Neural Computing and Applications (2023) 35:4855–4880
https://doi.org/10.1007/s00521-021-06054-y

S.I. : MACHINE LEARNING APPLICATIONS FOR SECURITY

Secure and efficient image retrieval through invariant features selection in insecure cloud environments

Sumit Kumar[1] · Arup Kumar Pal[1] · SK Hafizul Islam[2] · Mohammad Hammoudeh[3]

Received: 8 January 2021 / Accepted: 16 April 2021 / Published online: 6 June 2021

Abstract
Advances in computer vision technologies lead to a renewed focus on content-based image retrieval (CBIR) in computer multimedia content analysis applications. CBIR is a technique for image retrieval using automatically derived features. As the size of image repositories grew, supported by increased cloud storage adoption, security concern around trust in cloud service provider (CSP) witnessed a resurgence of interest in user privacy. Hence, unlike in traditional CBIR, cloud-based image retrieval is based on the encrypted feature vector. This may reduce the overall retrieval performance of the system. Consequently, mechanisms are needed to protect the feature vector and the actual images during transmission. Second, to provide image content security, images are often encrypted by users before uploading to the cloud. This article addresses the challenges of retrieving images securely from an untrusted cloud environment. Images are represented in terms of their local invariant features to form an image feature vector. Later, an asymmetric scalar-product-preserving encryption (ASPE) is applied to secure the feature vector. Then, images are encrypted before they are uploaded to a cloud server. The proposed method has been tested on various Corel image datasets and the medical image repository. Performance evaluation shows that the proposed method outperforms its best secure CBIR systems in the literature.

Keywords Content-based image retrieval (CBIR) · Color image cryptographic system · Invariant feature selection · Asymmetric scalar-product-preserving encryption

1 Introduction

Approximately from the last two decades, the amount of generated multimedia data has increased enormously due to advancements and consumer electronics availability, e.g., smartphones. This type of data is often shared with other persons or on various social media platforms. In one year, approximately 250 Billion and 40 Billion images were shared on Facebook and Instagram, respectively [1], which reflects the growth in multimedia data usage. With data sizes at this scale, it is challenging to retrieve specific images from a massive-scale repository. Recently, cloud service offered users a convenient and cost-effective way to store and share images. This involves transferring data to an unknown third-party as a cloud server to store, handle, and retrieve in the future.

Content-based image retrieval (CBIR) services typically incur high storage and computation complexities. Simultaneously, the cloud has the capabilities that can match the demand of high computation and gigantic storage for the

✉ SK Hafizul Islam
hafi786@iiitkalyani.ac.in

Sumit Kumar
sumitkumar@cse.ism.ac.in

Arup Kumar Pal
arupkrpal@iitism.ac.in

Mohammad Hammoudeh
M.Hammoudeh@mmu.ac.uk

[1] Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, Jharkhand 826004, India

[2] Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India

[3] Department of Computing and Mathematics, Manchester Metropolitan University, Manchester M1 5GD, UK

large-scale CBIR system, hence making the outsourcing of data practicable. By outsourcing CBIR functionalities to the cloud, the data owner may divert his/her duties of maintaining a local image database and satisfying user/application requirements. Despite its many benefits, cloud platforms pose security concerns related to image storage, transmission, and retrieval [2]. Images often contain personal- or business-sensitive information which must be handled with care. A reliable cloud service is expected to offer users secure means to transfer, share, and access data. For instance, a patient may want to share his/her medical images with their doctor. Additionally, a cloud service provider (CSP) may act maliciously, or a security infringement may compromise his systems. Hence, there is a risk that user's data may be leaked or accessed by an unauthorized entity. A common practice to protect against data breaches is encrypting users' images before uploading them to a cloud server. These stored images are usually retrieved using queries. In the healthcare scenario, a hospital outlets a large number of medical images, often on a remote cloud server(s). A doctor can review past cases of similar medical conditions for better analysis and treatment of a patient. The doctor may generate a query to retrieve similar cases based on medical features. If the CSP CBIR's service is not effective enough, it can return wrong images leading to inaccurate diagnosis.

Hence, to provide security, users encrypt their images before transferring them to the cloud. Image retrieval is carried out based on the encrypted features of these images. Initially, images will be retrieved using the annotation attached to them depending on the visual perception, which may vary from person to person for an image. However, with a large number of images, it is impractical to manually annotate each image. CBIR [3, 4] presents a feasible solution to retrieve similar images from a large set of images. In the CBIR system, the image features play a vital role in the retrieval process. There must be a close association between the primitive visual image features and the image's actual content. These primitive image features include color [5], texture [6], and shape [7]. During the initial phase of CBIR, researchers tend to incorporate only one feature. Based on only one feature, the results are derived. But now, various combinations of these primitive visual features are fused to form the final feature vector, and the same is used to attain better retrieval performance. In CBIR, image features are extracted in two ways. First one, i.e., the global image features, the selected image feature(s) is extracted from the entire image. But due to the increased enhancement in image quality and with such large variation within a small portion of an image, global features do not efficiently address the locality and spatiality of an image. This results in the degradation of the overall system performance. Hence, feature extraction starts with emphasizing the local visual features from non-overlapping blocks before merging them together. It is known that the local image features correlate more with the actual content. A minor change in an image orientation or size creates a big difference in the retrieved results. To overcome this issue, various invariant features [8] are incorporated. Another critical aspect of a conventional CBIR system is the similarity matching distance, e.g., Euclidean, Manhattan, Jaccard, Minkowski, etc. Based on the selected similarity measurement and distances, the top indexed images are the final retrieval results. When the feature vectors are also encrypted, conventional CBIR similarity measurements cannot be applied. In a conventional CBIR system, all the operations take place at one end. However, insecure image retrieval scenarios, CBIR operates at three different ends, as depicted in Fig. 1. Image features are extracted from the original images at the owner's part, then these images and feature vectors are encrypted and deployed at the cloud server. An authorized user issues an image query based on the image feature vector. This query image feature vector is again encrypted and stored in the cloud. Now, the cloud server is responsible for generating similar image indices and provide results.

This article focuses on secure image retrieval from cloud servers. The aim is to achieve an acceptable overall image retrieval system performance while maintaining data security and privacy. To achieve this, both the transferred images as well as their respective retrieved feature vectors are encrypted before transfer to the cloud server. This is essential as feature vectors may reveal some information to a malicious entity or eavesdropper. As feature vectors are also encrypted, a technique based on an asymmetric scalar-product-preserving encryption (ASPE)-based similarity measurement is implemented to retrieve image indexes.

In the literature, existing methods of secure image transfer assume that either the cloud server is fully honest-but-curious, or the registered user is fully trusted. However, these assumptions are not always met in real-world environments. Additionally, many systems have incorporated multiplicative group-based schemes to encrypt their data which incur a high computational cost. Therefore, in this article, we propose a scheme that provides a mechanism for securely transmitting images from an untrusted cloud to a designated user. A key management center is incorporated as a fully trusted-third party to keep track of key data exchange and to prevent forgery from the user side. The main contributions of the proposed CBIR system are as follows:

- The image retrieval performance is improved based on the color, shape, and texture image features.
- It is capable to handle rotational features by considering invariant features.

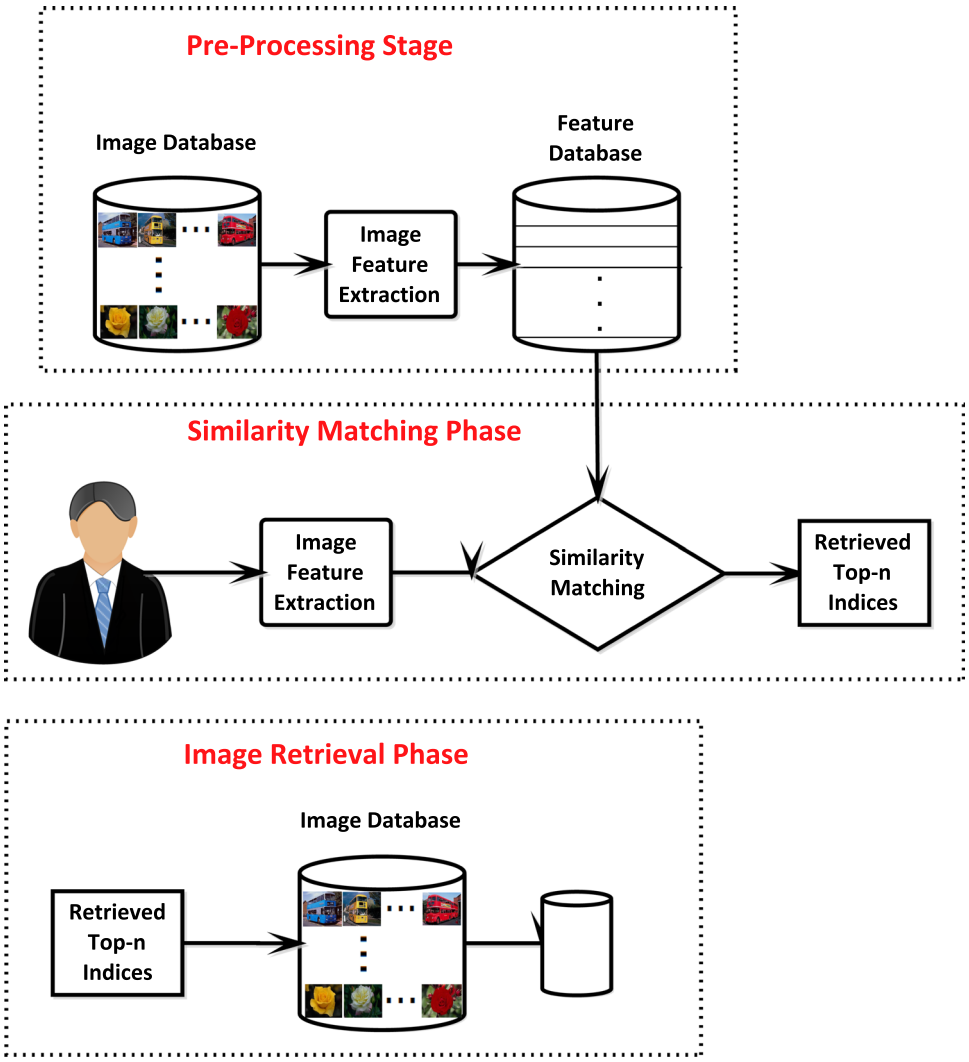


Fig. 1 Component diagram of a basic CBIR system

- A database owner can store images and their associated features using the proposed image encryption scheme and an ASPE respective.
- It is based on four entities, specifically, the database owner, CSP, authorized user, and a fully trusted KMC that establishes a trusted environment between the aforementioned entities.

The rest of the paper is organized as follows. In Sect. 2, similar works in the literature are critically reviewed. In Sect. 3, the overall system structure is discussed. In Sect. 4, we briefly present the various techniques used during the course of this article. In Sect. 5, the details of the feature vector construction are presented. In Sect. 6, various security requirements in a CBIR system are presented. In Sect. 7, we have represented the objective of this secure proposed CBIR system. In Sect. 8, various security measures supported by the proposed CBIR system are presented. Section 9 discusses the adopted updating mechanism. Section 10 presents the performance evaluation results. Section 11 concludes the article and discusses future work avenues.

2 Related work

The past decade witnessed unprecedented growth in the amount of image data generation [9, 10]. To meet the demand for image data processing, cloud service emerged as a suitable technology for effective image storage, sharing, and retrieval. Often, data hosted on the cloud in an encrypted form, which requires an efficient searchable encryption system to transfer images from a legitimate person in a secure way [11]. A searchable encryption system allows the user to get similar images from an encrypted form. Originally, this method was applied on text documents to find a document. However, with the increased availability of multimedia data like images, searchable encryption has been implemented for image

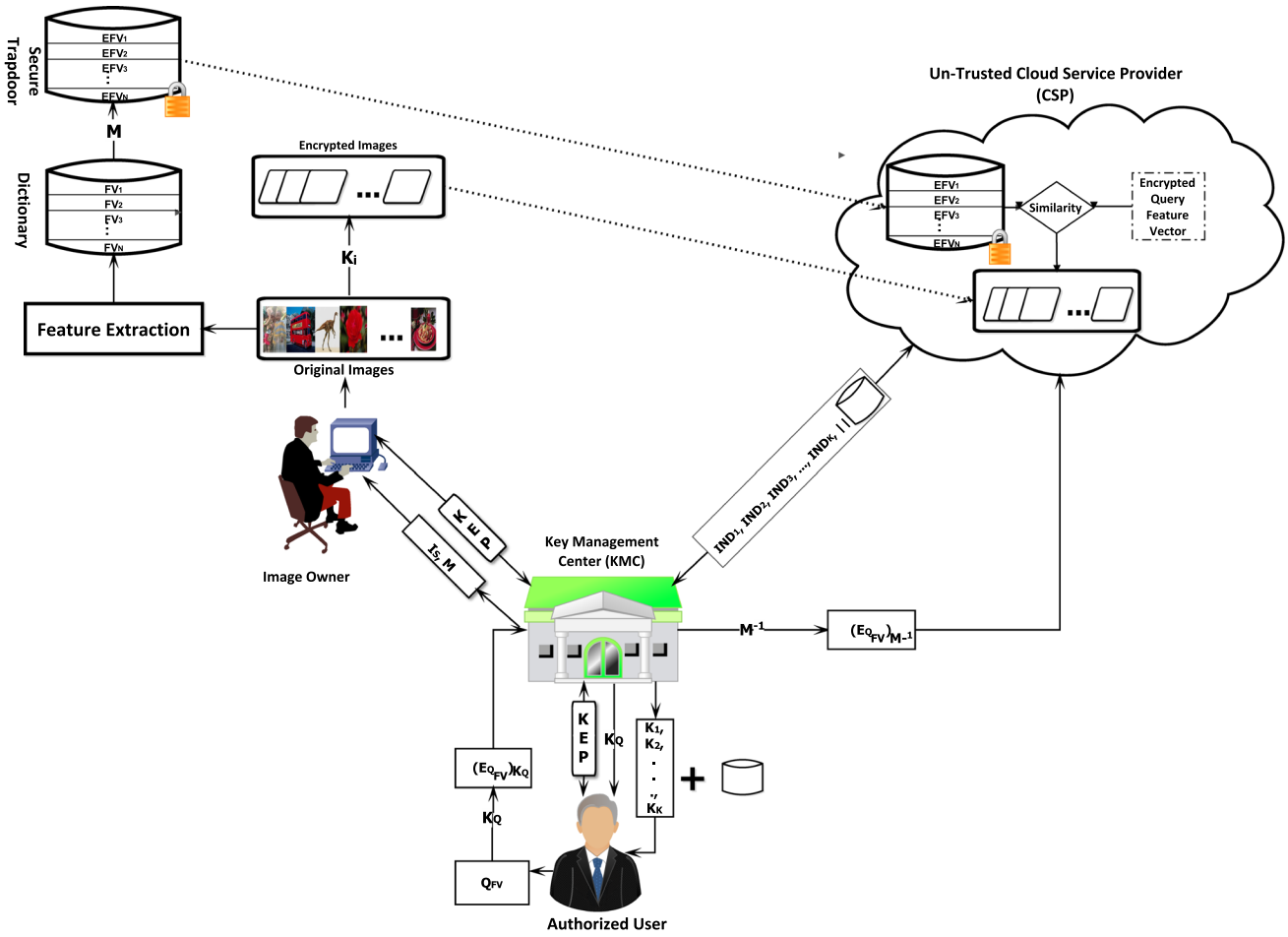
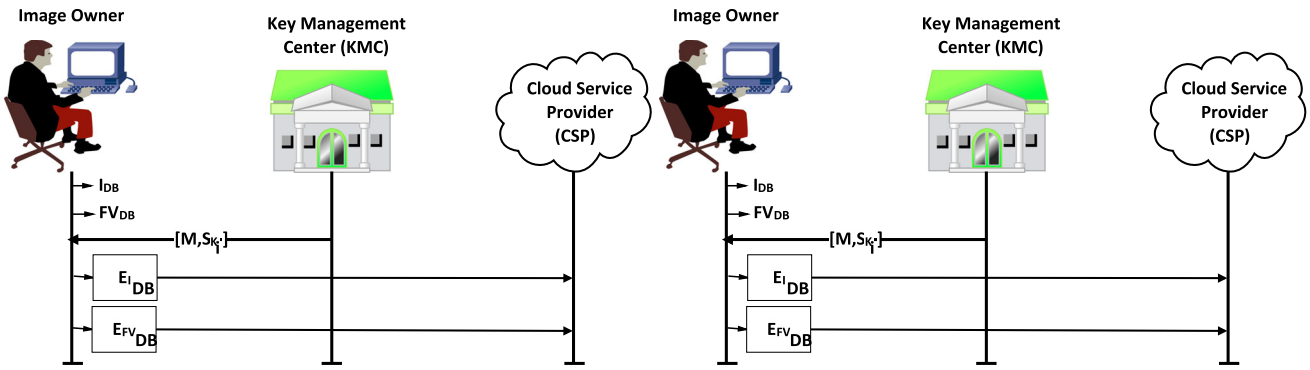


Fig. 2 Overall structure of the proposed system

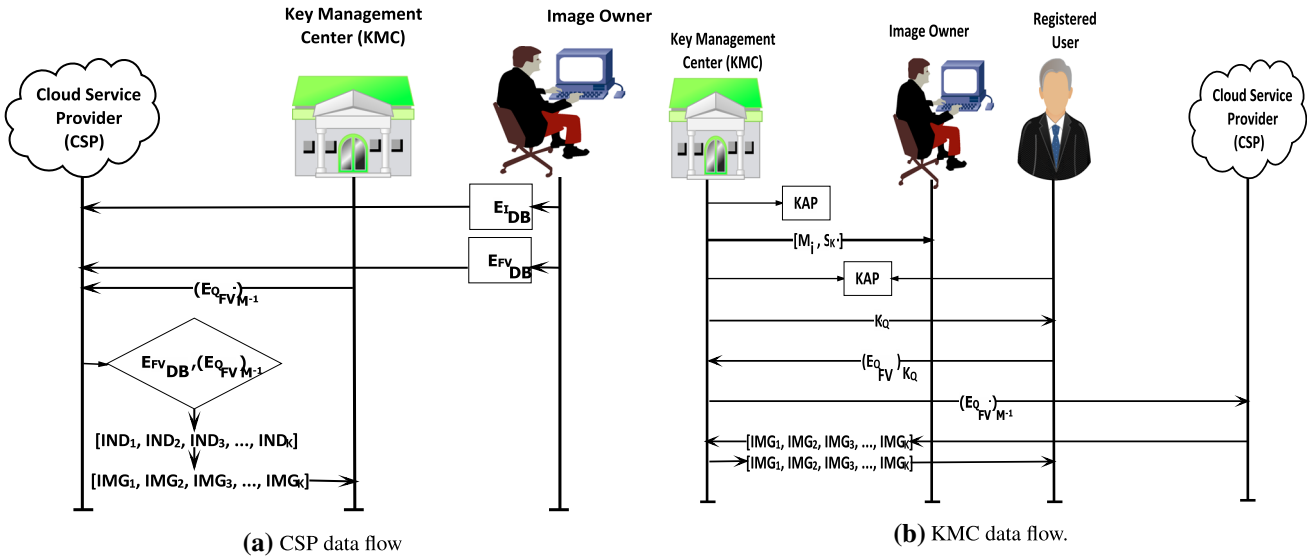
retrieval as well. Song et al. [12] had proposed one of the proposals to provide high retrieval time from a large database. However, their study did not cover statistical attacks. Then, Chang et al. [13] enhanced the [12] efficiency with encrypted hash tables to create image indexes. In this work, image pixels are encrypted using a suitable stream cipher and later from these retrieved encrypted images Markov features are directly extracted. Curtmola et al. [14] designed another searchable encryption scheme that incorporates some security features and achieves the optimal search time. All of these mentioned works are very early works which are related to searchable encryption. These works only support Boolean searching majorly for documents which result in whether a particular encrypted document is present or not in a database. In [15], cloud-based image retrieval method is proposed. This method may reveal all the content to the cloud service provider. The work [16] constructs the probably first scheme that performs image retrieval on encrypted images. Initially, image features are extracted to form the visual word and later to rank the similar images Jaccard similarity has been opted between the query image feature vector and the image feature database. In order to preserve the visual content of the image min-hash algorithm and order-preserving encryption has been imposed. In the work [17] proposed another scheme namely SEISA. The scheme has been used secure k-means along with access control to update an image dataset on a cloud server. In [18], respective feature vectors are constructed to represent the concerned image. Later, a hash table in combination with local sensitive hashing is incorporated to increase the efficiency of the system. In this scheme, a kNN-based algorithm is used to protect the feature vector. Similarly, the work proposed in [19] supports multi-owner such that individual owners have their keys so that no one can access other's content. This work registered user is assumed to be always trusted, but it is always not feasible to achieve. Xia et al. [20] presented a secure CBIR system where local visual image features are extracted to form the image feature vector and earth mover's distance has been used as similarity measurement. In [21], image encryption is applied based on the encrypted indexes. This method



(a) Data flow from the data owner end. **(b)** Data flow from the data registered user end.

Fig. 3 Flowcharts illustrating data flow from both data owner and authorized users ends

enhances the efficiency of the system by introducing automatic database upgrading in the cloud. Several works have been proposed which uses homomorphic encryption-based searchable encryption. This kind of scheme has a multiplicative group-based encryption process and similarity finding. As in the work [22], Bellafqira et al. proposed a homomorphic encryption-based system that incurs high computation and does not achieve accurate retrieval results. The work proposed by Weng et al. [23] provides two layers of security. At the first level, query image content and respective feature are made secured by incorporating robust hash values. This scheme gives access to the registered client to exclude some bits from the hash values. This step enhances the uncertainty at the server's end. Now, it will be really hard for the cloud server to know about the client's information. Thereafter, the client search to get the best available result. In this work, cloud server and client's privacy is preserved to either of them as only hash values are reciprocated. In the work proposed by Zhou et al. [24], authors have presented a secure scheme for e-health care in cloud-assisted environment which can efficiently preserve the privacy of the medical data and its corresponding extracted visual words. Here, privacy has been preserved based on the homomorphic encryption. Later, this scheme incorporated disease modeling outsourcing and also it's early-stage intervention. These are taken care by an efficient privacy-preserving function correlation matching retrieved from dynamic medical text mining and a secure image feature extraction mechanism for the medical images. The work presented by



(a) CSP data flow **(b)** KMC data flow.

Fig. 4 Flowcharts illustrating data flow from both the CSP and the KMC ends

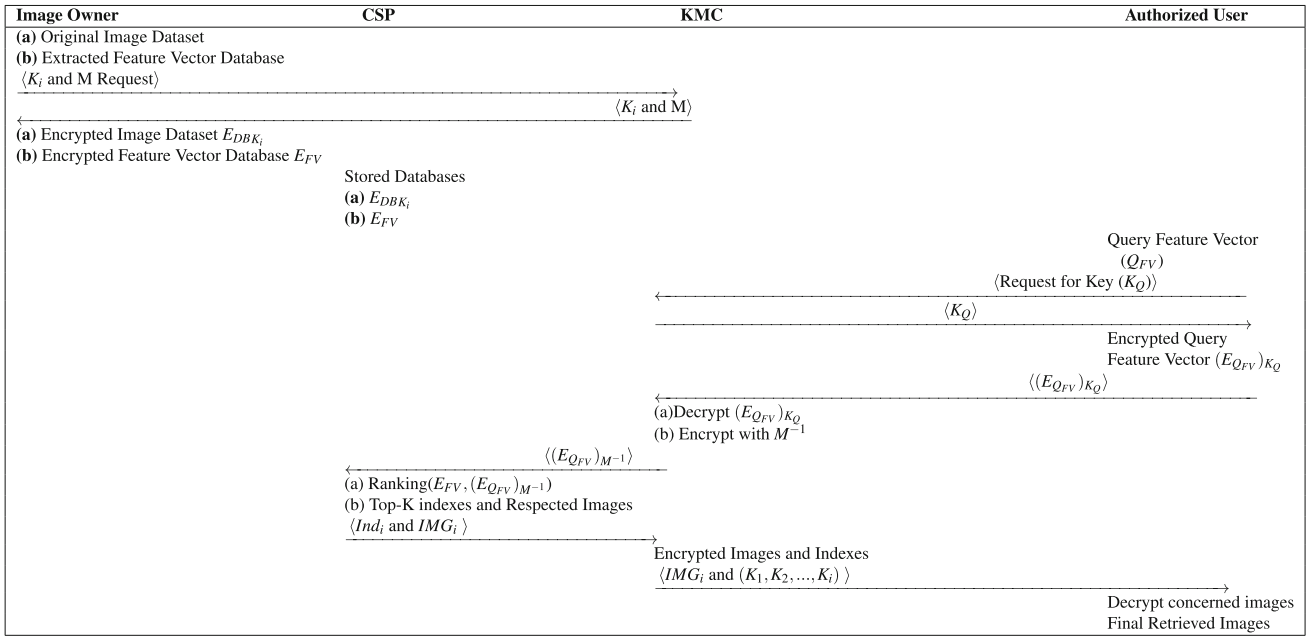


Fig. 5 Complete overview of communication between image owner, CSP, KMC, and authorized user in a cloud-assisted environment

Shengshan et al. [25] proposed an efficient privacy preserving scheme which is based on extraction of secure scale-invariant feature transform (SIFT) over a large volume of images. This work starts by pointing out the issues with previous similar works which lacks in security, practicality, or overall efficient system designing. Neither of the previously proposed similar works can optimize the use of secure SIFT image features in terms of robustness and distinctiveness. Further, they design a new efficient and secure scheme to meet the practicality. They started the scheme with splitting the input image. Second, incorporated two schemes for multiplications and secure comparison and lastly, algorithmically spread image feature extraction to two unknown cloud servers. In the literature, there has been very few works that consider the case where in a trusted user may also turn unfaithful during the course of time for some favor. To deal with this scenario, Zhihua et al. [26] presented a solution where the feature vector is constructed using EDH, color layout descriptor, color structure descriptor, and scalable color histogram. Then kNN-based security is imposed. This scheme is complex and incurs high computational cost.

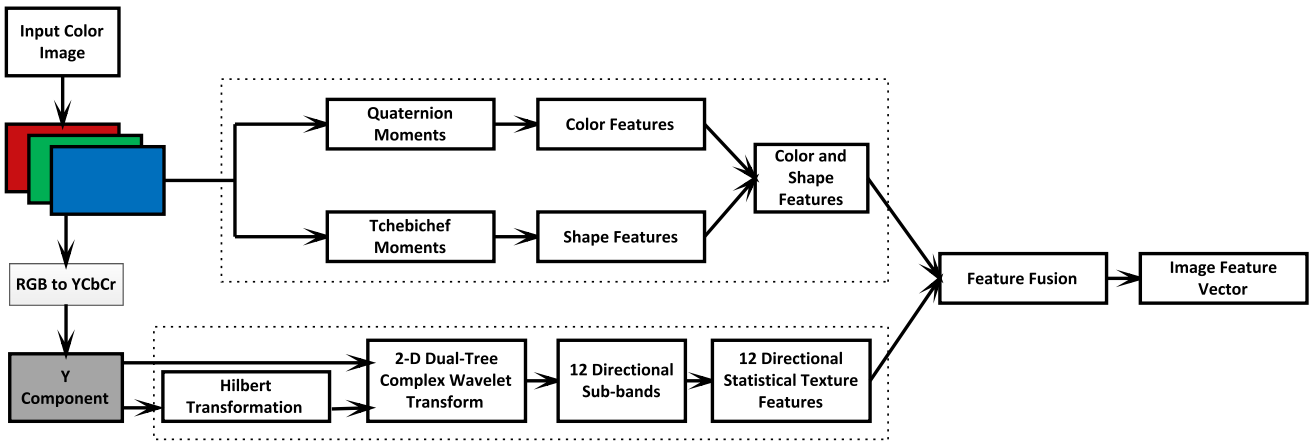


Fig. 6 Feature vector generation process

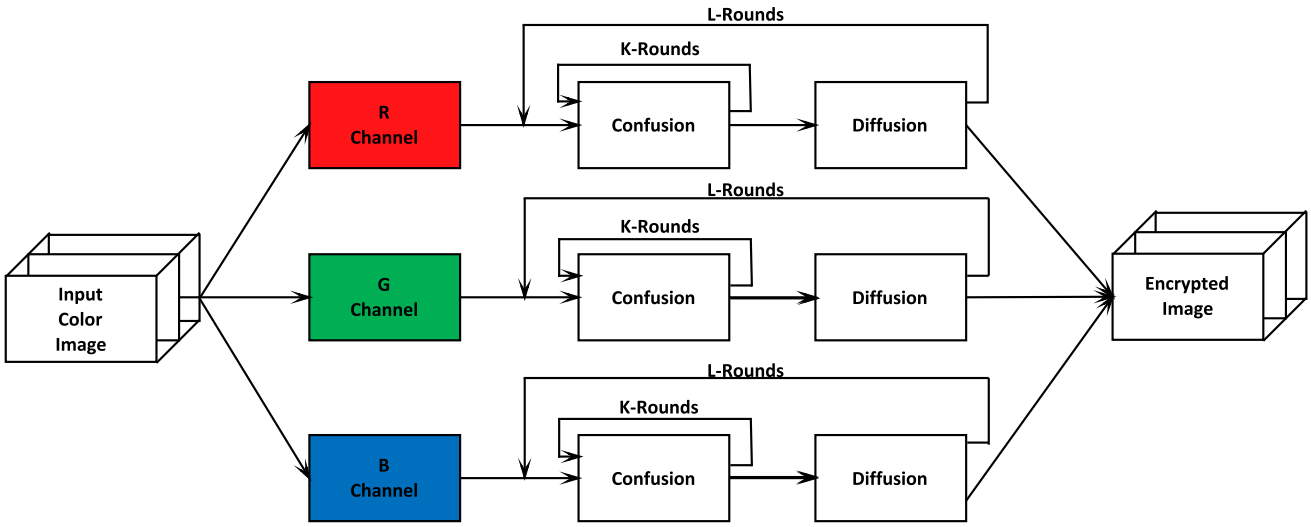


Fig. 7 Conventional confusion–diffusion structure of a color image encryption scheme

Based on this literature review, there is a need to design a system which can deal with unfaithful registered user efficiently. To address this gap, we present a solution where local invariant features are extracted to form the feature vector. Then, ASPE-based security is applied. A fully trusted third party based on KMC is introduced for generating image encryption keys, feature vector encryption keys, storing them, and transferring results. In this work, a registered user can access only the legitimately allowed number of images from the cloud in a secure environment. The overall system structure is presented in the following section.

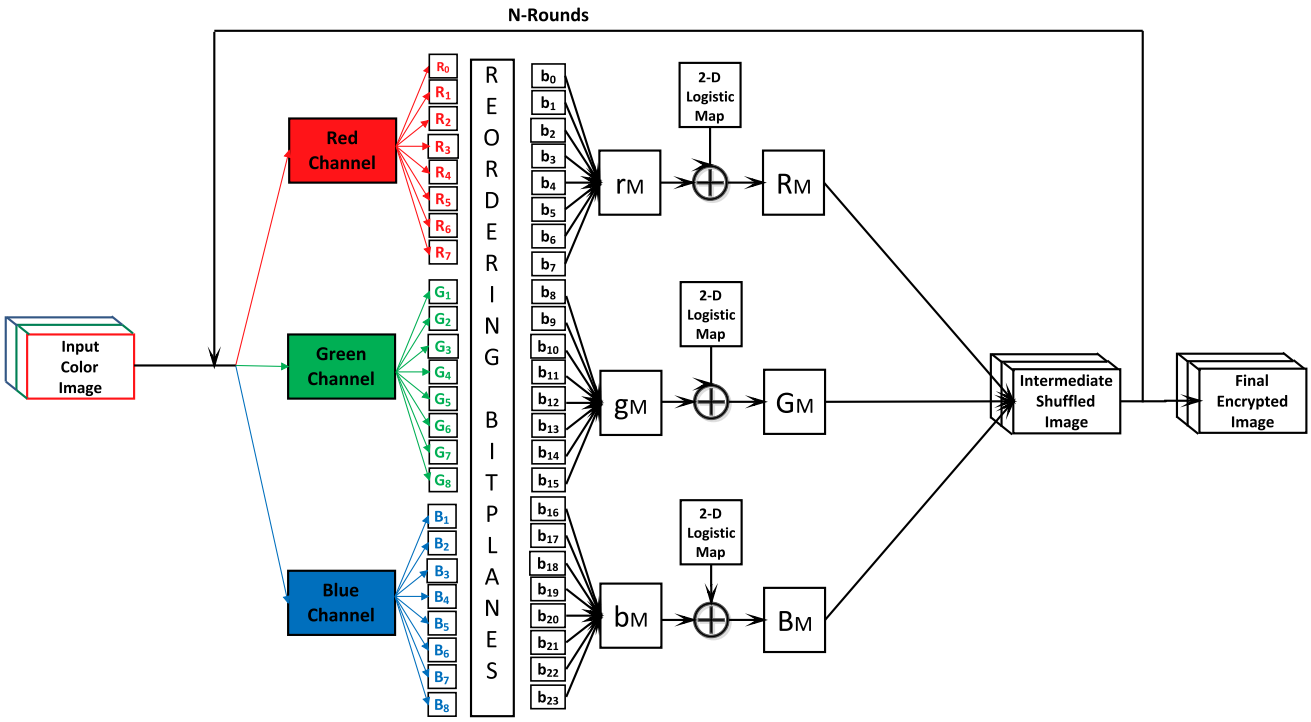


Fig. 8 Our proposed image encryption scheme

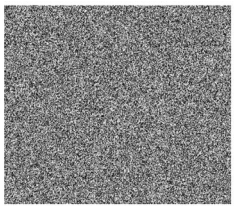
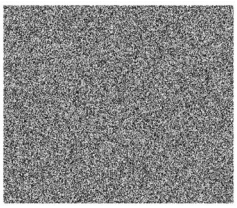
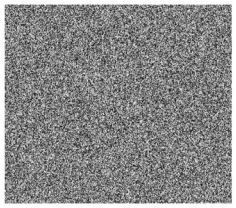
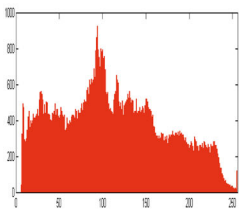
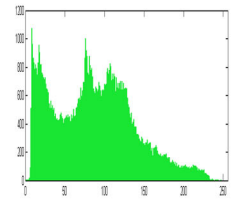
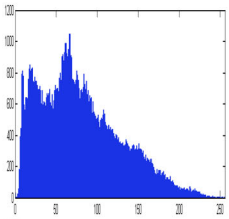
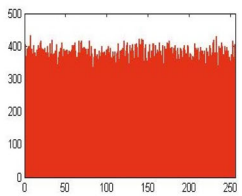
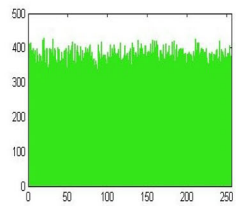
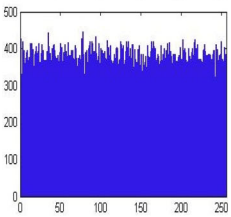
Input RGB Image				
Decomposed Components				
Cipher				
Entropy	**Original**	**7.8133**	**7.4219**	**7.5184**
	Cipher	**7.9981**	**7.9983**	**7.9984**
Histogram	**Original**			
	Cipher			
PSNR		**7.84**	**7.19**	**7.40**
Final Encrypted Image				

Fig. 9 Analysis of the proposed image encryption scheme

3 An overview of the system structure

Our proposed system comprises four entities, namely image owner, untrusted cloud server, the authorized user, and KMC. The overall structure of the proposed system is pictorially depicted in Fig. 2.

An image owner could be an individual or organization which wants to use a cloud service and has image database(s) DB_I. The main role of the image owner is to extract the visual image features based on color, texture, and shape and combine them together to form the feature vector of an image i such that $i \in DB_I$ and FV_i. Afterward, all the feature vectors are combined to form the feature vector database DB_{FV}. Next, the image owner setup a key exchange protocol (KEP) [27] between the image owner and the KMC to attain M and K_i. Then, the next duty of the image owner is to encrypt the entire image database using K_i, received from KMC, to get E_{DB_I} and encrypt FV_{DB} using M to attain $E_{DB_{FV}}$. Finally, the image owner communicates E_{DB_I} and $E_{DB_{FV}}$ to the specific cloud server for storage and future retrieval of similar images.

An authorized data user is a legitimate entity that queries an image and wants to retrieve similar images. One can illustrate an authorized user as a doctor working in a hospital who is permitted to access patient data stored on the cloud server. The user first runs the similar image feature extraction process from the query image to form the query feature vector Q_{FV}. Then, to retrieve similar images from the cloud, the user uses a KEP where the KMC provides a key K_{KMC}. The user encrypts the Q_{FV} using the cryptographic key K_Q received from KMC using suitable KEP and sends it to the KMC. The KMC retrieves similar results from the cloud and sends them back to the user with keys related to those image's index only. The roles of the image owner and the authorized users are illustrated using the flowcharts in Fig. 3.

The cloud server offers massive storage capacity. The functionalities of cloud are to accumulate E_{DB_I} and $E_{DB_{FV}}$. When the cloud server receives an encrypted query feature vector $E_{Q_{FV}}$ from the KMC, the cloud server performs similarity matching between $E_{Q_{FV}}$ and $E_{DB_{FV}}$ and generates the required indexes. In this work, just for the retrieval part, the cloud server is assumed to be honest-but-curious, i.e., it will perform this task fairly and produced the results correctly. Finally, the cloud server sends back those encrypted images along with indices to the requesting KMC. One another duty of the cloud server is to update the image database or the feature vector database when directed by the image owner.

The KMC is assumed to be a fully trusted third party. The duties of KMC include generating different keys, storing keys, collaboration with the cloud server to retrieve images and provide desired results to the user. When the KEP is set up between KMC and the image owner, the KMC provides a feature vector encryption key, i.e., invertible key M and image encryption keys K_i to the image owner to encrypt the feature database and each image. Another KEP runs between the KMC and the user to provide a random key to the user to encrypt the generated Q_{FV} and have $(E_{Q_{FV}})_{K_Q}$ before it sends them to the KMC. KMC decrypts Q_{FV} and again encrypt it with M^{-1}. Then, the KMC transfers this query feature vector to the cloud to find similar encrypted images. When the KMC receives the encrypted images from the cloud, it transfers them to the concerned user with the set of keys on only those images. Various operations of KMC and CSP are illustrated in Fig. 4. In the following, we summarize various steps in the proposed scheme:

1. The image owner extracts various visual image features and combines them to form the final feature vector. Then, the KEP is set up between the KMC and the image owner so that the image owner receives the feature vector key and image encryption keys. The image owner provides the KMC with the list of the registered users. A user encrypts the image database and feature database before she/he transfers them to the cloud server and find similar images later. When a new user is registered, the image owner sends an updated list so that the KMC can overwrite it with the older list.
2. The registered user extracts the same features from the query image and forms the query feature vector. The user runs a KEP with the KMC to get the one-time key to encrypt the query feature vector. After encrypting the query feature vector, it sends it to the KMC.
3. After receiving the encrypted query feature vector from the user, the KMC decrypts it and re-encrypt it using M^{-1} and sends the result to the cloud server.
4. When the cloud server receives the request from the KMC, it runs the designated similarity matching algorithm and finds the indexes and encrypted images. Then, the cloud server transmits these retrieved results to the KMC.
5. The KMC sends those retrieved images and their respective keys to the user where the user decrypts them to obtain the final retrieved results.

Various communication steps take place during the course of the proposed CBIR system, starting from owner feature extraction to the registered user getting the desired images. All four system entities are depicted in Fig. 5.

4 Preliminaries

In this section, we briefly explain various techniques that are used by our proposed CBIR system.

4.1 Quaternion moments

According to Hamilton [28], a quaternion, Q, can be generalized as a complex number such that $Q = r + s\hat{x} + t\hat{y} + u\hat{z}$, where r is the real part and others are the complex one. Here $r, s, t, u \in R$ and $\hat{x}$, $\hat{y}$, $\hat{z}$ are the complex units such that

$$\hat{x}^2 = \hat{y}^2 = \hat{z}^2 = \hat{x} \times \hat{y} \times \hat{z} = -1$$
$$\hat{x} \times \hat{y} = -\hat{y} \times \hat{x} = \hat{z},$$
$$\hat{y} \times \hat{z} = -\hat{z} \times \hat{y} = \hat{x},$$
$$\hat{z} \times \hat{x} = -\hat{x} \times \hat{z} = \hat{y}.$$

Here, a quaternion is called pure when its real part is zero $(r = 0)$. The conjugate and modulus of Q are defined as:

$$Q^c = r - s\hat{x} - t\hat{y} - u\hat{z}$$
$$|Q| = \sqrt{r^2 + s^2 + t^2 + u^2}$$

To the best of our knowledge, Ell and Sangwine [29] have first described an RGB image as a quaternion. Let $IM(x, y)$ be an RGB color image, it can be represented as a three of its primitive channel as a pure quaternion as $IM(x, y) = IM_R(x, y)\hat{x} + IM_G(x, y)\hat{y} + IM_B(x, y)\hat{Z}$ where $I_R(x, y), I_G(x, y)$, and $I_G(x, y)$ are the red, green, and blue color channels of a pixel.

In this paper, we use six different orthogonal moments [30], namely Quaternion Zernike Moments, Quaternion Pseudo Zernike Moments, Quaternion Legendre-Fourier Moments, Quaternion Exponent Moments, Quaternion Radial Harmonic Fourier Transform Moments, and Quaternion Radial Substituted Chebyshev Moments.

4.2 Tchebichef moments

The scaled Tchebichef polynomials are defined as

$$\bar{T}_n(x) = \frac{T_n(x)}{\beta(n, N)} \tag{1}$$

where $Tn(x)$ is the discrete Tchebichef polynomial of degree n, and (n, N) is a suitable constant which is independent of x. Under the above transformation, the squared-norm of the scaled polynomials gets modified according to the following formula

$$\bar{\rho}(n, N) = \frac{\rho(n, N)}{\beta(n, N)^2} \tag{2}$$

We now define the Tchebichef moments [31] as

$$T_{pq} = \frac{1}{\bar{\rho}(p, N)\bar{\rho}(q, N)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \bar{t}_p(x)\bar{t}_q(x) f(x, y) \tag{3}$$

where $x, y = 0, 1, 2, 3, \ldots, N-1$

5 Feature vector construction

A reliable CBIR system extracted features must reflect the actual content of an input image. Since the local image features are more associated with the image's actual content, hence, in our proposed CBIR system, we incorporate local color, shape, and texture visual image features to enhance the accuracy of the returned results. In this paper, we have also taken care of the fact that if the input image is scaled to smaller or larger dimension the content of the image does not change. But this scaling can lead to different output images for various scaled images. Hence, in this work, we have not only fused local image features but those local image features which are invariant in different ways.

In this work, the feature extraction process has been divided into two stages. The first stage evaluates the color and shape features providing color feature vector and shape feature vector. The second stage comprises of texture visual image feature that results in a texture feature vector. All three feature vectors are combined to form the final image feature vector. For the color evaluation, we adopt Quaternion moments. This provides not only local, but also rotation, scale, and translation invariant color features. Then, in the same stage, we deploy Tchebichef moments for the shape features. This process provides scale and translation invariant shape features. The result of this stage is color and shape features. These processes are described in Algorithm 1.

Algorithm 1 Color and shape feature extraction.

Input: RGB Image I_q.
Output: Color & shpae feature vectors F_{CS}.
Parameter: L & $B \geq 4$, where L and B are the length and breadth of I_q
Select a query input I_q of size $M \times N \times 3$.
Decompose I_q into its color components I_R^q, I_G^q, and I_B^q
Initialize an empty color feature vector F_C.
for $\forall I_K^q \in I_R^q, I_G^q, I_B^q$ **do**
  Compute F_{C1}= Quaternion Zernike Moments(I_K^q.)
  Compute F_{C2}= Quaternion Pseudo-Zernike Moments(I_K^q.)
  Compute F_{C3}= Quaternion Legendre-Fourrier Moments(I_K^q.)
  Compute F_{C4}= Quaternion Exponent Moments(I_K^q.)
  Compute F_{C5}= Quaternion Polar Harmonic Transforms Moments(I_K^q.)
  Compute F_{C6}= Quaternion Radial Substituted Chebyshev Moments(I_K^q.)
end
Append F_{C1} to F_{C6} in F_C.
Initialize an empty Shape feature vector F_S.
for $\forall I_K^q \in I_R^q, I_G^q, I_B^q$ **do**
  Compute F_{S1}= Translation & scale invariant Tchebichef moments (I_K^q).
  Append F_{S1} in F_S.
end
Compute F_{CS}= $[F_C, F_S]$.
Return F_{CS}.

In the second stage, texture information is evaluated converting the input RGB image to its YCbCr counterpart. After that, the Y-component has been decomposed. Then, texture information is extracted into two parts. In the first part, 2-D DT-CWT [32] is applied directly on the Y-component, and some statistical parameters are evaluated. In the second part, the first Hilbert transformation [33] of the decomposed Y-component is taken to avail more directional textural information. Then, a 2-D DT-CWT is applied to this transformed Y-component followed by evaluating some statistical parameters. Finally, to form the texture-based feature vector, both parts of texture information are merged. The texture feature extraction process is summarized in Algorithm 2.

Algorithm 2 Texture feature extraction process.

Input: RGB Image I_q.
Output: Texture feature vectors F_T.
Parameter: L & $B \geq 4$, where L & B are the length and breadth of I_q
Select a query input I_q of size $M \times N \times 3$.
Decompose I_q into its color components $I_R^q j$, I_G^q, and I_B^q
Compute the Luminance Component Y_q of I_q as
$Y_q = 0.299 \times I_R^q + 0.587 \times I_G^q + 0.14 \times I_B^q$
Compute Y_{qh}= Hilbert Transformation (Y_q) Initialize an empty feature vector F_T.

for $\forall Y_i \in Y_q, Y_{qh}$ **do**
  for $D = 1$ *to* 3 **do**
    Perform 2D DT-CWT for Y_i.
    Extract 6 directional Sub-bands.
    Compute Statistical Features from each directional sub-band.
    Store all the features in F_{Temp}.
    Assign Low-pass sub-band components to Y_i.
    Append F_{Temp}in F_i.
  end
  Append F_i in F_T
end
Return F_T.

To complete the feature vector process color, shape, and texture feature vectors are fused. The pictorial representation of the feature extraction process is shown in Fig. 6.

Algorithm 3 Feature Extraction.

Input: RGB Image I_q.
Output: Final feature vector fv_I.
Parameter: Size of Q_I is $M \times N \times 3$.
Select a query input I_q of size $M \times N \times 3$.
Decompose I_q into its color components I_R^q, I_G^q, and I_B^q
for $\forall I_K^q \in I_R^q, I_G^q, I_B^q$ **do**
  ${Q_K}^q$=Quaternion Moments(I_K^q.)
  Append ${Q_K}^q$ in the color feature vector FV_C. ${T_K}^q$=Tchebichef Moments(I_K^q.)
  Append ${T_K}^q$ in the shape feature vector FV_S.
end
Convert RGB I_q into YCbCr Color plane.
Extract luminance component Y_q of I_q.
Compute $Y_q^{'}$=Hilber Transformation(Y_q).
Initialize empty set ${FV_T}^q$ and ${FV_T}^{q'}$.
for *i=1 to 3* **do**
  Compute 2D DT-CWT of Y_q.
  Append statistical sub-bands texture feature with ${FV_T}^q$.
  Assign Y_q= Approximation coefficient of 2D DT-CWT(Y_q). Compute 2D DT-CWT of $Y_q^{'}$.
  Append statistical sub-bands texture feature with ${FV_T}^{q'}$.
  Assign $Y_q^{'}$= Approximation coefficient of 2D DT-CWT($Y_q^{'}$).
end
Combine ${FV_T}^q$ with ${FV_T}^{q'}$ to create FV_T.
Append FV_C, FV_S, and FV_T to create final feature vector FV_q.

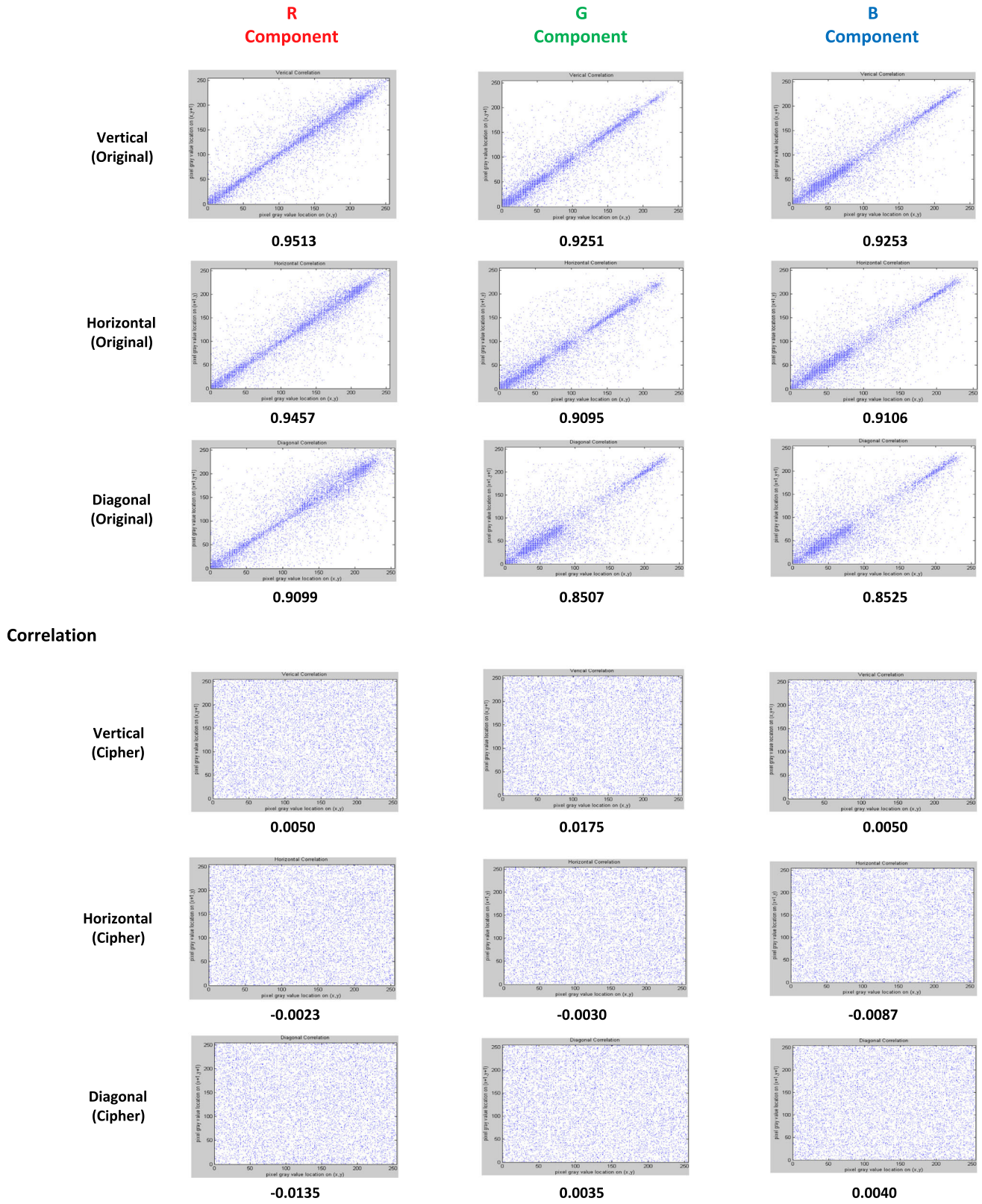


Fig. 10 Correlation analysis of the proposed image encryption scheme

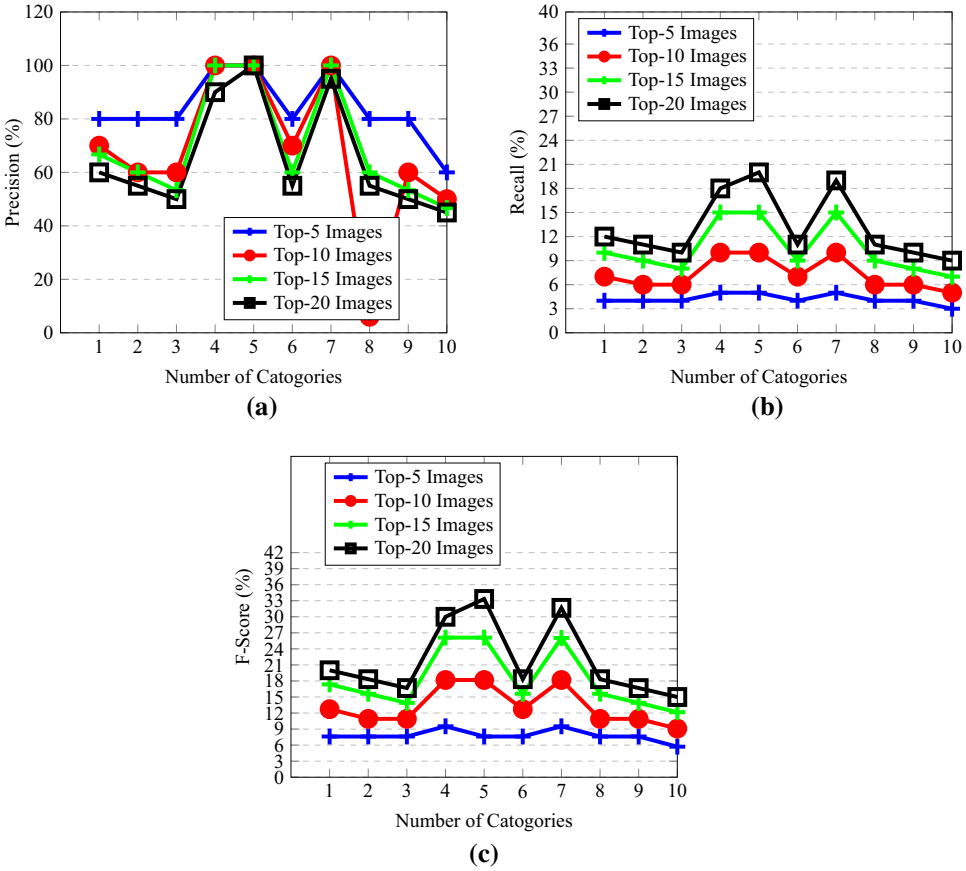


Fig. 11 **a** Precision, **b** recall, **c** F-score for different numbers of retrieved images for WANG dataset using encrypted similarity

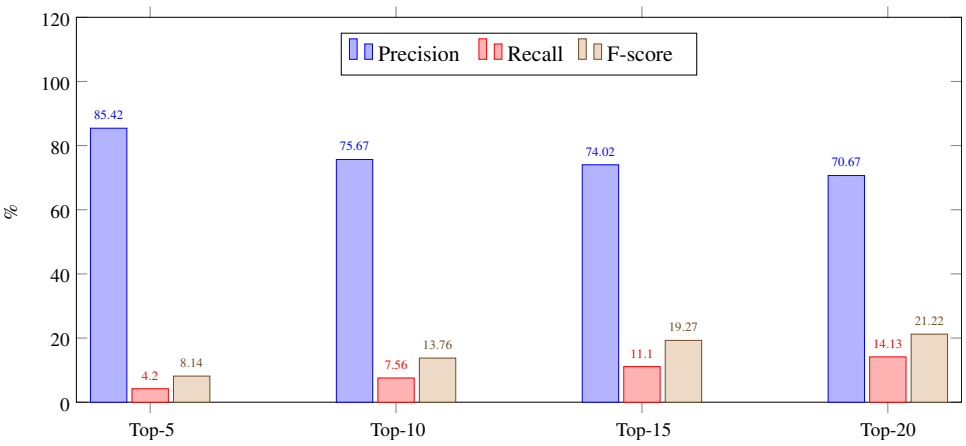


Fig. 12 Mean average precision, recall, and F-score for different numbers of outcome images of WANG database using Euclidean similarity measurement

5.1 Secure content-based image retrieval

Before retrieving the results, an authorized user first set up the key exchange between itself and its KMC to get a user query key U_{Key}. The registered user extracts the shared image features from the query image to form the query feature vector Q_{FV}. Then, the user encrypts the Q_{FV} using K_Q to get $(E_{Q_{\text{FV}}})_{K_Q}$ before sending it to the KMC. Since the key exchange is secret, no other entity can access the actual feature vector. On receiving the $(E_{Q_{\text{FV}}})_{K_Q}$, the KMC decrypts it. Then, the KMC encrypts it using M^{-1} of the corresponding owner to obtain $(E_{Q_{\text{FV}}})_{M^{-1}}$. The KMC transfers $(E_{Q_{\text{FV}}})_{M^{-1}}$ to the CSP. Because the feature vector is encrypted, the traditional similarity measures cannot be imposed here.

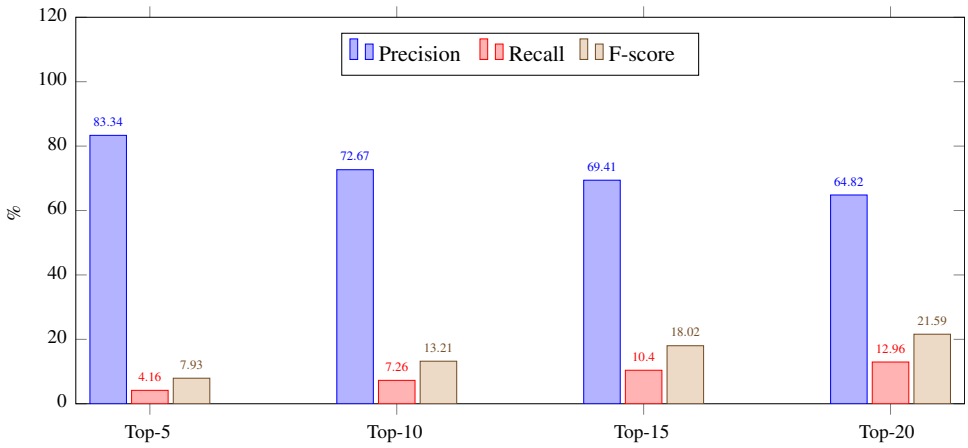


Fig. 13 Mean average precision, recall, and F-score for different numbers of outcome images of WANG database using encrypted similarity measurement

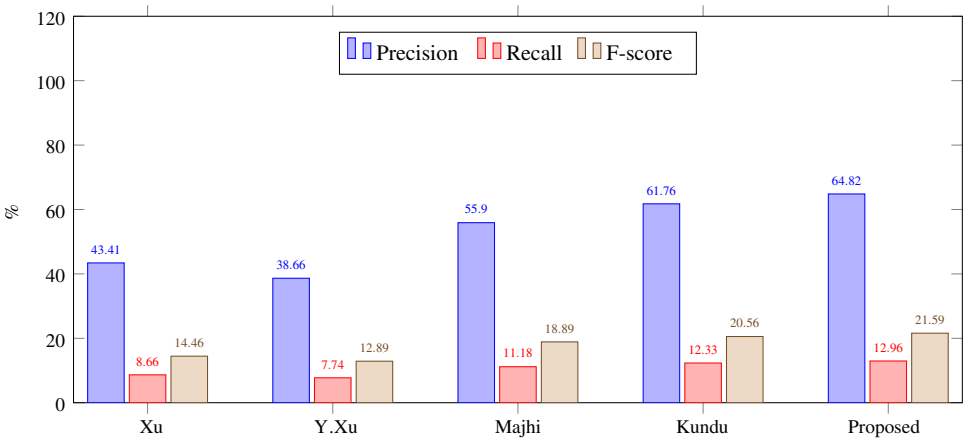


Fig. 14 Comparison results using the WANG dataset

In this work, we use the kNN similarity measurement technique to find similar indices at the CSP end. After retrieving the indices, the CSP transfers these indices along with the corresponding images to the KMP. The KMP forwards those images with only the keys required to decrypt them. After receiving the decrypted images and the associated keys, the authorized user decrypts the images and gets the final set of plain images.

6 Embedded security measures

In this proposed CBIR scheme, we implement security measures on the images and the retrieved feature vector. The feature vector can reveal information to a fraudulent person, e.g., an image with a blue color on the higher side, histogram values will be higher. This could be an indication that the images may represent beaches, mountains, or sky categories. Hence, it is necessary to embed security not only to protect images but also their feature vectors. In this section, we give the details of the different security measures implemented in our CBIR system.

6.1 Image encryption

Figure 7 depicts a cryptographic image encryption scheme that is based on confusion–diffusion paradigm. Initially, a colored image is decomposed into its principal components. Then, individual components pass through several rounds of confusion to scramble their pixels. This image goes through several rounds of overall confusion–diffusion process so that its statistical attributes can be modified. The diffusion process can be executed for multiple rounds to achieve better secrecy. However, this process is applied on the pixel level which means that each and every pixel is involved in image encryption making it more computationally expansive. In our proposed CBIR system, we use image encryption based on bitplanes levels. Initially, each decomposed color channel is divided into its respective bitplanes. Then, these bitplanes are randomly shuffled. The first eight bitplanes are encrypted as the red channel, the second eight bitplanes as the blue channel, and so on. These three encrypted channels are combined together to form the initial shuffled image. Now, here a logistic map is considered where the seed value will be

treated as a cryptographic key and the sequence obtained from the logistic map can be mapped to a number of 24 digits. Later, the bit-wise XOR operation is performed to get the final encrypted image. This process is carried out for N number of rounds to increase the encryption security (see Fig. 8). Various algorithmic steps involved in the image encryption process are presented in Algorithm 4.

Algorithm 4 Image encryption

Input: RGB Image I_q.
Output: Final encrypted image E_I.
Parameter: Size of Q_I is $M \times N \times 3$.
Select an input I_q of size $M \times N \times 3$.
Decompose I_q into its color components I_R^q, I_G^q, & I_B^q
for *($i = 1$ to Number of Rounds)* **do**
$[R_0, R_1, \cdots, R_7] = BitplaneDecompose(I_R^q)$.
$[G_0, G_1, \cdots, G_7] = BitplaneDecompose(I_G^q)$.
$[B_0, B_1, \cdots, B_7] = BitplaneDecompose(I_B^q)$.
$[b_0, b_1, b_2, \cdots, b_{23}] = Shuffling(R_0, R_1, \cdots, R_7, G_0, G_1, \cdots, G_7, B_0, B_1, \cdots, B_7)$.
$r_M = [b_0, b_1, \cdots, b_7]$.
$g_M = [b_8, b_9, \cdots, b_{15}]$.
$b_M = [b_{16}, b_{17}, \cdots, b_{23}]$.
$k_r = LogisticMap(Seed_r)$.
$k_g = LogisticMap(Seed_g)$.
$k_b = LogisticMap(Seed_b)$.
$R_M = r_M \oplus k_r$.
$G_M = g_M \oplus k_g$.
$B_M = b_M \oplus k_b$.
$C_{E_I} = [R_M, G_M, B_M]$.
end
Return E_I

6.2 Security analysis of the image encryption scheme

A robust image encryption scheme output cipher should not reveal any visual or statistical information about the encrypted data. Often, the pixels in an image are redundant and highly correlated, which preserves some statistical patterns. These patterns could reveal information on the content of the image; hence, pixel correlation must be minimal to reduce the risk of statistical attacks. In this paper, we analyze our algorithm under the influence of histogram analysis, entropy, peak signal-to-noise ratio (PSNR), and correlation coefficient.

6.2.1 Histogram analysis

The histogram of an image gives the number of pixels that belong to particular pixel intensity. For an 8-bit image, the pixel intensities range from 0 to 255. In different channels of an RGB image, one bin value will be higher than the other. For a good cipher image, it is always considered that the histogram should be symmetric, i.e., all bin values should be similar. In Fig. 9, we have displayed the histogram of individual channels as then the histogram of each cipher image.

6.2.2 Entropy-based analysis

Shannon [34] proposed a way to represent the information held by an image in terms of intensity. Intensity of an image can be calculated by Eq. 4.

$$\mathrm{ENT} = -\sum_{i=0}^{2^h-1} h_i \log_2(h_i) \tag{4}$$

where i is the probability of the pixel intensity h_i. As, in the original image, the pixel intensities are very different. Figure 9 shows the entropies for red, green, and blue color channels are 7.8133, 7.4219, and 7.5184, respectively. In the cipher image, as all the intensities are almost the same, their entropies should also be close to 8. In our proposed scheme, the calculated entropies are 7.9981, 7.9983, and 7.9994 which are very close to 8. This demonstrates that our proposed scheme produces a highly random cipher image.

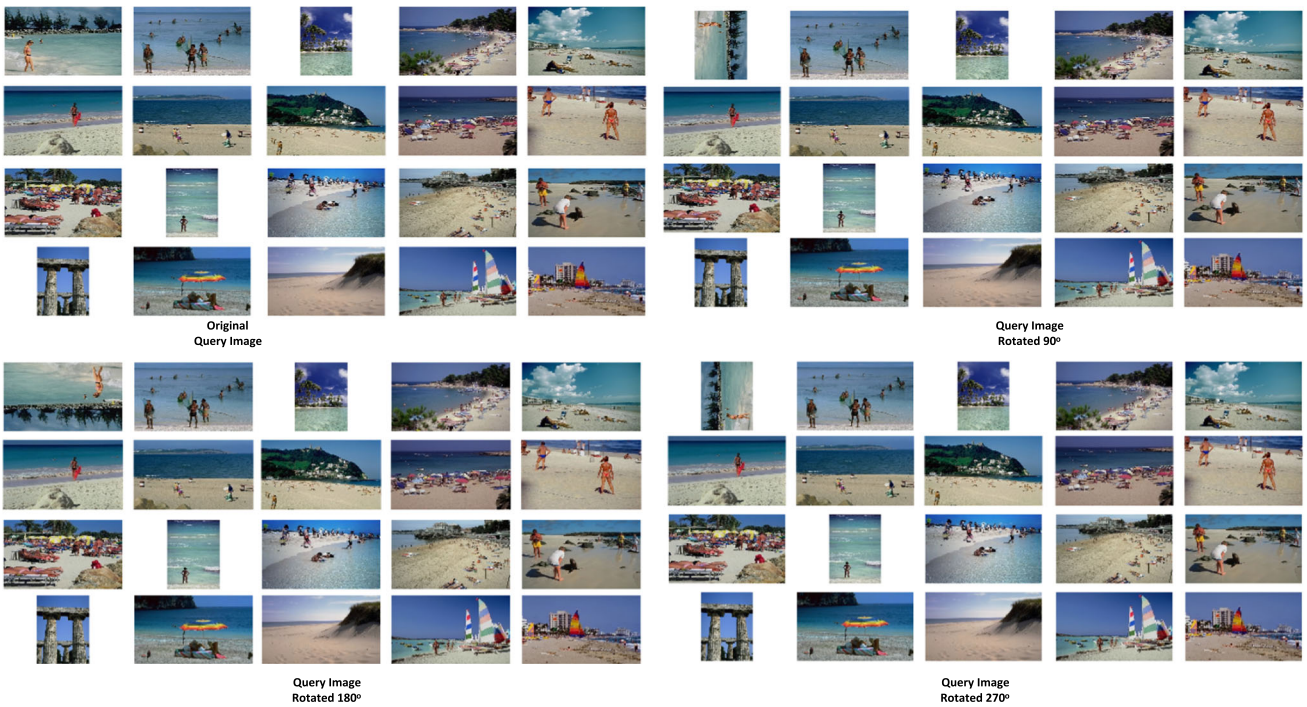


Fig. 15 Results of a beach image with various rotations

6.2.3 PSNR based analysis

The performance of a cryptographic system can also be measured in terms of the PSNR [35]. Let $I(x, y)$ be the original image which is encrypted using *key* to get C_I. The C_I undergoes noise attacks to obtain $C_I^{'}$. Now, $C_I^{'}$ is decrypted to get $I^{'}(x, y)$. If the dimension of the image is $l \times k$ and pixel depth is h, then PSNR can be evaluated using Eq. 5.

$$\text{PSNR} = 10 \times \log_{10} \frac{(2^h - 1)^2}{\frac{1}{l \times k} \sum_{x=1}^{l} \sum_{y=1}^{k} \{I^{'}(x, y) - I(x, y)\}^2} \tag{5}$$

For a cipher image, the value of the PSNR should be less than 10 dB. In this work, the proposed scheme produced the PSNR values of 7.94 dB, 7.19 dB, and 7.40 dB for the respective color component. This proves the proposed scheme produces a good cipher image.

6.2.4 Correlation coefficient-based analysis

In a plain image, neighboring pixel intensities are very similar, i.e., they are highly correlated. It is a requirement for a cipher image that all the pixel values are bifurcated randomly to reduce their correlation. As visualized in Fig. 10, for a plain individual color component, the values of the correlation coefficient in each direction are close to 1, but the values for the cipher images are close 0 which reflects that they are highly irrelated.

6.3 Feature vector encryption and decryption

Before transmitting the image feature database to the cloud server, each image feature is encrypted using a secret key K_i. Here, we have deployed encryption scheme based on ASPE that conserves scalar products. Distance comparisons are performed to find the neighbors of a query image feature vector. If the image feature vector $||f_v||$ is known to the attacker, he will come to know that f_v is located on the hypersphere, which is in turn centered through the origin with a radius $||f_v||$. The location of f_v will not be known, yet, the information which is revealed will identify the middle ground. In our proposed CBIR, this information is kept hidden by encrypting of both the feature vector $||f_v||$ and $||f_{vq}||$, and kNN is applied on such encrypted data. Initially, we have computed $||f_v||^2$ which is an image feature vector of $d + 1$ dimension. We create f_v from $(d + 1) - st$ dimension database feature vector f_v. The d dimension of f_v is the same as f_v. The dimension $(d + 1)$ is calculated as $-0.5||f_v||^2$. Then, the extended database feature vector are altered using ASPE. Likewise, before applying the ASPE query, the image feature vector is also extended to $(d + 1)$ dimension in f_{vq}. The $(d + 1)$ dimension in f_{vq} as is assigned the value 1.

The shortcoming of this basic technique is that the decoded feature vector f_{vq} all lies on a d-dimensional hyperplane with the unit vector in the $(d+1)-st$ dimension being the ordinary of the hyperplane. Since APSE is a direct change, the encoded feature vector all lies on a d-dimensional hyperplane in the transshipped space too. The attacker can decide the normal of that hyperplane in the changing space. By considering the typical in the first space and the ordinary in the changing space, the attacker gets unwanted data. To overcome this problem, we present an arbitrary factor. For each inquiry f_{vq}, we produce an irregular number $r > 0$ and scale f_{vq} by r, i.e., $f_{vq} = r(f_{vq}^T, 1)$. Theorem 2 shows this scaling does not affect the accuracy of the distance comparison operation. This operation is summarized in the following steps:

- **Key:** a $(d+1) \times (d+1)$ invertible matrix M.
- **Feature vector encryption function** $E_T(\cdot)$: Consider a database feature vector f_v. First, create a feature vector $f_v = (f_v^T, -0.5||f_v||^2)^T$ of dimension $(d+1)$. Second, compute an encrypted feature vector $f_v^{'} = M^T f_v$.
- **Query Image feature vector encryption function** $E_q(\cdot)$: Consider a query image feature vector f_{vq}. Generate a random number $r > 0$. Then, create a feature vector $f_{vq} = r(f_{vq}^T, 1)^T$ of dimension $(d+1)$. Finally, compute the encrypted query feature vector as $f_{vq}^{'} = M^{-1} f_{vq}$.
- **Distance comparison operator:** Let $f_{v1}^{'}$ and $f_{v2}^{'}$ be the encrypted feature vectors of f_{v1} and f_{v2}, respectively. To know whether the feature vector f_{v1} is nearer to the query image feature vector f_{vq} than f_{v2}, we calculate $(f_{v1}^{'} - f_{v2}^{'}) \cdot f_{vq}^{'} > 0$, where $f_{vq}^{'}$ is the encrypted feature vector f_{vq}.
- **Decryption Function:** Consider an encrypted feature vector $f_v^{'}$. The feature vector $f_v = \pi_d M^{T^{-1}} f_v^{'}$, where π_d is a $d \times (d+1)$ matrix and $\pi_d = (I_d, 0)$, where I_d is $d \times d$ identity matrix.

7 Design objectives of the proposed CBIR system

In this section, we discuss the need for embedding security at the different levels of a CBIR system. Then, the other functional requirements are presented.

7.1 Eavesdropping

The user's objective is to transmit images from the cloud to only a legitimate person. That information must be kept secret from any other unauthorized entity. For that, all the images and feature vectors are encrypted with different keys. When the data are captured from the communication channel an eavesdropper will not able to learn the content of the transmitted messages.

7.2 Untrusted CSP

The cloud server cannot be fully trusted; therefore, users must secure their data before transfer it to the cloud. In our proposed CBIR, the cloud server is assumed to be honest-but-curious just for the retrieval part, i.e., he will perform the retrieval task according to the designated algorithm and return the actually generated results back to the KMC without any interference. All the images and image feature vectors are encrypted and image similarity is computed on encrypted feature vectors. This prevents the cloud from learning the content of its hosted data.

7.3 Authorized registered user

Most solutions in the literature assume that the registered user is a fully trusted entity. However, in real-life implementations, insider threats present a serious challenge. In our proposed CBIR system, a registered data owner provides only the visual image encryption algorithm. The KMC sends him a temporary key to encrypt the query image feature vector. Then, this encrypted feature vector is transmitted back to the KMC. The KMC re-encrypt this using M^{-1} and transfers the data to the cloud. When the cloud sends the images back to the KMC, it redirects them to the user who possesses a key that is only related to those images.

7.4 Image and feature vector privacy

The proposed CBIR provides image privacy from the CSP, unauthorized entity, and in some cases even from authorized entities. Only the data owner has all the original image and feature vector content. Content stored on the cloud is encrypted with keys that are kept secret on the KMC. At the same time similarity matching is performed on the encrypted images and the encrypted feature vector.

7.5 Retrieval accuracy

Image feature vectors are constructed using local invariant features which combine color, texture, and shape information. The proposed CBIR system exploits the strong association between the actual content and local image features.

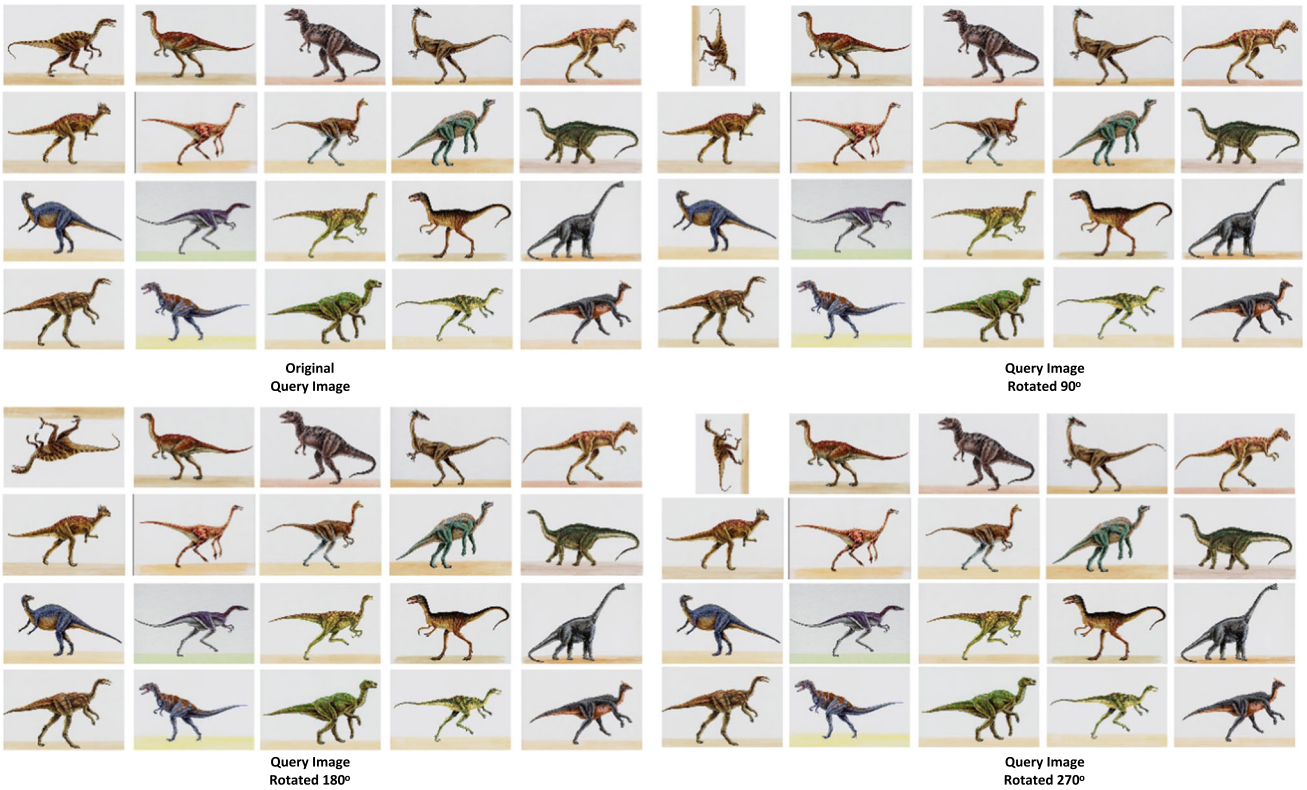


Fig. 16 Results of a dinosaur image with various rotations

7.6 Overall system efficiency

Real-time applications require low computation time delay. Hence, the proposed CBIR system does not incorporate any multiplicative group-based scheme. It also reduces the number of steps required to transmit the images from the cloud to an authorized person.

8 Security analysis

Theorem 1 *Scalar-product-preserving encryption is distance-recoverable.*

Proof Let $f_{v1}^{'}$ and $f_{v2}^{'}$ be the encrypted points of f_{v1} and f_{v2}, respectively, in a database. A function f can be defined as

$$f(f_{v1}^{'},f_{v2}^{'}) = \sqrt{f_{v1}^{'}.f_{v1}^{'} - 2(f_{v1}^{'}.f_{v2}^{'}) + f_{v2}^{'}.f_{v2}^{'}}$$

Since the encryption process conserves the scalar product, we can say that

$$RHS = \sqrt{f_{v1}.f_{v1} - 2(f_{v1}.f_{v2}) + f_{v2}.f_{v2}}$$

Let E is an encryption function and $E(f_{v1}, K)$ is the encrypted value of a feature vector f_v with the key K. E is an ASPE subject to E preserves the scalar products, two other types do not preserve, i.e.,

$[i] f_{vi}.f_{vq} = E(f_{vi}, K).E(f_{vq}, K)$ for any f_{vi} in DB and f_{vq} feature vector of query image.

$[ii] f_{vi}.f_{vj} \neq E(f_{vi}, K).E(f_{vj}, K)$ for any f_{vi} and f_{vj} in DB.

In Definition 1, the encrypted value of a query feature vector f_{vq} should be different to any point f_{vj} in DB, even when $f_{vq} = f_{vj}$. This requires that query feature vector and database feature vector must be encrypted such that the encryption functions $E_T()$ and $E_q()$ in the encryption outline are different.

The scalar product of f_v, f_{vq} (signified as column vectors) is denoted as $f_v^T I f_{vq}$ where f_v^T is the transpose of f_v and I is an $d \times d$ identity matrix. I can be further decomposed to MM^{-1} for any invertible matrix M, i.e., $f_v^T f_{vq} = (f_v^T M)(M^{-1} f_{vq})$.

If we set $f_v^{'} = E_T(f_v, K) = M^T f_v$ and $f_{vq}^{'} = E_q(q, k) = M^{-1} q$, getting the value of f_v and f_{vq} is not possible for anyone from $f_v^{'}$ and $f_{vq}^{'}$ without knowing M. Likewise, the $f_v^{'T} f_{vq}^{'} = f_v^T M M^{-1} f_{vq} = f_v^T f_{vq}$, i.e., the scalar product is conserved. Assume $f_{v1}^{'}$ and $f_{v2}^{'}$ are the encrypted feature vector of image 1 and image 2 of a database, then $f_{v1}^{'T} f_{v2}^{'} = f_{v1}^T M M^T f_{v2}$, which is different from $f_{v1}^T f_{v2}$.

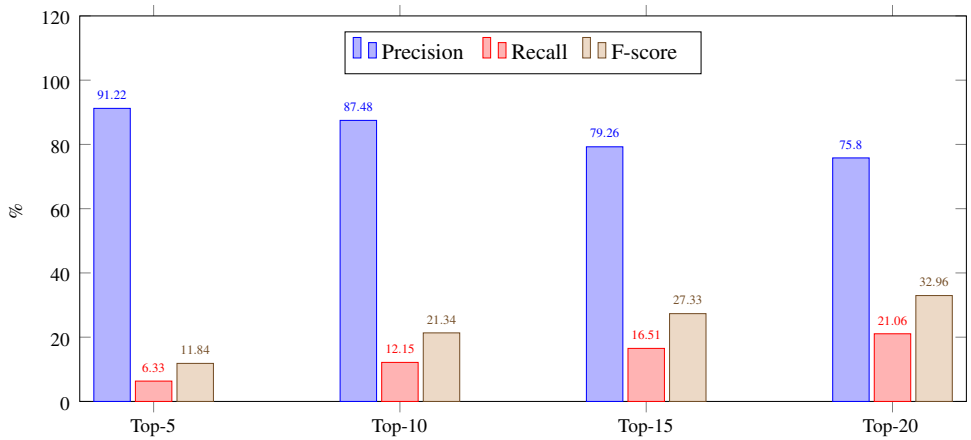


Fig. 17 Mean average precision, recall, and F-score for different numbers of outcome images of COIL-100 database using the Euclidean similarity measurement

Therefore, ASPE is applied by using M and M^{-1} as the alterations for a database image feature vector and query image feature vector, correspondingly. □

Theorem 2 *Let $f_{v1}^{'}$ and $f_{v2}^{'}$ be the encrypted feature vector of f_{v1} and f_{v2}, respectively. The feature vector f_{v1} is nearer to a query image feature vector f_{vq} than f_{v2}, the evaluation of $(f_{v1}^{'} - f_{v2}^{'}).f_{vq}^{'} > 0$, where $f_{vq}^{'}$ is the encrypted feature vector f_{vq}.*

Proof Consider that

$$\begin{aligned}(f_{v1}^{'} - f_{v2}^{'}).f_{vq}^{'} &= (f_{v1}^{'} - f_{v2}^{'})^T.f_{vq}^{'} \\ &= (M^T\hat{f_{v1}} - M^T\hat{f_{v2}})^T M^{-1}\hat{f_{vq}} \\ &= (\hat{f_{v1}} - \hat{f_{v2}})^T\hat{f_{vq}}.\end{aligned}$$

The scalar product of these two $(d+1)$ dimension feature vectors can be denoted as

$$\begin{aligned}&(f_{v1} - f_{v2})^T(rf_{vq}) + (-0.5||f_{v1}||^2 + 0.5||f_{v2}||^2)r \\ &= 0.5r(||f_{v2}||^2 - ||f_{v1}||^2 + 2(f_{v1} - f_{v2})^T f_{vq}) \\ &= 0.5r(d(f_{v2}, f_{vq}) - d(f_{v1}, f_{vq})) > 0 \\ &= d(f_{v2}, f_{vq}) \geq d(f_{v1}, f_{vq})\end{aligned}$$

□

9 Upgradation in images and indexes

There have been times when the image owner wants to update either the stored images or the encrypted feature vector to improve the retrieval accuracy. When there is an update in the number of images, then it must also reflect on their individual indexes. Our proposed CBIR scheme allows the image owner to change the number of images in a particular database or modify the stored encrypted feature vectors.

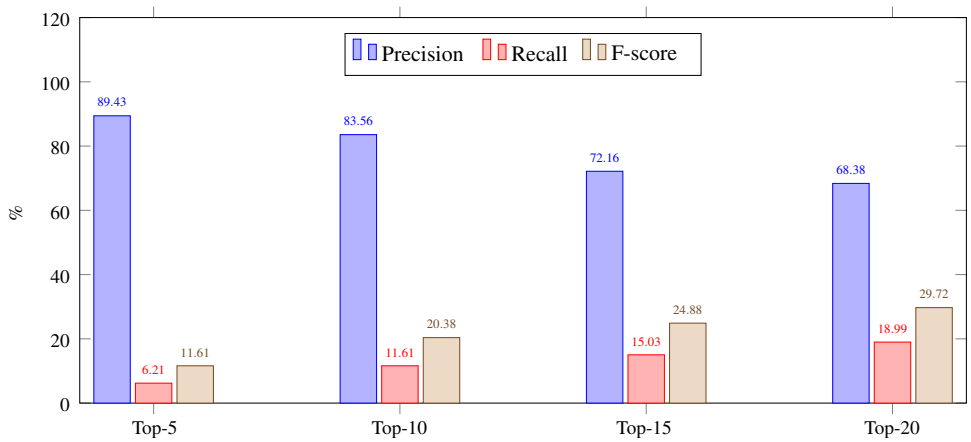


Fig. 18 Mean average precision, recall, and F-score for different numbers of outcome images of COIL-100 database using the encrypted similarity measurement

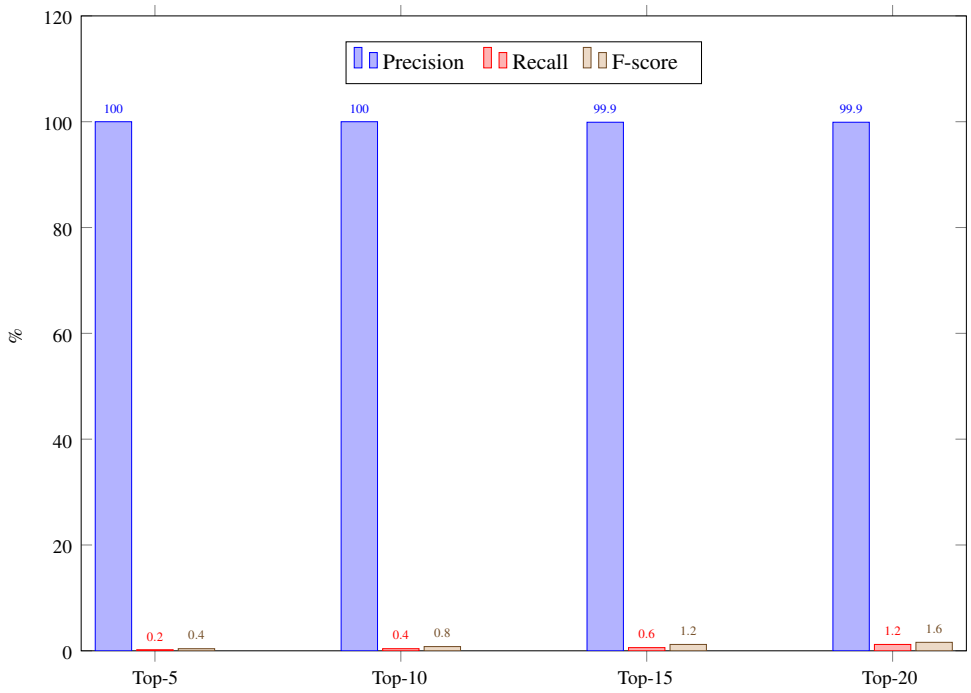


Fig. 19 Mean average precision, recall, and F-score for different numbers of outcome images of color medical image database using Euclidean similarity measurement

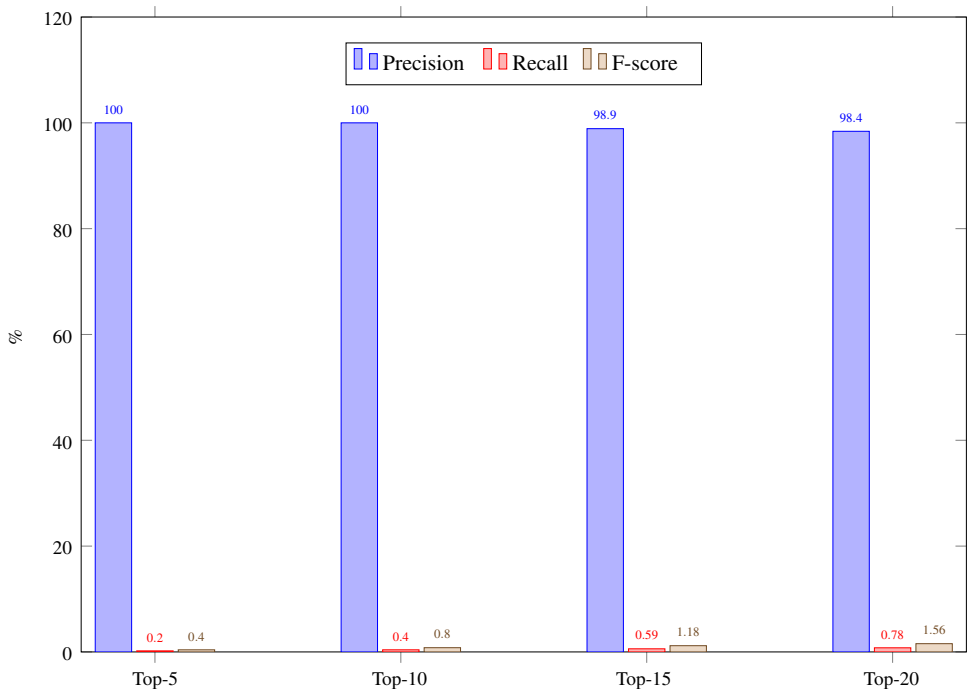


Fig. 20 Mean average precision, recall, and F-score for different numbers of outcome images of color medical image database using encrypted similarity measurement

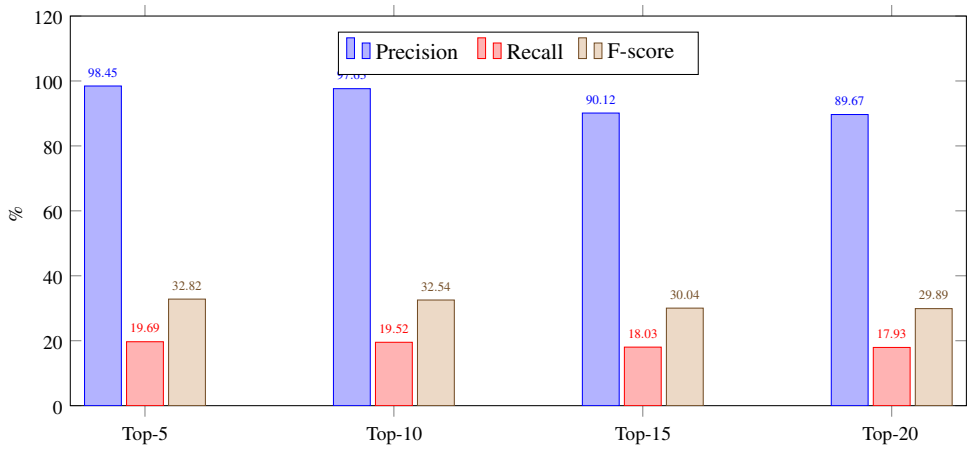


Fig. 21 Mean average precision, recall, and F-score for different numbers of outcome images of PRODUCE-1400 database using Euclidean similarity measurement

9.1 Insert new images

When the image owner wants to insert new images, he extracts the feature vector of each image and then encrypts them using the key received from the KMC. The owner requests the KMC to generate some new image encryption keys and transmit them back to him. Then, the image owner encrypts his images individually. Finally, the owner sends the new encrypted images and their respective feature vectors to the CSP. When the CSP receives these images, it stores them in a respective database and increases its indexes by the number of images. After adding those images to the database, the CSP inform the KMC about the upgradation with the list of indexes.

9.2 Remove images

If the image owner wants to delete some of his images, then he sends their indexes to the CSP. The CSP removes the indexed images and updates the index of the rest of the database. Then, the image owner communicates those images to the KMC so that the KMC can also delete those image indexes from its list.

9.3 Updating the stored data

Image owner sometimes wants to remove some stale images or may update the feature extraction method. When an image is updated, the visual image features are then communicated to authorized users to use the new information to perform a feature extraction query. The image

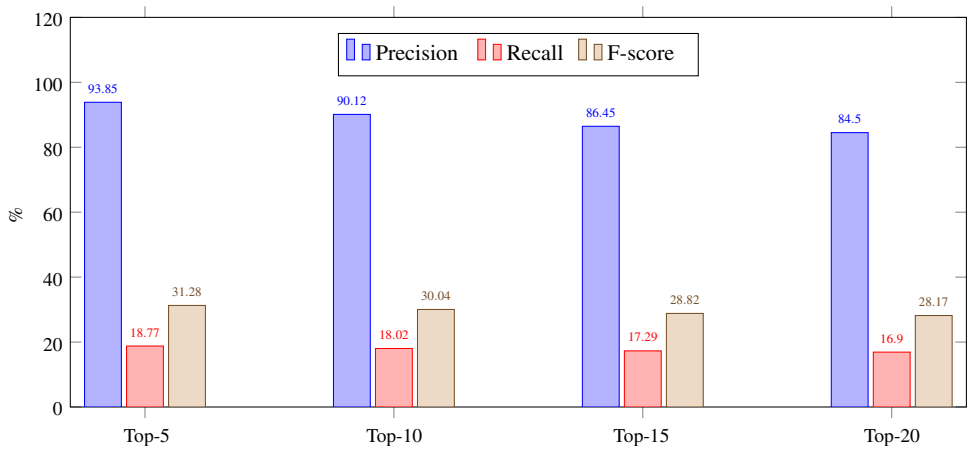


Fig. 22 Mean average precision, recall, and F-score for different numbers of outcome images of PRODUCE-1400 database using encrypted similarity measurement

owner then performs the feature extraction from all database images and sends the result to the cloud after encrypting them with M. Then, the CSP replaces the old stored data and with the new ones. There may be a chance that the data owner wants to update the index of the stored image, then the owner must communicate the updated list to the CSP as well as the KMC.

10 Performance evaluation

In this session, we evaluate the performance of the proposed CBIR system. The experiments were performed using MATLAB installed on a system having, Intel(R) Core(TM) i7-4770 CPU@3.4-GHz, 4GB RAM.

The performance of the proposed system is measured using the precision, recall, and F-score parameters. Precision is described as the ratio of relevant images retrieved to the total images retrieved. If T_I is the total number of images of the same category present in the database and T_R, is the total number of images retrieved which include relevant as well as non-relevant images then

$$\text{precision} = \frac{T_I \cap T_R}{T_I}$$

Recall is the ratio of the total relevant image retrieved to the total images present for the same category in the database

$$\text{recall} = \frac{T_I \cap T_R}{D_I}$$

where D_I in the total number of images present in the respective database of a particular category.

F-measure or F-score is the harmonic mean of the precision and recall and is the single-valued measures of the overall system.

$$F\_\text{Score} = \frac{2 \times \text{precision} \times \text{recall}}{(\text{precision} + \text{recall})}$$

In our experiments, we use the Wang image dataset. This dataset [36] has 1000 corel images classified into ten categories, namely people, beaches, buildings, buses, dinosaurs, elephants, flowers, horses, mountains, and foods. Each category has 100 images. Figure 11 shows the result for each category of images on the basis of a different number of outcome images. Figure 12 shows the proposed CBIR system results based on the original feature vector and using the Euclidean distance as a similarity measure. As similar images are retrieved from an encrypted environment using encrypted feature vector-based similarity, Fig. 13 shows the results for the different number of output images. It can be observed that the encrypted similarity-based results are on the lower side when compared to the Euclidean-based results.

We compare our proposed CBIR system using the encrypted feature vector-based similarity measures. We compare the performance of the proposed system with Xu et al. [37], Xu et al. [38], Majhi et al. [39], and Kundu et al. [40]. Our solution enhances the percentage of average precision by 21.41%, 26.16%, 3.12%, and 8.92% when compared to the aforementioned solutions from the literature, respectively. The comparison based on average precision, recall, and f-score are displayed in Fig. 14.

10.1 Result comparison based on different machine learning algorithms

In recent literature, researchers started combining different approaches to classify their data and then retrieve results based on individual category. In this work, we evaluate our proposed solution on similar basis. We classify our database through four well-known techniques, namely decision tree [41], random forest [42], support vector machine (SVM) [43], and multilayer perceptron (MLP) [44]. Initially, the database is randomly divided into 80 : 20 ratio, i.e., 80% training images and 20% testing images. Afterward, all the mentioned techniques are applied in order. Their classification efficiencies are 58.23%, 68.23%, 70.29%, and 57.35%, respectively. Since the SVM provides the best efficiency, it is selected for the retrieval process. We draw results for the Top-10, Top-15, Top-20, Top-25, Top-30, Top-35, Top-40, Top-45, and Top-50 images and their respective precision values in percentage are 68.4, 68.66, 67.75, 69.14, 71.67, 72.02, 72.5, 76, and 78.21.

10.2 Rotation invariance

In our proposed CBIR, the extracted features invariant and the rotation of the image will not have any effect on the retrieved results. When a user uploads an image to the cloud server, one cannot guarantee about the its orientation. In a conventional CBIR cannot deal with such situation as different orientation will produce different retrieved results which will degrade the overall system performance. Hence, in this paper, we have chosen such image features who can produce similar results with different orientations of query image. Figure 15 shows a beach image and its retrieved results. A similar experiment was performed on the Dinosaur image, see Fig. 16.

10.3 COIL dataset based result analysis

The Columbia Object Image Library (COIL-100) [45] has 7200 object images divided into 100 different

categories such that each category has 72 images. Figure 17 gives the average precision, recall, and F-score values for various numbers of output images using the Euclidean distance on original feature vectors. In Fig. 18, our average results based on the encrypted similarity over secure trapdoors are displayed.

10.3.1 Result comparison based on different machine learning algorithms

Classification efficiency experimental results obtained from the decision tree, random forest, SVM, MLP classification mechanism are 83.79%, 98.73%, 98.76%, and 97.79%, respectively, where data is divided into 80 : 20 ratio training and testing portions. SVM exhibits the best performance among all studied mechanisms; hence, we draw the retrieval results based on that method only. The retrieved precision values of Top-10, Top-15, Top-20, Top-25, Top-30, Top-35, Top-40, Top-45, and Top-50 images are, respectively, 98.61%, 98.52%, 98.75%, 98.53%, 98.79%, 98.61%, 98.23%, 98.08%, and 98.78%.

10.4 Color medical image dataset-based result analysis

This color medical image dataset contains a total of 10000 images from four categories. These categories are skin cancer, retina, endoscopy, and whole slide image (WSI). In each category, 2500 images are present. This dataset is formed by combining different semantically similar datasets. To make it workable in this environment, a region of interest of the whole slide images is extracted and further decomposed into 2500 unique 2014×2014 regions of interest (ROI) whole slide images. Figure 19 shows the average precision, recall, and F-score values for the various number of output images using the Euclidean distance on the original feature vector. Figure 20 shows the average results based on the encrypted similarity over secure trapdoors.

10.4.1 Result comparison based on different machine learning algorithms

We perform an experiment based on the different classification mechanisms. The results obtained on the classification efficiency are 94.63%, 97.56%, 99.45%, and 99.02%, respectively, for Decision Tree, Random forest, SVM, MLP where data is divided into 80 : 20 ratio training and testing portions. Since SVM performs the best among all studied methods, it is used to study the retrieval results. The retrieved precision values of Top-5, Top-10, Top-15, and Top-20 images are, respectively, 100%, 100%, 100%, and 99.9%, respectively.

From the above experimental results analysis, we conclude that even after applying classifications, the retrieval results are not significantly improved. Implementing classification increases the computational overhead with no significant benefits.

10.5 Produce-1400 dataset based result analysis

Produce-1400 [46] has 1400 images divided into 14 different categories where each category has 100 images. In Fig. 21, we plot the average precision, recall, and F-score values for various number of output images using the Euclidean distance on the original feature vector. Figure 22 shows the average results based on the encrypted similarity over secure trapdoors.

10.5.1 Result comparison based on different machine learning algorithms

Using different classification mechanism, we obtain the classification efficiency of 84.24%, 95.21%, 96.32%, and 95.38%, respectively, for decision tree, random forest, SVM, MLP where data is divided into 80 : 20 ratio training and testing portions. Only the top performer SVM is used to draw the retrieval results. The retrieved average precision values of Top-5, Top-10, Top-15, and Top-20 images are 99.45%, 98.63%, 97.12%, and 90.67%, respectively. From the above results analysis, we conclude that even after applying classifications, the retrieval results are not significantly enhanced. Therefore, implementing classification will only increase the computational overhead.

11 Conclusion

In this article, a secure CBIR system has been proposed to retrieve color image data securely. The proposed solution comprises four entities Image Owner, Registered User, CSP, and KMC. The data owner can secure the transfer of his images to the cloud using the proposed image encryption technique. This encryption method is practical in real-time applications where the number of confusion-diffusion rounds are less compared to the conventional confusion–diffusion color image encryption techniques. Additionally, the image owner transmits the encrypted feature vector, which is received from the ASPE. This prevents any malicious entity from learning any information from the encrypted feature vector. The proposed CBIR system also works with both natural and medical images. The experimental results are promising and be used safely to secure medical data. One of the future directives will be to enhance retrieval performance further while increasing the security level.

Acknowledgements The author Mr. Sumit Kumar (Admission No: 2015DR0056) is supported by the institute Ph.D. scholarship, IIT[-ISM], Dhanbad, Jharkhand, India.

Declarations

Conflict of Interest The authors declare that there are no conflicts of interest regarding the publication of this paper.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

1. 53 incredible facebook statistics and facts. https://www.brandwatch.com/blog/facebook-statistics/#:~:text=Facebook%20generates%204%20new%20petabytes,have%20been%20uploaded%20to%20Facebook. Accessed on 27 Dec 2020
2. Belguith S, Kaaniche N, Hammoudeh M, Dargahi T (2020) Proud: verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted IOT applications. Fut Gener Comput Syst 111:899–918
3. Kumar S, Pradhan J, Pal AK (2017) A CBIR scheme using GLCM features in DCT domain. In: Proceedings of the 2017 IEEE international conference on computational intelligence and computing research
4. Kumar S, Pradhan J, Pal AK (2018) A CBIR technique based on the combination of shape and color features. Adv Comput Commun Parad 706:737–744
5. Huang J, Kumar SR, Mitra M, Zhu W-J, Zabih R (1997) Image indexing using color correlograms. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, pp 762–768. IEEE
6. Pradhan J, Ajad A, Pal AK, Banka H (2019) Multi-level colored directional motif histograms for content-based image retrieval. Vis Comput 36:1–22
7. Zhang D, Guojun L (2004) Review of shape representation and description techniques. Pattern Recogn 37(1):1–19
8. Khotanzad A, Lu J-H (1990) Classification of invariant image representations using a neural network. IEEE Trans Acoust Speech Signal Process 38(6):1028–1038
9. Hammoudeh M, Newman R, Dennett C, Mount S (2013) Interpolation techniques for building a continuous map from discrete wireless sensor network data. Wirel Commun Mob Comput 13(9):809–827
10. Hammoudeh M, Newman R (2015) Information extraction from sensor networks using the watershed transform algorithm. Inform Fusion 22:39–49
11. Majhi M, Pal AK, Pradhan J, Islam SKH, Khan MK (2021) Computational intelligence based secure three-party CBIR scheme for medical data for cloud-assisted healthcare applications. Multimed Tools Appl 1–33
12. Song DX, Wagner D, Perrig A (2000) Practical techniques for searches on encrypted data. In: Proceeding 2000 IEEE symposium on security and privacy. S&P 2000, pp 44–55. IEEE
13. Chang Y-C, Mitzenmacher M (2005) Privacy preserving keyword searches on remote encrypted data. Int Conf Appl Cryptogr Netw Sec 3531:442–455
14. Curtmola R, Garay J, Kamara S, Ostrovsky R (2011) Searchable symmetric encryption: improved definitions and efficient constructions. J Comput Sec 19(5):895–934
15. Shashank J, Kowshik P, Srinathan K, Jawahar CV (2008) Private content based image retrieval. In: Proceedings of the 2008 IEEE conference on computer vision and pattern recognition, pp 1–8. IEE
16. Lu W, Swaminathan A, Varna AL, Wu M (2009) Enabling search over encrypted multimedia databases. Int Soc Opt Photon Media Forens Sec 7254:725418
17. Yuan J, Yu S, Guo L (2015) Seisa: secure and efficient encrypted image search with access control. In: Proceedings of the 2015 IEEE conference on computer communications (INFOCOM), pp 2083–2091. IEEE
18. Xia Z, Xiong NN, Vasilakos AV, Sun X (2017) Epcbir: an efficient and privacy-preserving content-based image retrieval scheme in cloud computing. Inf Sci 387:195–204
19. Shen M, Cheng G, Zhu L, Xiaojiang D, Jiankun H (2020) Content-based multi-source encrypted image retrieval in clouds with privacy preservation. Fut Gener Comput Syst 109:621–632
20. Xia Z, Zhu Y, Sun X, Qin Z, Ren K (2015) Towards privacy-preserving content-based image retrieval in cloud computing. IEEE Trans Cloud Comput 6(1):276–286
21. Yuan X, Wang X, Wang C, Squicciarini AC, Ren K (2016) Towards privacy-preserving and practical image-centric social discovery. IEEE Trans Depend Sec Comput 15(5):868–882
22. Bellafqira R, Coatrieux G, Bouslimi D, Quellec G (2015) Content-based image retrieval in homomorphic encryption domain. In: Proceeding of the 2015 37th annual international conference of the ieee engineering in medicine and biology society (EMBC), pp 2944–2947. IEEE
23. Weng L, Amsaleg L, Morton A, Marchand-Maillet S (2014) A privacy-preserving framework for large-scale content-based information retrieval. IEEE Trans Inf Forens Sec 10(1):152–167
24. Zhou J, Cao Z, Dong X, Lin X (2015) Ppdm: a privacy-preserving protocol for cloud-assisted e-healthcare systems. IEEE J Select Top Signal Process 9(7):1332–1344
25. Shengshan H, Wang Q, Wang J, Qin Z, Ren K (2016) Securing sift: privacy-preserving outsourcing computation of feature extractions over encrypted image data. IEEE Trans Image Process 25(7):3411–3425
26. Xia Z, Wang X, Zhang L, Qin Z, Sun X, Ren K (2016) A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. IEEE Trans Inf Forens Sec 11(11):2594–2608
27. Steiner M, Tsudik G, Waidner M (1996) Diffie-hellman key distribution extended to group communication. In: Proceedings of the 3rd ACM conference on computer and communications security, pp 31–37
28. William Rowan Hamilton (1866) Elements of quaternions. Green, & Company, Longmans
29. Ell TA, Sangwine SJ (2006) Hypercomplex fourier transforms of color images. IEEE Trans Image Process 16(1):22–35
30. Hosny KM, Darwish MM (2019) Feature extraction of color images using quaternion moments. Recent Adv Comput Vis 804:141–167
31. Mukundan R, Ong SH, Lee PA (2001) Image analysis by tchebichef moments. IEEE Trans Image Process 10(9):1357–1364
32. Pradhan J, Kumar S, Pal AK, Banka H (2018) Texture and color visual features based CBIR using 2D DT-CWT and histograms. Int Conf Math Comput 834:84–96
33. Feldman M (2011) Hilbert transform in vibration analysis. Mech Syst Signal Process 25(3):735–802
34. Lin J (1991) Divergence measures based on the shannon entropy. IEEE Trans Inf Theory 37(1):145–151
35. Huynh-Thu Q, Ghanbari M (2008) Scope of validity of PSNR in image/video quality assessment. Electron Lett 44(13):800–801
36. Li J, Wang JZ (2008) Real-time computerized annotation of pictures. IEEE Trans Pattern Anal Mach Intell 30(6):985–1002

37. Yanyan X, Zhao X, Gong J (2019) A large-scale secure image retrieval method in cloud environment. IEEE Access 7:160082–160090
38. Yanyan X, Gong J, Xiong L, Zhengquan X, Wang J, Shi Y (2017) A privacy-preserving content-based image retrieval method in cloud environment. J Vis Commun Image Represent 43:164–172
39. Majhi M, Pal AK, Islam SKH, Khurram KM (2021) Secure content-based image retrieval using modified Euclidean distance for encrypted features. Trans Emerg Telecommun Technol 32(2):e4013
40. Kundu MK, Chowdhury M, Bulo SR (2015) A graph-based relevance feedback mechanism in content-based image retrieval. Knowl Based Syst 73:254–264
41. Chandra B, Varghese PP (2008) Fuzzy sliq decision tree algorithm. IEEE Trans Syst Man Cybern B (Cybern) 38(5):1294–1301
42. Barjinder K, Dinesh S, Partha PR (2019) Age and gender classification using brain–computer interface. Neural Comput Appl 31(10):5887–5900
43. Aslahi-Shahri BM, Rahmani R, Chizari M, Maralani A, Eslami M, Golkar MJ, Ebrahimi A (2016) A hybrid method consisting of GA and SVM for intrusion detection system. Neural Comput Appl 27(6):1669–1676
44. Bahrammirzaee A (2010) A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. Neural Comput Appl 19(8):1165–1195
45. Nene Sameer A, Nayar Shree K, Murase H, et al (1996) Columbia object image library (coil-100)
46. Produce-1400 Database. http://www.ic.unicamp.br/~rocha/pub/downloads/tropical-fruits-DB-1024x768.tar.gz/. Accessed: 2018-04-07