small step-size for LL sub-band since it contains most visual information of the image and a large step-size (*Δ*) for HH_1 tiles. If the image is decomposed by 3-level DWT then the types of step-size will be five (5) as we take the same step-size for both HL and LH sub-band.

Step 5: Binary Dither Generation for Each Sub band: Binary dither (d_b) of length 'L' for a sub-band is generated pseudo-randomly using the 'key' by choosing $d_{b\,,q}(0)$ as:

$$d_{b,q}(0) = (rand(\,key) \times \Delta_b) - \frac{\Delta_b}{2}, rand(key) \in (0,1) \qquad (3)$$

$$d_{b,q}(1) = \begin{cases} d_{b,q}(0) + \frac{\Delta_b}{2}, if\; d_{b,q}(0) < 0 \;\; and\; 0 \le q < L \\ d_{bq}(0) - \frac{\Delta_b}{2}, if\; d_{b,q}(0) \ge 0 \end{cases} \qquad (4)$$

The length (L) depends on the number of coefficients to be considered in different sub-band(b) to insert a bit.

Step 6: Watermark Insertion: Each bit of W' is embedded into the selected coefficients of different energy levels. The bit is embedded according to Eq. (5) as follows [12,13].

$$S_q = \begin{cases} Q\left(X_q - k'.d_{b,q}(0), \Delta_b\right) + d_{b,q}(0), if\;\; W'(i,j) = 0 \\ Q\left(X_q + k'.d_{b,q}(1), \Delta_b\right) - d_{bq}(1), if\;\; W'(i,j) = 1 \end{cases} \qquad (5)$$

Where *'S'* and *'X'* are the DWT coefficient of the watermarked and the host image respectively and 'Q' is the quantizer with step Δ_b for type *'b'*. The symbol k' represents the degree of quality degradation for quality access control and is set to two (2) in this paper. After watermark embedding, inverse lifting-based DWT is applied to the Y component. The luminance(Y), chrominance-blue (C_b), and chrominance-red (C_r) are merged and then converted to RGB color space and a watermarked image is formed.

Step 7: Steps 1 to Step 6 are repeated until all images in the image database are watermarked and uploaded to the cloud server.

Step 8: Security of the Secret Key (K): The secret key '*K*' is encrypted using public key (P_u) cryptography (like RSA) so that it cannot be disclosed to unauthorized user during transmission.

2.2 Deep Learning-Based Image Retrieval at Cloud Server

The proposed scheme reduces the user's computational burden by transferring feature extraction and indexing tasks to the cloud server. When the image owner uploads the watermarked image database to the cloud server, the cloud server extracts image features directly from watermarked images using Deep Learning. Let **O** be the number of object classes. Given this configuration, the Deep Neural Network

(DNN) learns to estimate the probabilities that an image represents an object class as [14,15]:

$$f(\boldsymbol{Y}) = \left(P(C_1|\boldsymbol{Y}), \ldots \ldots P(C_O|\boldsymbol{Y})\right) \in [0,1]^{O} \tag{6}$$

where **Y** is an input image, and C_i is the i^{th} object class, with $i \in \{1 \ldots O\}$. Let us note $\mathbf{Y}_{\text{Query}}$ the query image, and $\mathbf{Y}_{\text{j}}$ the training database images, where $j \in \{1 \ldots M\}$ and M is the number of images in the database. Then the database images are ranked according to the Euclidean distances between the probability vector $f\,(\mathbf{Y}_{\text{Query}})$ associated with the query image, and the probability vector $f(\mathbf{Y}_j)$ associated with the database images. For example, the most similar image in the database can be obtained as follows:

$$S = arg_{j \in \{1 \ldots M\}}^{\ \ min} \left\| f\left(Y_{Query}\right) - f\left(Y_j\right) \right\| \tag{7}$$

It is then possible to obtain the k most similar images in the database, as ranked by $\left\| f\left(Y_{Query}\right) - f\left(Y_j\right) \right\|$. In the proposed scheme, we have used *ResNet-18* and G*oogLeNet* pre-trained networks to achieve the goal. The ResNet18 is used as it successfully solves many problems related to image processing and classification [16,17]. Moreover, ResNet18 reduces computational costs and time complexity. On the other hand, G*oogLeNet* offers high accuracy using the final feature vector.

2.3 Watermark Decoding by User for Quality Access Control

Step 1: The same key 'K' is used for the generation of the dither that was used at the time of watermark embedding. The authorized user having the access right of the private key (P_r) decrypts the encrypted secret key $P_u(K)$ using Eq. 8.

$$K = P_r(P_u(K)), \tag{8}$$

where $P_u(K)$ denotes the encryption of K with public key P_u and $P_r(P_u(K))$ denotes the decryption of $P_u(K)$ with private key P_r.

Step 2: Steps 1 to 5 of Section 2.1 are performed on the retrieved images found by Section 2.2.

Step 3: Watermark Bit Extraction: For all of the retrieved images watermark bits are extracted based on the principle of minimum distance decoding to determine which quantizer has been used at the encoder side. A watermark bit (ẃ (i,j)) is decoded by examining the coefficient of different sub-band of luminance(Y) components using the following rule.

$$\begin{aligned} A &= \textstyle\sum_{q=0}^{L-1}(|\,\mathrm{Q}\,(\mathrm{y}_{\,q} - \ d_{b,q}(0), \Delta_b) + d_{b,q}(0) - \mathrm{y}_q|) \\ B &= \textstyle\sum_{q=0}^{L-1} |\,\mathrm{Q}\left(\mathrm{y}_{\,q} + \ d_{b,q}(0), \Delta_b\right) - d_{b,q}(0) - \mathrm{y}_q|) \end{aligned} \tag{9a}$$

$$\begin{aligned} &\text{if A} < \text{ B }, \widehat{\mathrm{w}}(\mathrm{i},\mathrm{j}) = 0 \\ &\quad else\ \widehat{\mathrm{w}}(\mathrm{i},\mathrm{j}) = 1 \end{aligned} \tag{9b}$$

where 'y' is the received lifting-based wavelet transformation of the luminance component.

Step 4: Watermarked Noise Cancellation for Access Control: Self-noise due to watermark embedding is eliminated to get a better-quality image using Eq. (10).

$$\acute{y}_q = \begin{cases} y_q + (k' - 1).\, d_{b,q}(1); & if\ \widehat{\mathrm{w}}(\mathrm{i},\mathrm{j}) = 0 \\ y_q - (k' - 1).\, d_{b,q}(1); & if\ \widehat{\mathrm{w}}(\mathrm{i},\mathrm{j}) = 1 \end{cases} \tag{10}$$

where $\acute{y}_q$ is the watermarked signal after noise elimination. Then inverse lifting-based DWT is applied on the luminance(Y) component and then luminance(Y), chrominance-blue ($\mathrm{C_b}$), and chrominance-red ($\mathrm{C_r}$) components are merged. Then $\mathrm{YC_bC_r}$ is converted to RGB color space to get a relatively high quality of retrieved watermarked images.

3. Experiments and Results

In this part, we express broad tests to demonstrate the usefulness of various pre-train deep learning models for the extraction of pictures from picture databases. Intel(R) Pentium (R) CPU, 4 GB RAM, 2.80 GHz processor of system configuration, and MATLAB 2021 are used during the present investigation. We have performed tests on the widely accessible benchmark Corel-1k dataset. The Corel-1k dataset consists of a thousand images of 10 semantic categories (such as buildings, beaches, elephants, etc.), where each group contains a hundred pictures. The evaluation was primarily done to the Corel-1K dataset because it is a well-known, standardized, and computationally efficient benchmark that allows direct comparison with existing CBIR methods. When developing or verifying new models, its moderate size makes it computationally manageable for testing and enables quick experimentation and parameter customization. The Corel-1K dataset is well-annotated, openly accessible, and doesn't need a particular license. Before expanding to bigger or more complicated datasets, it offers a clean and balanced testbed for preliminary validation. Many CBIR papers use Corel-1K for proof-of-concept experiments—to verify the effectiveness of the proposed feature extraction, embedding, or retrieval techniques before scaling to larger "big data" scenarios. Some of the images are shown in Fig. 2(a-j). Figs. 2(k) & (l) are the watermark logo and encrypted watermark used in this scheme, respectively.

We have used ResNet-18 and GoogLeNet pre-trained networks during performance evaluation. *ResNet-18:* It is a convolutional neural network that is 18 layers deep. The pre-trained version of the network trained on more than a million images from the ImageNet database [18]. *GoogLeNet:* It is a convolutional neural network that is 22 layers deep. The network trained on ImageNet classifies images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. Furthermore, when trained or refined on limited datasets, deep models with a lot of parameters (like ResNet-50 and VGG-19) are prone to overfitting. With fewer layers and parameters, ResNet-18 is relatively lightweight, making it more appropriate for small-scale datasets while retaining rich visual features. On the other hand, the GoogLeNet pre-trained model was optimal for the Corel-1K dataset because it offers an effective balance between multi-scale feature extraction, model compactness, and generalization capability**.** Its inception architecture captures diverse spatial information while avoiding overfitting on small datasets, and its pre-trained ImageNet features transfer efficiently to Corel-1K, producing discriminative and computationally efficient embeddings for image retrieval.

In CBIR, the implementation details (parameter settings, network fine-tuning steps, and key generation process) are critical and describe as follows**.** For GoogLeNet the parameters are as follows: It has 144 layers, 170 connections, the used activation function is Rectified Linear Unit (ReLU); Pooling Layers- Max-pooling (after initial convolutions) and a final average pooling. There is 1000 fully connected layer, the used filter sizes in Inception are 1×1, 3×3, and 7×7 convolutions, plus 3×3 max pooling. The classification output is 1×1×1000. For ResNet18 the parameters are as follows: It has 71 layers, the used activation function is Rectified Linear Unit (ReLU), the used Pooling Layers are Max-pooling and a final average pooling. There are 1000 fully connected layer and the Filter Sizes are 1×1, 3×3, and 7×7 convolutions, plus 3×3 max pooling. The classification output is 1×1×1000.

We also extend our evaluation to Caltech 256 Image Dataset [19] and ImageNet Dataset [20]. **Caltech 256 Image Dataset:** There are 30,607 images in this dataset spanning 257 object categories. Object categories are extremely diverse, ranging from grasshopper to tuning fork and available at Kaggle [21]. The scheme is tested on a sample of 25 class for ease of implementation. **ImageNet Dataset:** This dataset spans 1000 object classes and contains 1,281,167 training images, 50,000 validation images and 100,000 test images. ImageNet 1000 (mini) contains 1000 samples from ImageNet and available at Kaggle [21]. The scheme is tested on a sample of 100 class for ease of implementation.

In CBIR, the precession (P) is expressed as the quantity of extracted related pictures/images with respect to total extracted images. Precession is expressed as:

$$P = \frac{E}{T} \tag{11a}$$

where, the symbol 'E' is the related extracted pictures and 'T' is the entire amount of retrieved images. The ratio of the retrieved relevant images (A) to the total number of relevant photos in the database (C) is known as the recall (R) in image retrieval. The definition of recall is:

$$R = \frac{A}{C} \tag{11b}$$

F-score can be defined as:

$$F = 2 \times \frac{(P \times R)}{(P + R)} \tag{11c}$$

The present scheme uses precession, recall and F-score to calculate the efficiency of retrieval performance.

3.1. Effectiveness of Image Watermarking

Fig. 3 shows sample watermarked images from each category that are uploaded to the cloud server for image retrieval. The present study uses Peak Signal to Noise Ratio (PSNR) and Mean Structure Similarity Index Measure (MSSIM)[22] as a distortion measure for the watermarked image under inspection with respect to the original image. The high PSNR and MSSIM values indicate that fidelity is preserved for the watermarked image.

The scheme effectively combines Quantization Index Modulation (QIM)-based watermarking with deep learning feature extraction, offering a dual benefit of data security and efficient image retrieval. Theoretically the QIM can impact the discriminative capacity of deep features because the quantization and embedding process modifies feature values and potentially alters feature distribution. However, the impact can be reduced or removed by carefully optimizing parameters (quantization step size, embedding strength, key design), preserving retrieval performance while enhancing robustness or watermarking capacity. In our scheme, the watermark information is embedded into the HH subbands of all levels and LL subbands of the 3rd level. The choice of embedding the watermark information into the HH subbands of all levels and LL subbands of the 3rd level was motivated by the experimental tests, as this one offers the best compromise between robustness and invisibility. The relatively high value of PSNR (in dB) and MSSIM values highlights the imperceptibility of the proposed scheme. It is also to be noted that as there is no such deviation in structural information (MSSIM) due to information embedding, the proposed QIM did not impact the discriminative capacity of deep features and did not alter feature distribution that impact CBIR. As a result, the data embedding scheme did not influence the CBIR outcome. If one increases the step-size of QIM to improve the security to high value, this will ultimately decrease the fidelity of the watermarked images that finally effects its commercial values with low or no use. We select small step-size for the coefficients in *LL* subband since it contains most visual information of the image and large step size (*Δ*) for HH_1 subband. The step sizes (*Δ*) for dither are taken as 11, 12,13,14,15 for LL, HL &LH, HH_3, HH_2 and HH_1 respectively. The Table shows the variation of PSNR and MSSIM due to change of step-size (*Δ*). The Table 1 shows the variation of PSNR and MSSIM due to change of step-size (*Δ*).

Table 1: Variation of PSNR and MSSIM due to change of step-size (*Δ*).

Values of step-size (*Δ*).	**Before Decoding**		**After Decoding**		**NCC**
	PSNR (dB)	**MSSIM**	**PSNR (dB)**	**MSSIM**	
{11, 12,13,14,15}	31.67	0.96	33.69	0.97	1
{12, 13,14,15,16}	30.86	0.96	33.04	0.97	1
{13, 14,15,16,17}	29.87	0.95	32.18	0.97	1
{14, 15,16,17,18}	29.29	0.94	31.76	0.96	1

The scheme is also tested for robustness evaluation against common attacks (e.g., compression, cropping, or geometric transformations) available at StirMark 4.0 benchmark [23].

The values in the Table 2 depicts that the scheme is robust to common image processing operations.

Table 2: Experimental results with StirMark 4.0

Attacks	NCC
Median filter 3×3	1.00
Median filter 5×5	1.00
Median filter 7×7	0.84
Rotation-scaling 0.25	0.86
Rotation-scaling -0.25	1.00
Rotation-cropping 0.25	1.00
Rotation-cropping -0.25	1.00
Rotation_0.25	1.00
Rotation_5	1.00
Rotation_90	1.00
LATESTRNDDIST_1	0.84
LATESTRNDDIST_1.05	0.87
Remov_lines_10	1.00
Remov_lines_50	0.97
Remov_lines_70	1.00
Remov_lines_100	1.00
JPEG_80	1.00
JPEG_50	0.96
Cropping_50	1.00
Cropping_75	1.00
AFFINE_2	0.74
AFFINE_4	0.87
AFFINE_6	0.86
CONV_1	0.82

3.2. Retrieval Accuracy

Table 3 depicts the performance of the proposed scheme for different pre-trained neural networks along with retrieval latency in Table 4. The results show that GoogLeNet offers relatively better outcome in term of precision and recall, with a moderate increase in retrieval latency. Table 3 also presents the performance of the proposed scheme for different pre-trained neural networks in term of mean average precision (mAP) for various database images. The results in Table 3 depicts that GoogLeNet offers batter performance than ResNet-18 for CBIR in term of mean average precision (mAP). Table 5 shows the average performance of the proposed scheme for different pre-trained neural networks like ResNet-18 and *GoogLeNet* in terms of precession for different categories of images along with various similar works found in the literature. It is evident from Table 5 that the proposed scheme offers quite better results than the related work. The numerical values in Table 5 are obtained as the average value of 100 independent experimentations conducted over a large number of benchmark images having varied image characteristics. Fig. 4 shows the result for *ResNet-18* and *GoogLeNet* respectively, for query image Bus. Fig. 5 (a)-(j) shows the recovered image for each category by the authentic user along with PSNR and MSSIM values. It is seen that an authentic user who has the secret key(K) can retrieve better-quality images than the normal user. Fig. 5(k) shows the extracted watermark for all categories of images along with the Normalized Cross Correlation (NCC) value. Table 6 lists the comparison in terms of precision. The results in Table 6 depicts that the scheme offers more efficient result than others. Fig. 6 and 7 shows the result for various query images from Caltech256 (American-Flag), and ImageNet (n01443537) database.

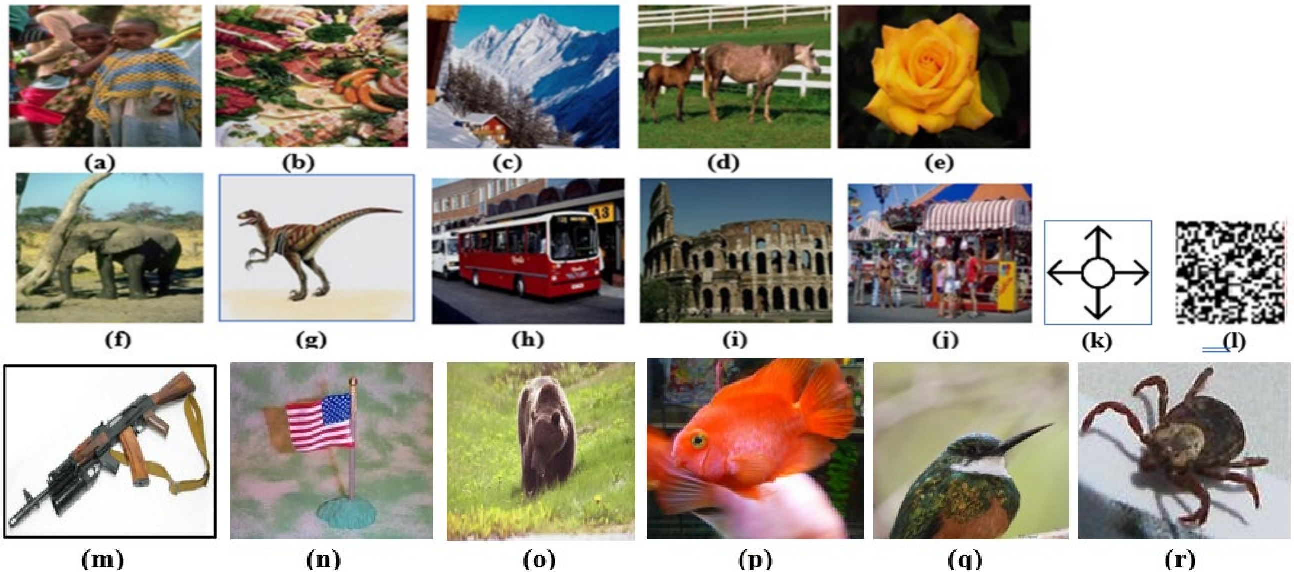


Figure 2. Test images (Corel-1k dataset): (a) People, (b) Foods, (c) Mountains, (d) Horses, (e) Roses, (f) Elephants, (g) Dinosaurs, (h) Buses, (i) Buildings, (j) Beaches. (k): Watermark image(W); (l): Encrypted

Watermark (W'). Test images (Caltech 256 dataset): (m) Ak47, (n) American-Flag, (o) Bear; Test images (Image Net dataset): (p) n01443537, (q) n01843065, (r) n01776313.

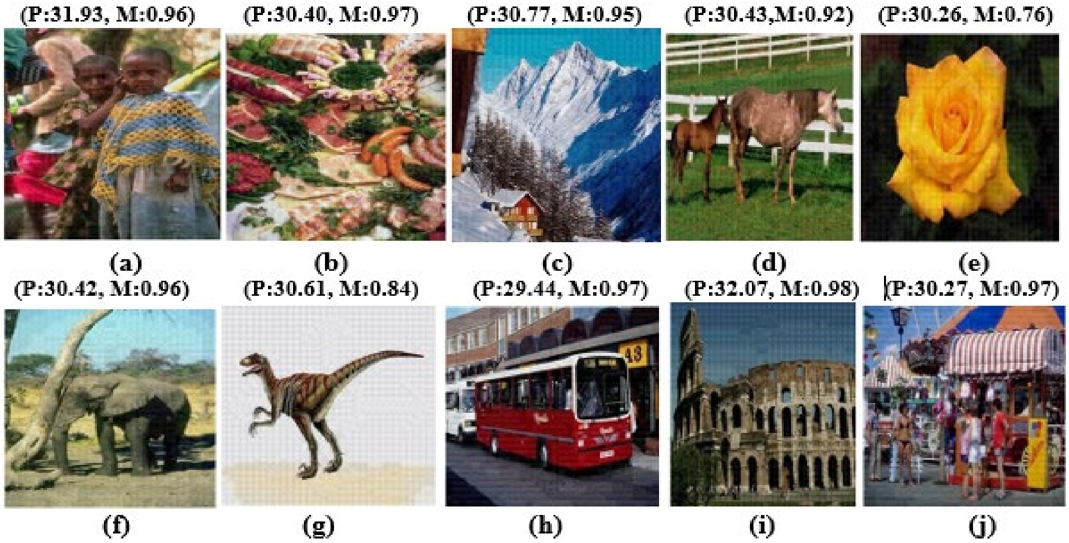


Figure 3. Watermarked images (Corel-1k dataset): (a) People, (b) Foods, (c) Mountains, (d) Horses, (e) Roses, (f) Elephants, (g) Dinosaurs, (h) Buses, (i) Buildings, (j) Beaches.

Table 3: Performance of the proposed scheme for different pre-trained neural networks.

Dataset	**Class**	**Precision**		**Recall**		**F1-score**	
		ResNet-18	GoogLeNet	ResNet-18	GoogLeNet	ResNet-18	GoogLeNet
Corel-1K	Beaches	100	100	60	60	63.31	75
	Buses	100	100	100	100	100	100
	Roses	100	100	95	95	97.44	97.44
	Dinosaurs	100	100	97	97	98.48	98.48
	People	100	100	77	77	87.01	87.01
	Foods	100	100	84	84	91.30	91.30
	Mountains	93.75	100	62	62	64.40	74.4
	Horses	100	100	76	76	86.36	86.36
	Elephants	100	100	63	63	77.30	75.12
	Buildings	100	100	55	55	67.39	62.73
	Average	**99.37**	**100**	**76.9**	**76.9**	**83.29**	**84.78**
Caltech-256	Ak47	89	100	94.18	100	73	84.39
	American-Flag	75	100	85.72	100	69	81.66
	Bear	76	100	86.36	100	68	80.95
	Average	**80**	**100**	**88.75**	**100**	**70**	**82.33**
ImageNet	n01443537	97	100	98.48	100	61	75.77
	n01843065	100	100	100	100	100	100
	n01776313	100	100	100	100	100	100
	Average	**93**	**100**	**99.49**	**100**	**87**	**91.92**
Mean Average Precision (mAP)		**90.79**	**100**	**88.38**	**92.3**	**80.09**	**86.34**

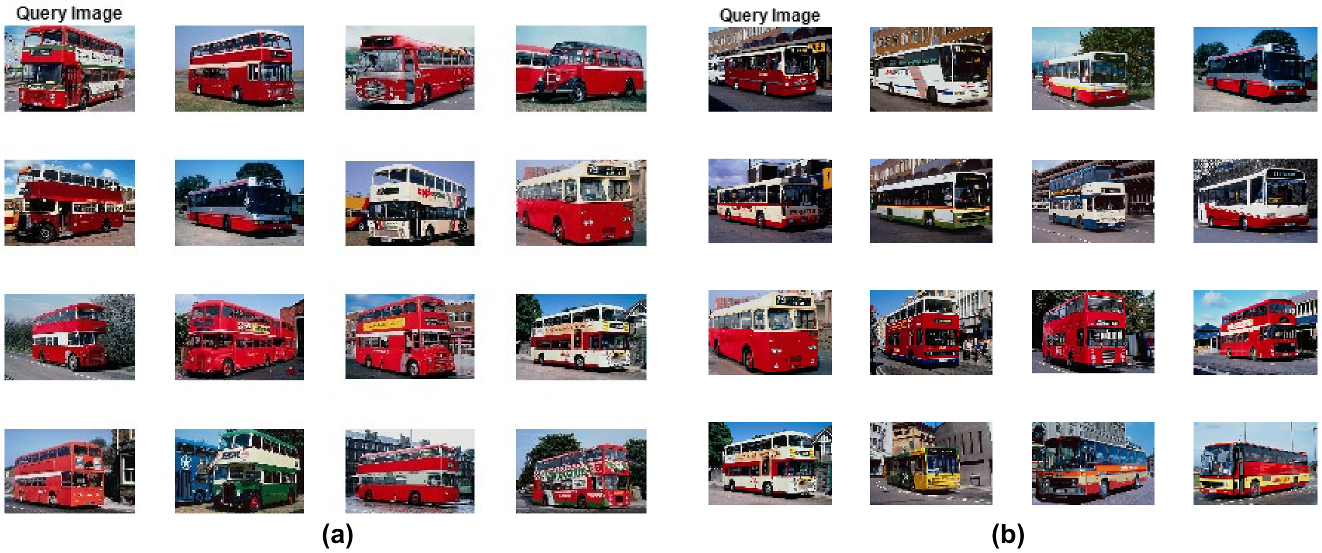
Table 4: The retrieval latency (in Second).

	Corel-1K		**Caltech-256**		**ImageNet**	
Size of Database	GoogLeNet	ResNet18	GoogLeNet	ResNet18	GoogLeNet	ResNet18
Feature Database Creation	137	111	299.90	258.05	351	280.62

Image Retrieval	7.9	8.7	46.00	38.00	66.73	52.35
Total	**144.9**	**119.7**	**345.9**	**296.05**	**417.73**	**332.97**

Table 5: Performance of the proposed scheme for different pre-trained neural networks along with related work.

Class	**Maji et al. [14]**	**Hamreras et al. [15]**	**Ramanjaneyulu et al. [24]**	**Proposed**	
				ResNet-18	GoogLeNet
Beaches	96.60	90.00	74.62	100	100
Buses	100	100	88.27	100	100
Roses	97.25	96.67	96.95	100	100
Dinosaurs	100	100	98.55	100	100
People	79.35	93.33	81.91	100	100
Foods	95.55	96.67	88.23	100	100
Mountains	98.95	96.62	91.62	93.75	100
Horses	99.90	100	91.65	100	100
Elephants	100	100	89.54	100	100
Buildings	93.55	96.62	80.84	100	100
Average	**96.11**	**96.99**	**88.21**	**99.37**	**100**



(a) (b)

Figure 4. Retrieval results for query image bus: (a) ResNet-18; (b) Google Net.

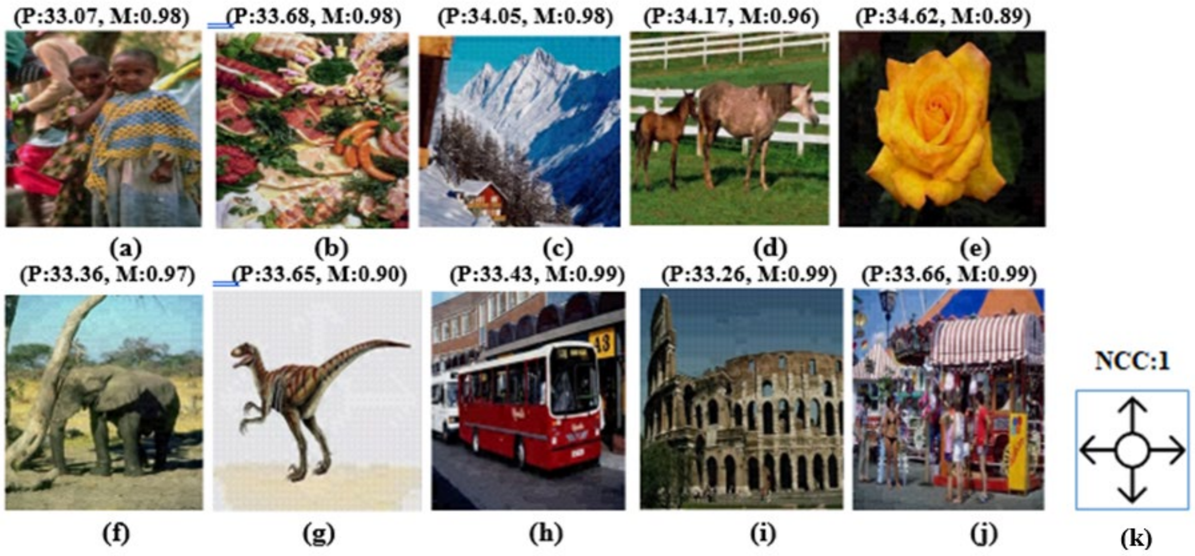
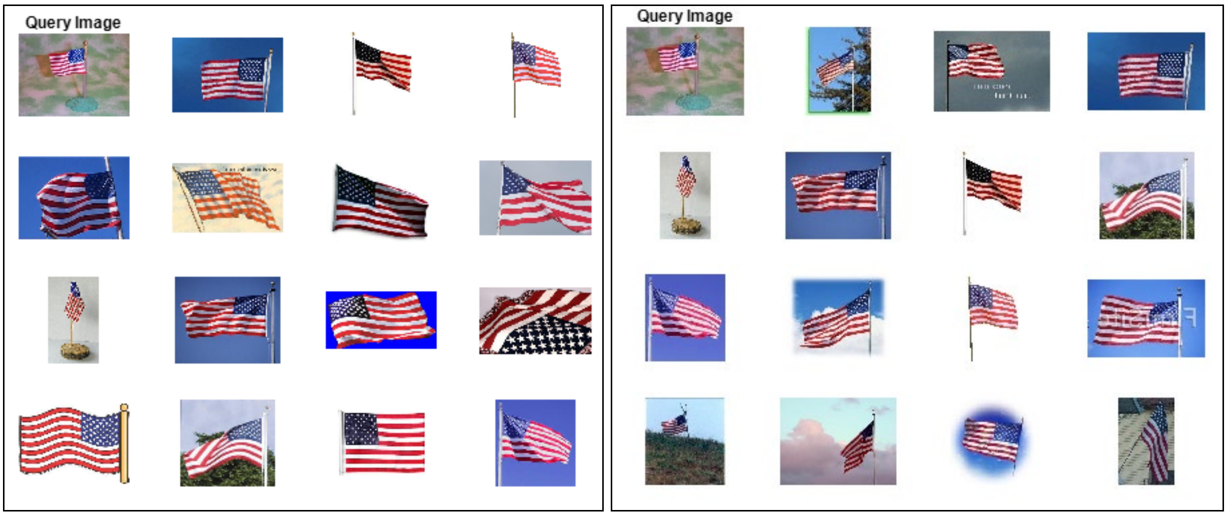
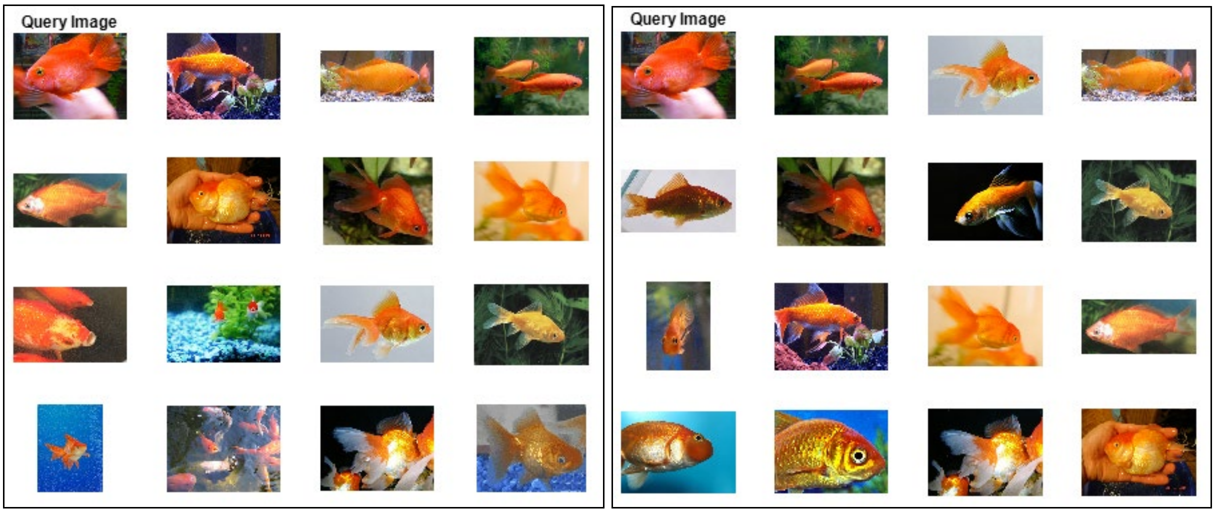


Figure 5. Recovered watermarked images (Corel-1k): (a) People, (b) Foods, (c) Mountains, (d) Horses, (e) Roses, (f) Elephants, (g) Dinosaurs, (h) Buses, (i) Buildings, (j) Beaches. (k): Extracted watermark.



(a) (b)

Figure 6. Retrieval results for query image Caltech256 (American-Flag): (a) ResNet-18; (b) Google Net.



(a) (b)

Figure 7. Retrieval results for query image ImageNet (n01443537): (a) ResNet-18; (b) Google Net.

Table 6: Comparison of precision with existing techniques.

Ref	Year	Dataset Used	%Precision
Singh et al. [25]	2020	Corel1K	92.2
Sikha et al. [26]	2021	WANG	80.3
Maji et al. [14]	2021	Imagedb2000, Caltech101, corel1K	96.11
Taheri et al. [27]	2022	Corel1K, Corel5K, caltech256	99.4
Kenchappa et al. [28]	2022	Corel, VOC	96.03
Kabir et al. [29]	2022	Corel10K, CAFIR10	98.
Salih et al. [30]	2023	Corel1K	86.06
Rastegar et al. [31]	2023	OT, caltech101, corel1K	98.78

Proposed (ResNet-18)	-	Corel1k	99.37
Proposed (GoogLeNet)	-	Corel1k	100

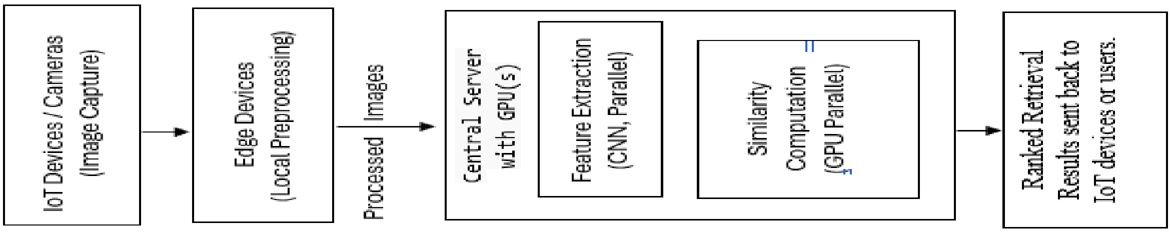

IoT Devices / Cameras (Image Capture)
Edge Devices (Local Preprocessing)
Processed Images
Central Server with GPU(s)
Feature Extraction (CNN, Parallel)
Similarity Computation (GPU Parallel)
Ranked Retrieval Results sent back to IoT devices or users.

Figure 8. GPU-parallelized CBIR workflow in IoT.

3.3 Efficiency

Efficiency is a significant measurement standard, and it includes the time consumption of image watermarking, index construction, and image searching.

- *The Time Consumption of Image Watermarking:* Our scheme is based on a lifting-base DWT method that is two times faster (though complexity is O (n)) than normal DWT and requires less amount of memory. So, the scheme is efficient for real-time implementation of the quality-aware CBIR scheme.
- *The Time Consumption of Image Retrieval:* As the index construction is done by a pre-trained network model from a deep-learning convolution at the cloud server, it seems to work faster without sacrificing retrieval performance. It reduces the computational load on the user.

The retrieval latency is presented in Table 4. It is cleared from the Table 4 that GoogLeNet takes large time for execution than ResNet-18. It is also cleared that the execution time increases with the increase in image database size.

3.4 Parallelization and GPU Optimization in CBIR Systems

The proposed system involves watermark embedding, feature extraction, and similarity computation. The each of the steps are potentially resource-intensive in large-scale deployments. To overcome the problem, one may think about parallelization and GPU optimization in CBIR systems that can be incorporated for scalability of IoT applications, enabling fast, efficient, and simultaneous processing of large volumes of images from distributed sensors. This ensures that retrieval latency remains low while handling growing datasets and query loads. Fig. 8 shows an illustrative diagram for GPU-parallelized CBIR workflow in IoT. The IoT devices/cameras capture images in real-time. The edge devices perform the pre-processing and optionally extract lightweight features to reduce bandwidth. The central GPU server extracts the deep features from images using CNNs in parallel and computes similarity against database features using GPU-accelerated parallel computation. Then ranked results are sent back to IoT devices or end users. The parallelization and GPU optimization ensure the low latency and scalability for real-time CBIR in IoT networks.

4. Conclusion and Scope of Future Works

In this paper, a secure CBIR scheme is proposed using QIM watermarking. Feature extraction from the images is performed using pre-trained models i.e. ResNet-18 and GoogLeNet to lower the computational cost. This study proves that, by using pre-trained deep learning (DL) models, better features can be extracted. The scheme finally affirms that the incorporation of pre-trained deep learning models not only enhances CBIR but also makes it more efficient in the process of watermarking that keeps the data safe. The scheme provides better results in terms of precision concerning the manual feature descriptors derived from traditional methods. In future the scheme can be extended for multiple architectures (e.g., VGG19, ResNet, Inception) to evaluate feature robustness and invariance to watermark embedding.

5. Ethics declarations

5.1 Ethical Approval

The submitted work is original and has not been published elsewhere in any form or language. This article contains no studies with human participants or animals performed by the author.

5.2 Data Availability Statement

The author confirms that the data supporting the findings of this study are available within the article.

5.3 Authors Contributions

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

5.4 Funding

No funding was received for this article.

5.5 Competing Interests

The author declares no conflict of interest.

References

[1] S.R. 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, p. 2687-2704, 2022.

[2] A. Latif, A. Rasheed, U. Sajid, Jameel Ahmed, Nouman Ali, Naeem Iqbal Ratyal, Bushra Zafar, Saadat Hanif Dar, Muhammad Sajid, and Tehmina Khalil, "Content-based image retrieval and feature extraction: a comprehensive review," Mathematical Problems in Engineering, vol. 2019, p. 1-21, 2019.

[3] M.K. Alsmadi, "Content-based image retrieval using color, shape and texture descriptors and features," Arabian Journal for Science and Engineering, vol. 45, no. 4, p. 3317–3330, 2020.

[4] G. Yosr, N. Baklouti, H. Hagras, M. Ben Ayed, and A. M. Alimi, "Interval type-2 beta fuzzy near sets approach to content-based image retrieval," IEEE Transactions on Fuzzy Systems, vol. 30, no. 3, p. 805-817, 2022.

[5] S. Singh, and S. Batra, "An efficient bi-layer content-based image retrieval system," Multimedia Tools and Applications, vol. 79, no. 25-26, p. 17731–17759, 2020.

[6] A. Khan, and A. Jalal, "A visual saliency-based approach for content-based image retrieval," International Journal of Cognitive Informatics and Natural Intelligence, vol. 15, no. 1, p. 1-15, 2021.

[7] W. Chen et al., "Deep learning for instance retrieval: a survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 6, p. 7270-7292, 2023.

[8] L. N. Rani, and Y. Yuhandri, "Similarity measurement on logo image using cbir (content base image retrieval) and cnn resnet-18 architecture," in Proceedings of the International Conference on Computer Science, Information Technology and Engineering Jakarta, Indonesia, p. 228-233, 2023.

[9] M. A. Mohammed, Z. A. Oraibi, and M. A. Hussain, "Content based image retrieval using fine-tuned deep features with transfer learning," in Proceedings of the 2nd International Conference on Computer System, Information Technology, and Electrical Engineering, Banda Aceh, Indonesia, p. 108-113, 2023.

[10] H. Xu, J. Wang, and L. Mao, "Relevance feedback for content-based image retrieval using deep learning," in Proceedings of the 2nd International Conference on Image, Vision and Computing, Chengdu, China, p. 629-633, 2017.

[11] D. Liu, J. Shen, Z. Xia, and X. Sun, "A content-based image retrieval scheme using an encrypted difference histogram in cloud computing," Information, vol. *8*, no. 96, 2017.

[12] A. Phadikar, S. P. Maity and M. K. Kundu, "Quantization based data hiding scheme for efficient quality access control of images using dwt via lifting," in Proceedings of the Sixth Indian Conference on Computer Vision, Graphics & Image Processing, Bhubaneswar, India, p. 265-272, 2008.

[13] B. Chen and G. W. Wornell, "Digital watermarking and information embedding using dither modulation," in Proceedings of the IEEE Workshop on Multimedia Signal Processing, Redondo Beach, CA, p. 273-278, 1998.

[14] S. Maji, and S. Bose, "CBIR using features derived by deep learning," ACM/IMS Transactions on Data Science, vol. 2, no. 3, p. 1–24, 2012.

[15] S. Hamreras, R. Benítez-Rochel, B. Boucheham, M.A. Molina-Cabello, E. López-Rubio, "Content based image retrieval by convolutional neural networks," Edited Ferrández Vicente, J., Álvarez-Sánchez, J., de la Paz López, F., Toledo Moreo, J., Adeli, From Bioinspired Systems and Biomedical Applications to Machine Learning. Lecture Notes in Computer Science, Springer, vol. 11487, 2019.

[16] G. Litjens, T. Kooi, B.E. Bejnordi, A. Setio, F. Ciompi, M. Ghafoorian, JAWM van der Laak, B van Ginneken, C.I. Sánchez, "A survey on deep learning in medical image analysis," Med Image Anal, vol. 42, p. 60-88, 2017.

[17] S. Sikandar, R. Mahum, and A. Alsalman, "A novel hybrid approach for a content-based image retrieval using feature fusion," Applied Sciences, vol. 13, no. 7, p. 4581, 2023.

[18] J. Deng, O. Russakovsky, J. Krause, M. Bernstein, A. Berg, L. Fei-Fei, "Scalable multi-label annotation", ACM

conference on human factors in computing (CHI), 2014, ImageNet.: http://www.image-net.org, Last accessed: November 2025.

[19] G. Griffin, A. Holub, and P. Perona, Caltech 256 (1.0) [Data set], CaltechDATA, (2022).

[20] O. Russakovsky et al. ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision, vol. 115, no. 3, p. 211-252, 2015.

[21] A. Howard, E. Park, and W. Kan, "ImageNet Object Localization Challenge", https://www.kaggle.com/datasets/ifigotin/imagenetmini-1000, 2018, Kaggle, Last accessed: November 2025.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error measurement to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 1, p. 600-612, 2004.

[23] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, "Attacks on copyright marking systems", in David Aucsmith (Ed), Information Hiding, Second International Workshop, IH'98, Portland, Oregon, U.S.A., April 15-17, 1998, Proceedings, LNCS 1525, Springer-Verlag, ISBN 3-540-65386-4, pp. 219-239, https://www.petitcolas.net/watermarking/stirmark/, Last accessed Nov. 2025.

[24] K. Ramanjaneyulu, K. Swamy, and Ch. Rao, "CBIR system using integrated dwt and cnn architecture," Journal of Physics: Conference Series, vol. 1228, 2018.

[25] S. Singh, and S. Batra, "An efficient bi-layer content-based image retrieval system," Multimedia Tools and Applications, vol. 79, no. 25-26, p. 17731–17759, 2020.

[26] K. Sikha, and K.P. Soman, "Dynamic mode decomposition based salient edge/region features for content-based image retrieval," Multimedia Tools and Applications, vol. 80, no. 10, p. 15937–15958, 2021.

[27] F. Taheri, K. Rahbar, and P. Salimi, "Effective features in content-based image retrieval from a combination of low-level features and deep boltzmann machine," Multimedia Tools and Applications, vol. 24, p. 37959-37982, 2022.

[28] Y.D. Kenchappa, and K. Kwadiki, "Content-based image retrieval using integrated features and multi-subspace randomization and collaboration," in International Journal of System Assurance Engineering and Management, vol. 13, p. 2540–2550, 2022.

[29] M.M. Kabir, A. Ishraq, K. Nur, M.F. Mridha, "Content-based image retrieval using auto embedder," Journal of Advances in Information Technology, vol. 13, no. 3, p. 240-248, 2022.

[30] S.F. Salih, and A.A. Abdulla, "An effective bi-layer content-based image retrieval technique," The Journal of Supercomputing, vol. 79, no. 2, p. 2308–2331, 2023.

[31] H. Rastegar, and D. Giveki, "Designing a new deep convolutional neural network for content-based image retrieval with relevance feedback," Computers and Electrical Engineering, vol. 106, p. 108593, 2023.