The ISC Int'l Journal of Information Security
ISeCure

January 2021, Volume 13, Number 1 (pp. 1–17)
http://www.isecure-journal.org

OT-Feature Extraction on Scrambled Images with Instantaneous Clustering for CBIR Scheme in Cloud Computing

K. Nalini Sujantha Bel [1,*] and I. Shatheesh Sam [2]

[1] Research Scholar, Reg. No. 18123112162001, Dept. of Computer Science, Nesamony Memorial Christian College affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli 627 012,Tamil Nadu, India.
[2] Associate Professor, Dept. of PG Computer Science, Nesamony Memorial Christian College affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli 627 012,Tamil Nadu, India.

ARTICLE INFO.

Article history:
Received: November 23, 2019
Revised: July 7, 2020
Accepted: October 10, 2020
Published Online: October 18, 2020

Keywords:
Content-Based Image Retrieval (CBIR), Clustering, Feature Extraction, Otsu's Threshold (OT), Scrambled Images

Type: Research Article

doi: 10.22042/isecure.2020.209056.497

ABSTRACT

A novel feature extraction algorithm using Otsu's Threshold (OT-features) on scrambled images and the Instantaneous Clustering (IC-CBIR) approach is proposed for Content-Based Image Retrieval in cloud computing. Images are stored in the cloud in an encrypted or scrambled form to preserve the privacy content of the images. The proposed method extracts the features from the scrambled images using the Otsu's threshold. Initially, the Otsu's threshold is estimated from the scrambled image and based on this threshold the image is divided into two classes in the first iteration. Again, the new threshold values are estimated from two classes. The difference between the new threshold and the previous threshold gives two features. This process is repeated for L number of iteration to obtain the complete OT-features of the scrambled image. This paper also proposes an instantaneous clustering approach (IC-CBIR) where the image is moved into a cluster as soon as the image is uploaded by the image owner. Therefore while retrieving the images, the images near to a particular cluster are matched instead of matching with a complete set of image features in the dataset which reduces the search time. The performance of the proposed algorithm is being tested using four different types of the dataset such as Corel 10K, Misc, Oxford flower, and INRIA Holidays dataset. The experimental evaluation reveals that the proposed method outperforms better than the traditional CBIR algorithm on encrypted images in terms of precision, time of search and time of index construction.

1 Introduction

Due to the large storage capacity provided by the cloud service provider, outsourcing of image to cloud and retrieval of the image by Content-Based Image Retrieval (CBIR) [1–3] has become very attractive. Extracting the feature on encrypted images is a major concern in cloud content-based image retrieval. Cloud computing [4, 5] has become more prevalent in different fields because of its high storage ability. A large number of cloud service providers are available such as Amazon cloud drive, Google, Flicker, Apple in cloud, etc. Content-based image search and

[*] Corresponding author.
Email addresses: nalinisb85@gmail.com, shatheeshsam@yahoo.com
ISSN: 2008-2045 © 2020 ISC. All rights reserved.

retrieval are essential functions in searching and retrieving images from cloud storage. Such content-based image retrieval [6–8] is used in a wide range of applications such as medical image diagnosis, criminal investigation, biometric search, personal image management, etc. Privacy-preserving is a significant concern in a cloud computing paradigm. Because the photos/images that are uploaded to the cloud storage may contain sensitive/confidential information and this sensitive information may make the cloud user to minimize the use of cloud storage. In content-based image retrieval from the cloud server, the user directly uploads the query images which can contain sensitive information. Also, the image outsourced by the image owner will be stored in the cloud server without encryption, where an untrustworthy cloud server can view the sensitive information of the image owner and image user.

Image encryption [9, 10] is one of the techniques that preserve the confidentiality of an image. In this, the user can encrypt the image using a key which converts the plain image into a cipher image. This cipher image which is uploaded to the cloud storage can preserve the sensitive information of users since the image in cloud storage is available in encrypted form. The owner's image is also stored in the cloud server in encrypted form and the features are extracted from the encrypted image and index is constructed using the extracted features. When the user sends an encrypted image request for retrieval, the feature is extracted from the encrypted query image and the best match results will be returned to the user. The encryption method makes the user and owner's image content more secure, but it is a challenging task to extract the features from the encrypted images. Several researchers are doing in content based image retrieval from encrypted images and few of them are discussed below.

Searchable Encryption (SE) techniques [11–14], allows the user to search the query image over the encrypted dataset images. The SE technique such as the Boolean search is most commonly used in retrieving text documents. This technique is extended in images which can extract the local features from all the images available in the database and generate visual words by clustering process. The content of the image is preserved by Min-hash and order preserving.

Erkin *et al.* [15] introduced a face recognition system that recognizes the faces from the privacy preserved biometric faces. This method protects only the confidentiality of the query image because the system matches the feature of the signature extracted from the encrypted query image with the un-encrypted image present in the dataset. In [16], the images are encrypted with homomorphic encryption and the SIFT [17] features are extracted from the encrypted image. This methods [18, 19] has high computational complexity since it needs 1-round communication between the cloud server and the user. SIFT algorithm is commonly used in extracting the features from the images. Privacy-Preserving SIFT [20] (PPSIFT) is used to extract the features from the encrypted images. In this method, both the query image and the owner's image are initially encrypted to assure privacy preserving. This method performs a Difference of Gaussian (DOG) transform on the encrypted image. This DOG transform is similar to the transforms such as DWT [21], and DCT [22] where the transform is performed in encrypted images. This PPSIFT technique also focuses on the behaviour of DOG transform in paillier cryptosystem. The PPSIFT algorithm can extract local extrema features without performing multiple rounds of communication between the cloud server and the user. But this PPSIFT technique requires a pre-communication for data synchronization. In [23], similar to SIFT features, SURF features are extracted from the encrypted images.

Lu *et al.* [24] compared the retrieval efficiency, precision and storage overhead by proposing a retrieval algorithm based on homomorphic encryption. The experimental results of Lu *et al.* show that the homomorphic encryption is time-consuming since it provides high security. Xia *et al.* proposed [25] an image retrieval method which uses SIFT features and Earth Movers Distance (EMD). The frequency histogram of the distance is calculated by the EMD algorithm. Cheng *et al.* [26] used a Markov process for image retrieval in the encrypted domain and this method extracts the Markov features directly from the encrypted image. Degang *et al.* [27] proposed a quantization based image retrieval on three bit. This method applies the asymmetric algorithm after assigning three bit to each dimension. This method overcomes the privacy issues, but the complexity on the user side is high. Bellafgira *et al.* [28] proposed a feature extraction method in encrypted images and Ferreira *et al.* [29] proposed an encryption scheme for image retrieval which separately processes texture and color information. The protected color information is encrypted by deterministic encryption, and the texture information is encrypted by the random encryption algorithm. In this method, the color histogram is used to extract image retrieval which reduces the complexity of users.

In [30, 31], the features of the images are extracted from the dataset images. After extracting the features, pre-filter tables are created by locality sensitive hashing to increase the speed of searching the query image. The extracted features are protected by a secure kNN algorithm and the dataset images are protected by a standard stream cipher. This method provides a way to detect illegal distributions using a watermarked based protocol. This method extracts the features in four different ways such as Scalable Color Descriptor (SCD), Color Structure Descriptor (CSD), Color Layout Descriptor (CLD) and Edge Histogram Descriptor (EHD). The SCD descriptor represents the features in HSV (Hue, Saturation and Value) color space where it uses Haar transform-based encoding. CSD descriptor uses a structuring window to detect localized color distribution. The color structure histogram is constructed in Hue-Min-Max-Difference (HMMD) color space. The EHD descriptor detects the distribution of edges in the spatial domain.

In [32], the authors proposed an CBIR method in cloud environment that uses orthogonal decomposition. In this method, the image is decomposed into different components, where feature extraction and encryption are performed separately. Therefore the cloud server can extract the features from the encrypted Query image and matches with the features present in the cloud server. In [33], the authors proposed an image retrieval mechanism using improved BoVM model. The improved BoVM model is derived from Hamming embedding. The improved BoVM model can refine the matching based on visual words that provide binary signatures. This method also uses orthogonal transformation, where the features of the image are separated into two different components where the distance comparison and encryption are separately executed. Therefore the cloud server can match the features of the encrypted Query image with the encrypted images present in the cloud.

In this method [34], the authors proposed a multiple Image Owners with Privacy Protection (MIPP) based CBIR. Here the secure multi-party computation scheme is used to encrypt the image features. Different image owners can encrypt their images with their own keys. Therefore in image retrieval process, the retrieved images contain the images that are uploaded by different sources. This method does not provide a way to leak the image privacy of an image owner to other owners. In [35], the authors proposed a CBIR system for medical applications which is termed as PPDP (Privacy preserving disease prediction). The medical history of the parents are encrypted and outsourced to the cloud server. The outsourced encrypted data is used to train the prediction models by single layer perceptron learning algorithm.

In [36], the authors proposed a method to represent an image with hash codes. The hash code representation is the compressed representation of features extracted by deep convolution. It uses deep auto encoder to represent the features in compressed representation. Light weight encryption is used for encrypting the image before uploading to the cloud. The features are extracted using pre-trained convolutional neural network. A deep auto-encoder is used to transform the extracted features to binary compact codes. To retrieve the desired images Approximate nearest neighbor search is used to search the images in the dataset. In [37], the authors combine textual and visual features for a CBIR system. This method initially separates the query image features as non-textual and textual features. The image is classified as textual, if any text is present in the image and the image is classified as non-textual if there is no text present in the image. Visual salient features are extracted if the image is classified as non-textual. Bag of Textual words are extracted if the image is classified as textual. The fused feature vector is obtained by fusing the Bag of Textual words features and Visual salient features. The fused feature vector is used to retrieve the images present in the dataset.

In [38], the authors have proposed a bag-of-encrypted-words (BOEW) model for outsourced CBIR. The images to be outsourced are encrypted by color value substitution, block permutation and intra-block pixel permutation. The cloud server then estimates the local histograms from the blocks of the encrypted image. The cluster centers are estimated by clustering the local histograms. The BOEW model is created using the feature vector of the images. The image present in the cloud server is retrieved using the query image by measuring Manhattan distance. In [39], the authors proposed an robust image hashing technique in finding the related content in the encrypted images. This algorithm finds its applications in finding child sexual abuse material, terrorism related content and dangerous conspiracy material. This algorithm is robust to small modifications in the image, and can be operated without decipher the encrypted image. In [40], the authors proposed an method for extracting the features from the encrypted images by preserving the privacy of the outsourced images. In this method, the cloud server is assumed to be semi-honest. In [41], the authors proposed an CBIR system, that focus on distance threshold in classifying the images as retrieved image from the dataset. It uses a statistical scheme to estimate the number of features that intersect with the query images and candidate images. After extracting the features, Bhattacharyya distance (BD) is used to measure the dis-similarity score that varies between 0 and 1.

Instead of encrypting the images, the proposed scheme scrambles the images that are to be outsourced to the cloud. The rest of the research paper is being arranged into four sections. Section 2 shows the system model for the proposed work and Section 3 shows the proposed feature extraction algorithm from scrambled images for Content-based image retrieval. The performance of the proposed feature extraction algorithm is discussed in Section 4, and Section 5 depicts the conclusion of the paper.

2 Proposed Method

2.1 System Model

The system model of the proposed work contains three different entities such as image owner, cloud server and image user.

Image owner: Image owners are the people who outsource the original image. There may be multiple image owners. Consider an image owner contains n number of images represented by,

$$X = \{X_i\}_{i=1}^{n} = X_1, X_2, \ldots, X_n. \tag{1}$$

The images X are scrambled by the image owner using the key K and outsourced to the cloud server in scrambled form as,

$$Y = \{Y_i\}_{i=1}^{n} = Y_1, Y_2, \ldots, Y_n. \tag{2}$$

In this scheme, a single image owner is being considered. If there are many image owners, the image owners can scramble their image using their own secret keys. So that the users can search the images from all owners and only the authorized users can descramble the search results.

Cloud server: The cloud server receives the scrambled image Y from the image owner and extract the features using the Otsu's threshold based feature extraction algorithm. Let the features from n number of the scrambled images be,

$$F = F_1, F_2, \ldots, F_n. \tag{3}$$

This feature vector F is extracted from the scrambled images Y. When an image owner upload an i^{th} image whose feature is F_i, the image is grouped in a particular cluster. The indices are constructed with the features F on the cloud server along with the cluster number. When the search request in scrambled form is received from the user, the cloud server will extract the features from the scrambled query image and searches the nearest cluster center with similar features available in the index. The top k search results (scrambled images) from the cloud database will be returned to the user.

Image user: Image users are the authorized person to retrieve the images from the cloud server. The image user first scramble the query image using the key K_1 of the user and upload the scrambled query image to the cloud. The cloud server returns the top k best matched images to the image user. Using the secret key K shared by the image owner the retrieved scrambled images Y' is descrambled to view the retrieved content X'.

In the proposed scheme, the cloud provided is assumed to be untrustworthy. Therefore the proposed method prevents the cloud from learning the useful information present in the image dataset (ownerâĂŹs image) and query image (user image). This method does not discuss the behaviour of the CBIR service provider.

2.2 Architecture of Proposed Method

The block diagram of Otsu's threshold based feature extraction and instantaneous clustering for content-based image retrieval in cloud computing is depicted in Fig. (1). This method has three major entities such as Image owner, Cloud server and Image user. Consider that the image owner has n number of images represented as $X = \{X_i\}_{i=1}^{n} = \{X_1, X_2, \ldots, X_n\}$, the image owner scramble the image X using the key K to obtain the scrambled images Y using the position scrambling algorithm shown in Section 2.3, where $Y = \{Y_i\}_{i=1}^{n} = \{Y_1, Y_2, \ldots, Y_n\}$ represents the scrambled image. The image owner outsources the scrambled image to the cloud server where the features are extracted from the scrambled images using the proposed Otsu's threshold based feature extraction shown in Section 2.4. The features of n number of images are represented as, $F = \{F_1, F_2, \ldots, F_n\}$. Using the features F, the clustering is performed and the index is constructed. The clustering process is instantaneous, i.e. the image with the feature F_i will be moved to a particular cluster as soon as any image has been uploaded by the image owner. It does not require a complete set of features for the clustering process. The instantaneous clustering process is explained in Section 2.6. The index, cluster number and scrambled images (database) Y are stored in the cloud server. During the image retrieval process, the user scrambles the query image Q using the key K_1 to obtain the scrambled query image Q_E and uploads the scrambled query image Q_E to the cloud for image retrieval. When the CBIR system receives the scrambled query image Q_E, it extracts the OT-features from the scrambled query image Q_E using the Otsu's threshold based feature extraction shown in Section 2.4. Let the features extracted from the scrambled query image be F_Q . The CBIR system searches the best match clusters for F_Q using the cluster number and index number.

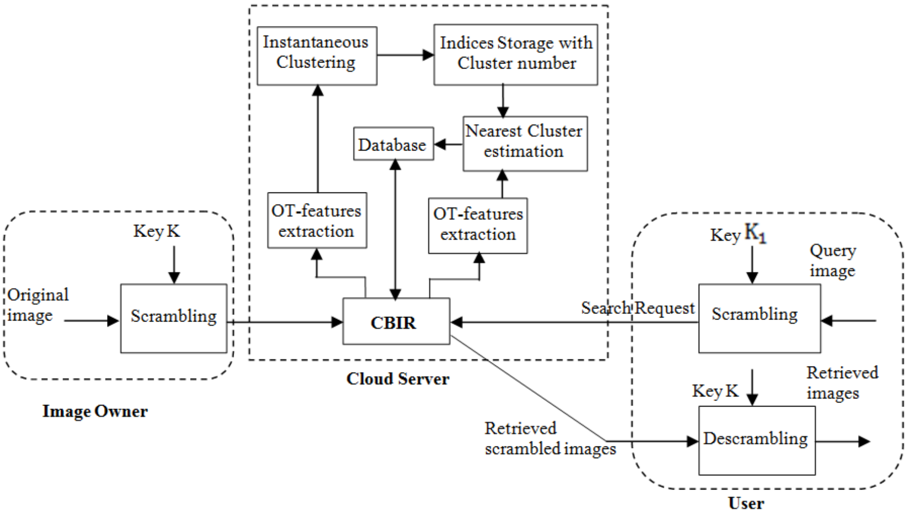


Figure 1. Block diagram of proposed Otsu's Threshold based feature extraction (OT-features) and Instantaneous Clustering for CBIR (IC-CBIR)

The scrambled database image corresponding to the first k nearest Euclidean distance images in the cluster will be returned to the user. Let the retrieved scrambled images be represented as, $Y' = \{Y_i'\}_{i=1}^{k}$ or $\{Y_1', Y_2', \ldots, Y_k'\}$. The user can descrambled the retrieved scrambled image Y' using the authorized key shared by the image owner as shown in Section 2.7. The top k matched retrieved descrambled image is represented as, $X' = \{X_i'\}_{i=1}^{k} = \{X_1', X_2', \ldots, X_k'\}$.

2.3 Scrambling

This scheme scrambles the image using position scrambling. Consider an original image I that contains Red, Green and Blue components as I_R, I_G and I_B respectively. Let the size of the image be $M \times N$. Using the key K, $M \times N$ pseudo-random sequence is generated. The component matrices I_R, I_G and I_B are converted in the form of the vector so that the length of I_R, I_G and I_B vector are $M \times N$. The position of the elements in the vector I_R, I_G and I_B are changed based on the pseudo-random sequence generated by the key K. The Key $K = \{K_1, K_2, K_3\}$ is a 48 bit binary number, where the first 16 bits forms key K_1 is used to scramble the rows of the image. The next 16 bits forms key K_2 which is used to scramble the columns of the image. The last 16 bits K_3 is used to scramble the complete sequence of the image. The 3 stage scrambling is applied in a sequential order, starting from row scrambling followed by column scrambling and finally complete scrambling. Let I_R^{rc}, I_G^{rc}, I_B^{rc} be the image that is scrambled using key K_1 and K_2. Let the pseudo-random sequence generated by key K_3 be $r = \{r_1, r_2, \ldots, r_{M\times N}\}$. The scrambled sequence for the I_R', I_G' and I_B' can be generated using the random sequence as,

$$I_R'(r) = I_R^{rc} \quad ; \quad I_G'(r) = I_G^{rc} \quad ; \quad I_B'(r) = I_B^{rc} \quad (4)$$

After scrambling, the vector is converted in the form of matrices having a size of $M \times N$ to obtain the scrambled image I' which has the components I_R', I_G' and I_B' respectively. Fig. (2) shows an example of the original image and the scrambled image scrambled by position scrambling.

The histogram of Red, Green and Blue components of the scrambled image shown in Fig. (2) is depicted in Fig. (3). The histogram remains unchanged before and after scrambling since the intensity of the pixel is unchanged.

2.4 Feature Extraction

The feature extraction algorithm extracts the features iteratively by calculating the difference of Otsu's threshold obtained during different iteration. Consider an scrambled image I' with Red, Green and Blue components as I_R', I_G' and I_B' respectively. Let the pixels present in the Red image I_R' component be represented as, $\{P_R(l)\}_{l=1}^{M\times N}$, where $M \times N$ is the size of the scrambled Red image component I_R'. Initially, the entire pixels present in the image I_R' is considered as a class C_R, i.e.

$$C_R = \{P_R(l)\}_{l=1}^{M\times N} = \{P_R(1), P_R(2), \ldots , P_R(M \times N)\}. \quad (5)$$

Estimate the Otsu's threshold of the class C_R. Let the Otsu's threshold be T_R as shown in Fig. (4)(a) (intensity corresponding to red line). During the first iteration, group the class C_R into two classes C_{11} and C_{21}. C_{11} contains the pixels whose intensity is less than the initial Otsu's threshold T_R, i.e.

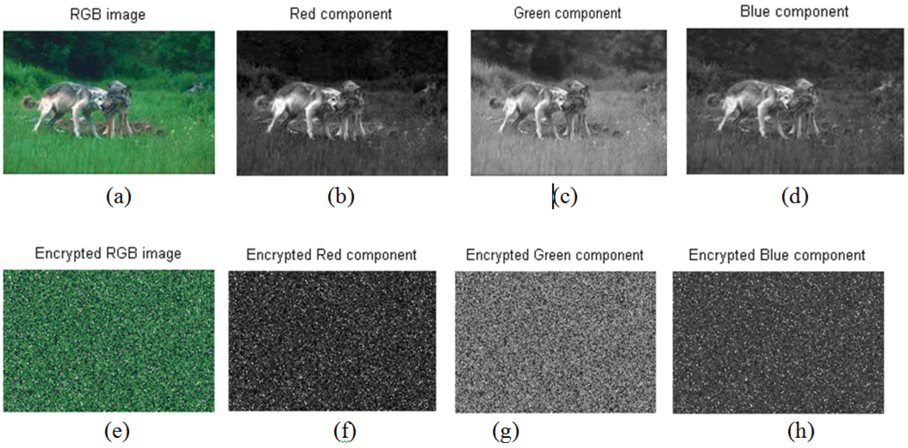


Figure 2. Example of Original images and its scrambled form

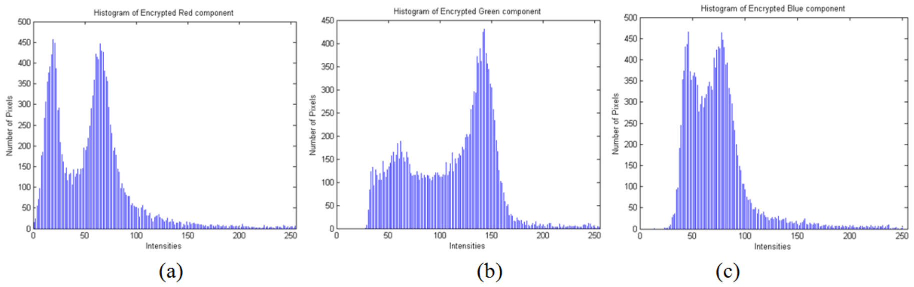


Figure 3. Histogram of Red, Green and Blue component of Scrambled image

$$C_{11} = \{P_{11}(l)\}_{l=1}^{U_{11}} = \{P_{11}(1), P_{11}(2), \ldots, P_{11}(U_{11})\} \quad (6)$$

where $0 \leq P_{11}(l) \leq T_R$ and U_{11} is the number of pixel whose intensity is less than or equal to the initial threshold T_R. Similarly, the class C_{21} contains the pixels whose intensity is greater than the threshold T_R.

$$C_{21} = \{P_{21}(l)\}_{l=1}^{U_{21}} = \{P_{21}(1), P_{21}(2), \ldots, P_{21}(U_{21})\} \quad (7)$$

$$T_R + 1 \leq P_{21}(l) \leq 255 \quad (8)$$

The initial class C_R and first iteration classes C_{11} and C_{21} is related as,

$$C_R = C_{11} \cup C_{21} \quad (9)$$

where,∪ represents the union operator. Estimate the Otsu's threshold for the two classes C_{11} and C_{21}. Let the Otsu's threshold be T_{11} and T_{21} calculated from the class C_{11} and C_{21} respectively as shown in Fig. (4)(b) (intensity corresponding to green lines). At the end of the first iteration, the feature of the image I'_R is calculated as,

$$F_{1R} = \{T_R - T_{11}, T_{21} - T_R\}. \quad (10)$$

During the second iteration, the class C_{11} is divided into two classes C_{12} and C_{22}. Similarly, the class C_{21} is divided into two classes C_{32} and C_{42}. The class C_{12} contains the pixels whose intensity is less than the first iteration threshold T_{11}.

$$C_{12} = \{P_{12}(l)\}_{l=1}^{U_{12}} = \{P_{12}(1), P_{12}(2), P_{12}(3), \ldots , P_{12}(U_{12})\} \quad (11)$$

where $0 \leq P_{12}(l) \leq T_{11}$ and U_{12} is the number of the pixel whose intensity is less than the threshold T_{11}. The class C_{22} contains the pixels whose intensity is between $T_{11} + 1$ and T_R.

$$C_{22} = \{P_{22}(l)\}_{l=1}^{U_{22}} = \{P_{22}(1), P_{22}(2), P_{22}(3), \ldots , P_{22}(U_{22})\} \quad (12)$$

$T_{11}+1 \leq P_{22}(l) \leq T_R$ and U_{22} is the number of pixels whose intensity lies between $T_{11}+1$ and T_R. Similarly, the class C_{32} contains the pixels whose intensity lies between $T_R + 1$ and T_{21}.

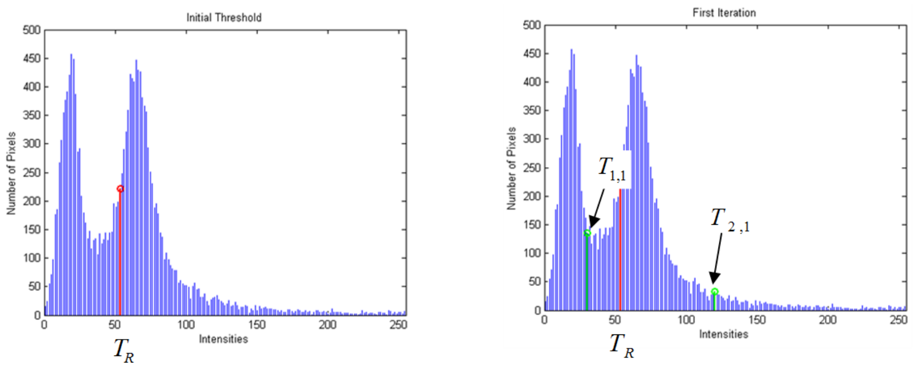
$$C_{32} = \{P_{32}(l)\}_{l=1}^{U_{32}} \quad (13)$$

$$C_{32} = \{P_{32}(1), P_{32}(2), P_{32}(3), \ldots, P_{32}(U_{32})\} \quad (14)$$

$T_R + 1 \leq P_{32}(l) \leq T_{21}$, where U_{32} is the number of pixels whose intensity lies between $T_R + 1$ and T_{21}. The class C_{42} contains the pixels whose intensity greater than $T_{21} + 1$.

$$C_{42} = \{P_{42}(l)\}_{l=1}^{U_{42}} \quad (15)$$

$$C_{42} = \{P_{42}(1), P_{42}(2), P_{42}(3), \ldots, P_{42}(U_{42})\} \quad (16)$$



(a) (b)

Figure 4. Otsu's Threshold (a) Initial Threshold (b) First iteration

$T_{21}+1 \leq P_{42}(l) \leq 255$, where U_{42} is the number of pixels whose intensity is greater than $T_{21}+1$. At the second iteration, estimate the threshold for the four different classes C_{12}, C_{22}, C_{32} and C_{42}. The thresholds of the classes C_{12}, C_{22}, C_{32} and C_{42} are represented as T_{12}, T_{22}, T_{32} and T_{42} respectively as shown in Fig. (5)(a) (intensity corresponding to yellow lines). The feature of the image I'_R is calculated at the end of the second iteration as,

$$F_{2R} = \{T_{11}-T_{12}, T_{22}-T_{11}, T_R-T_{22}, T_{32}-T_R, T_{21}-T_{32}, T_{42}-T_{21}\}. \quad (17)$$

The same process is repeated for the L number of iteration, to obtain the features F, Fig. (5)(b) shows the Otsu's threshold estimated in the third iteration (represented as magenta lines).

$$F_R = \{F_{1R}, F_{2R}, F_{3R}, \ldots, F_{LR}\} \quad (18)$$

$$F_R = \{T_R-T_{11}, T_{21}-T_R, T_{11}-T_{12}, T_{22}-T_{11}, T_R-T_{22}, T_{32}-T_R, T_{21}-T_{32}, T_{42}-T_{21}, \ldots\} \quad (19)$$

The number of features present in F_R is $N_R = \sum_{j=1}^{L} 2^j$. The features extracted from the scrambled Red component I'_R is F_R. Similarly, the features are extracted from the scrambled Green and Blue components I'_G and I'_B is F_G and F_B. Therefore the complete set of features from the scrambled image I' is

$$F = [F_R \quad F_G \quad F_B]. \quad (20)$$

The total number of features from the scrambled image I' be

$$N_F = 3 \times N_R = 3 \times \sum_{j=1}^{L} 2^j. \quad (21)$$

The feature F_R, F_G and F_B gives the spacing between the thresholds obtained at different iterations for Red, Green and Blue histogram. Fig. (6) shows the classification diagram which represents the features and threshold estimated at different levels.

2.5 Otsu's Threshold Estimation

The Otsu's threshold is estimated as follows.

Step 1: Estimate the normalized histogram from the class of pixels. Let the i^{th} class of j^{th} iteration is represented as

$$C_{ij} = \{P_{ij}(l)\}_{l=1}^{U_{ij}} \quad (22)$$

U_{ij} is the number of pixels present in the i^{th} class of j^{th} iteration. Let the minimum intensity value present in the class C_{ij} be t_{min}, and maximum intensity present in the class C_{ij} be t_{max}. Therefore the component of the histogram is denoted as, h_u where $u \in [t_{min}, t_{max}]$.

Step 2: Estimate the cumulative sums $S(v)$ for $v \in [t_{min}, t_{max}]$,

$$S(v) = \sum_{u=t_{min}}^{v} h_u \quad (23)$$

Step 3: Estimate the cumulative means $\mu(v)$ for $v \in [t_{min}, t_{max}]$,

$$\mu(v) = \sum_{j=t_{min}}^{v} jh_j \quad (24)$$

Step 4: Estimate the global intensity mean

$$\mu_G = \sum_{u=t_{min}}^{v} uh_u \quad (25)$$

Step 5: Estimate the between-class variance $\sigma^2(v)$, for $v \in [t_{min}, t_{max}]$

$$\sigma^2(v) = \left[\frac{\mu_G S(v) - \mu(v)}{S(v)[1-S(v)]}\right] \quad (26)$$

Step 6: Estimate the Otsu's threshold v', as the value of v for which the variance $\sigma^2(v)$ is maximum.

$$\sigma^2(v') = \max_{t_{min} \leq v \leq t_{max}} \sigma^2(v) \quad (27)$$

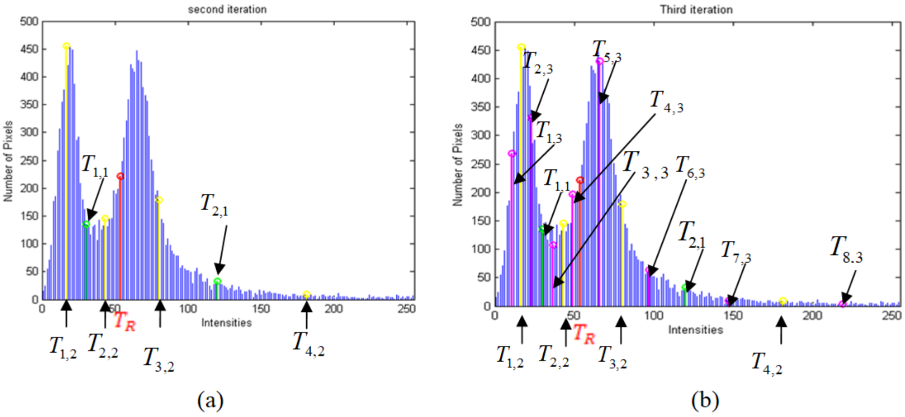


Figure 5. Otsu's threshold (a) Second Iteration (b) Third Iteration

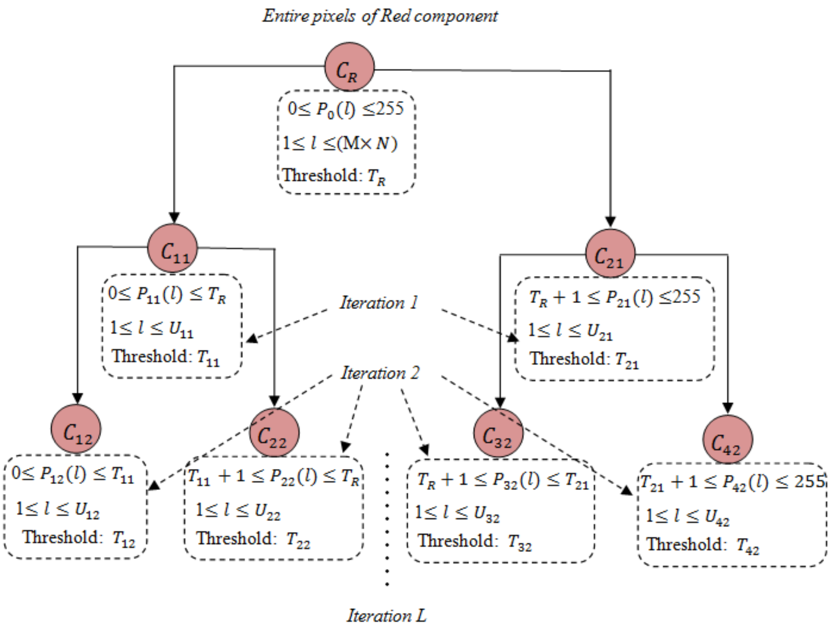


Figure 6. Classification diagram of the proposed feature extraction

If the maximum is not unique, find v' by averaging the values v corresponding to the different maxima obtained.

2.6 Instantaneous Clustering and Content Based Image Retrieval (IC-CBIR)

Let F_i be the feature extracted from an i^{th} image. Let Δ be the parameter that represents the cluster threshold distance. When an image owner uploads the i^{th} image having the features F_i it will be immediately moved in any one cluster. If there is no cluster available, it will be moved to the first cluster. If there is already only one cluster having a Centroid Cl_1, then the Euclidean distance between the Centroid Cl_1 and the feature F_i is estimated as $d_{i,1}$. If $d_{i,1} > \Delta$, the image will be moved into the new cluster having the Centroid $Cl_2 = F_i$. If $d_{i,1} < \Delta$, then the image will be moved in the same cluster whose Centroid is C_1. If there is more than one cluster already available whose Centroid is represented by $\{Cl_1, Cl_2, \ldots, Cl_j\}$. The Euclidean distance between the i^{th} image feature and clusters Centroid are estimated as $\{d_{i,1}, d_{i,2}, \ldots, d_{i,j}\}$, the minimum distance between the feature F_i and cluster Centroid $\{Cl_1, Cl_2, \ldots, Cl_j\}$ is estimated by

$$d_n = min\{d_{i,1}, d_{i,2}, \ldots, d_{i,j}\} \qquad (28)$$

where d_n represents the nearest cluster center from the feature space F_i. If $d_n < \Delta$, then the image corresponds to the feature F_i belongs to the n^{th} cluster. If $d_n > \Delta$, then the image corresponds to the new cluster center $Cl_{j+1} = F_i$. An image is uploaded in the cloud by the image owner; it is instantaneously clustered in any one of the clusters. Therefore while retrieving an image, the feature of the query image F_Q is compared with the cluster centers $\{Cl_1, Cl_2, \ldots, Cl_N\}$. The Euclidean distance between the features F_Q and cluster centers $\{Cl_1, Cl_2, \ldots, Cl_N\}$ is represented by $\{d_{Q,1}, d_{Q,2}, \ldots, d_{Q,N}\}$. If the user request for top k images using the query image whose feature is F_Q, then the cluster that has minimum distance with F_Q is chosen and k images nearest to feature space F_Q are retrieved. If the selected clusters have exactly k number of images, then all the images present in the cluster are retrieved. If the selected cluster has more than k images, then k images that are closer to the Centroid of the cluster are retrieved back. If the number of images in the selected cluster is less than k, then k number of images is retrieved from the selected cluster and nearby cluster. The best match features present in the selected cluster or nearby cluster of the selected cluster are chosen based on the minimum Euclidean distance. The Euclidean distance between the scrambled query image features F_Q and the feature F_j is estimated using Eq. (29).

$$D_i = \sqrt{\sum_{k=1}^{3N_f} (F_Q - F_j(k))^2} \tag{29}$$

where $0 \leq j \leq L_c$ and $3N_f$ is the number of features extracted from an scrambled image and L_c is the number of feature vectors present in the selected clusters. The Euclidean distance D_i is arranged in ascending order to obtain the top k retrieved scrambled images. The retrieved scrambled images are descrambled using the key K to obtain the actual retrieved images.

2.7 Descrambling

The descrambling of the retrieved scrambled image is portrayed as follows. Consider an scrambled image I' that contains Red, Green and Blue components as I'_R, I'_G and I'_B respectively. The pseudo-random sequence $M \times N$ is generated using the key K_3. The length of I'_R, I'_G and I'_B vector be $M \times N$. The shuffled position of the elements in the vector I'_R, I'_G and I'_B are reconstructed using the pseudo-random sequence generated by the key K_3, that is used for scrambling. The key $K = \{K_1, K_2, K_3\}$ is of 48 bit where the last 16 bits represent key K_3. Let the pseudo-random sequence generated by key K_3 be $r = \{r_1, r_2, \ldots, r_{M\times N}\}$. The descrambled image for the Red, Green and Blue components I_R^{rc} $I_G^{rc}$and I_B^{rc} is generated using the equation as,

$$I_R^{rc} = I'_R(r) \quad ; \quad I_G^{rc} = I'_G(r) \quad ; \quad I_B^{rc} = I'_B(r) \tag{30}$$

After descrambling using the key K_3, the columns are descrambled using the random sequence generated using the key K_2. Finally the rows are descrambled using the random sequence generated using the key K_1. After three stage de-scrambling, the descrambled image I has the Red, Green and Blue components I_R, I_G and I_B respectively.

3 Experimental Evaluation

The experimental evaluation of the proposed feature extraction algorithm is done on four different datasets such as the Corel 10K image dataset [42], Misc dataset [43] Oxford flower dataset [44] and INRIA holidays dataset [45]. The Corel 10K image dataset contains hundred categories of images and each category contains a hundred similar images. The Misc dataset contains 10000 images and the Oxford flower dataset contains 1360 images with different categories. The INRIA Holidays dataset contains 1491 images of personal holidays photos. To measure the performance of the proposed feature extraction 250 test images from the Corel 10K dataset, 250 images from the Misc dataset, 34 images from the Oxford flower dataset and 150 images from INRIA Holidays dataset are randomly selected. Fig. (7) shows sample test images from the four different datasets. The proposed algorithm is implemented using MATLAB. The performance of the proposed method is evaluated using metrics such as retrieval precision and efficiency. The efficiency is measured in terms of Time of Query search, Time of Index construction, Time of feature extraction and storage consumption of index.

3.1 Retrieval Precision

The precision for a query image is defined as $P_k = \frac{k_1}{k}$ where k_1 is the number of the similar images retrieved out of k images. The average precision is estimated from the precision of all query images for each dataset with different values of L, where L represent the number of iteration performed for feature extraction. Fig. (8) shows the top 40 search results for $L = 5$ the query image shown in Fig. (7)(e) from the Corel 10K image dataset. Fig. (9) shows the Instantaneous clustering for five different categories of images that contain the first 500 images from the dataset. The precision is tested for different values of L. The precision is found to be maximum for $L = 5$. Precision decreases as the value of k increases. Table 1 shows the precision comparison for four different datasets with different values of k and it is plotted as shown in Fig. (10). The precision differs for all the four dataset. The Precision is high for INRIA Holiday dataset when compared to Corel 10K, MISC and flower dataset for all values of k. We have compared the precision of the proposed method with the traditional methods such as BoEW [38], PPP-CBIR [29], PP-CBIR [32], PP-CD [30]. The precision thus estimated for $L = 5$ and $L = 6$ founds to be greater than the traditional feature descriptors such as CSD, CLD, EHD and SCD presented in [30]. The average precision for $L = 5$ is around 62.2175% and 49.017% for $k = 20$ and $k = 100$ respectively. The comparison of

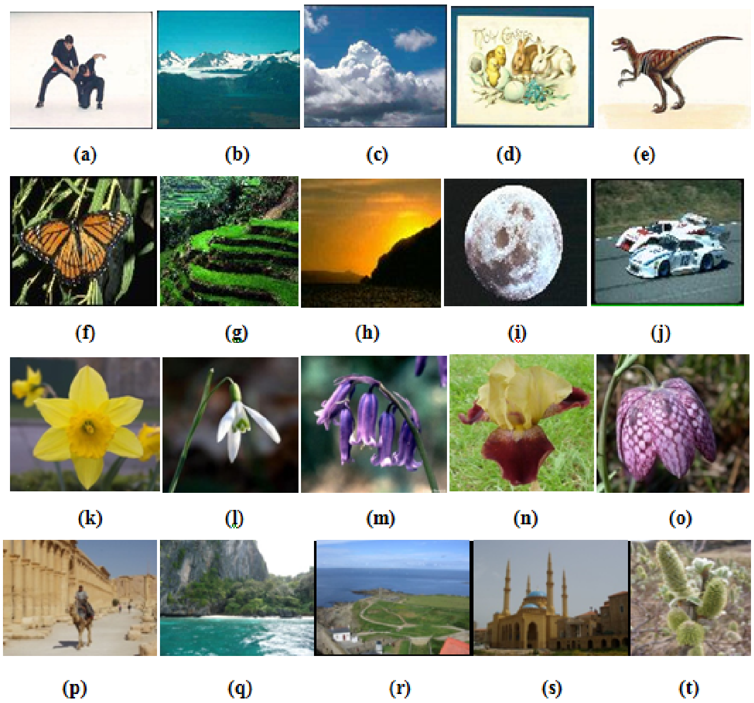


Figure 7. Sample Test Images from different dataset (a)-(e) Corel 10K dataset (f)-(j) misc dataset (k)-(o) Oxford flower dataset (p)-(t) INRIA Holidays dataset

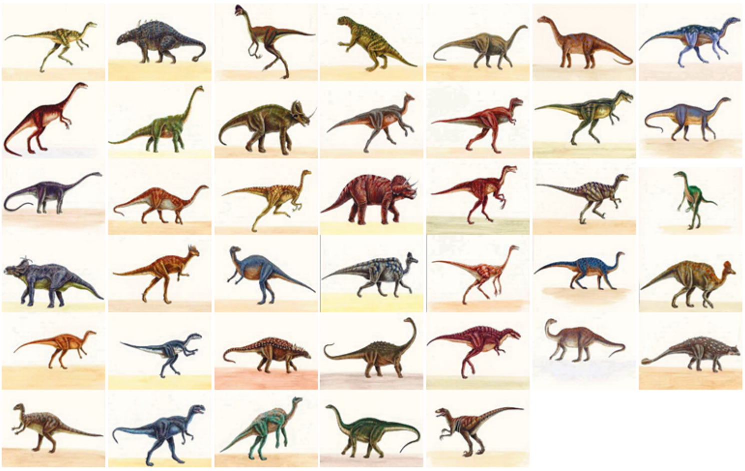


Figure 8. Top 40 retrieved images for the Query image shown in Fig. (7)(e)

average precision between proposed and CSD, CLD, EHD and SCD descriptor from [30] is shown in Fig. (11) and Table 2. Table 3 shows the average precision comparison for proposed and traditional methods such as BoEW [38], PPP-CBIR [29], PP-CBIR [32], PP-CD [30] . The precision for the proposed method was 60.2%, 56.2%, 61.2% and 71.27% for the dataset Corel 10K, Misc, Oxford flower, INRIA Holidays respectively. The presion of the proposed method was higher than the tradiional methods and its graphical comparision is shown in Fig. (12). The precision estimated for $L = 5$ was found to be higher than the traditional methods. For the INRIA dataset, the precision was 71.27% which is higher than BoEW and PP-CBIR method. For the Corel 10K dataset, the precision was 60.2% which is higher than PPP-CBIR, PP-CBIR and PP-CD methods.

3.2 Time of Index Construction

The time of index construction depends on the time of building the index table. The time complexity of index construction is $O(nL)$. The time consumption of index construction depends on the number of iteration L and the number of images in the database n. The time consumption of index construction for proposed

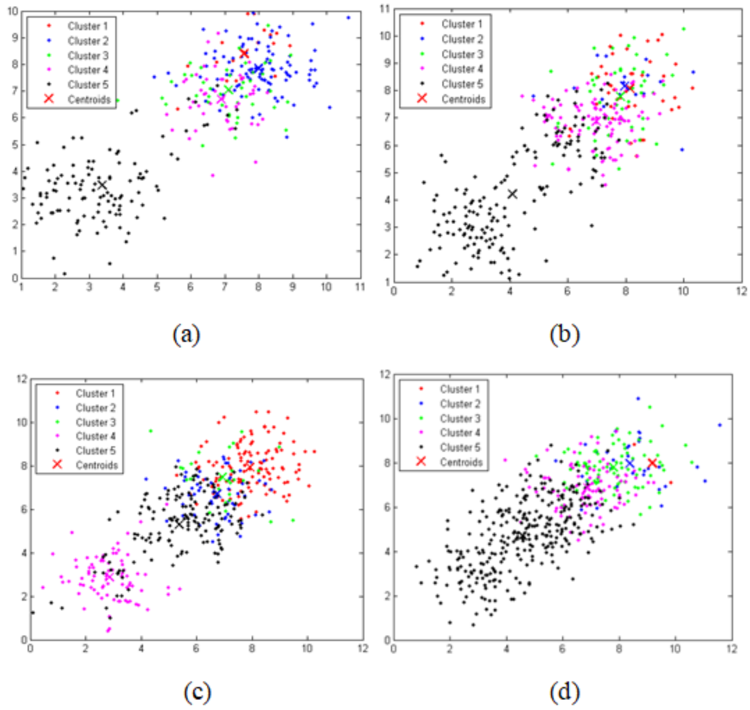


Figure 9. Instantaneous clustering for five different categories with $\Delta = 0.3$ (a) Corel 10K dataset (b) Misc dataset (c) Oxford flower dataset (d) INRIA Holiday dataset

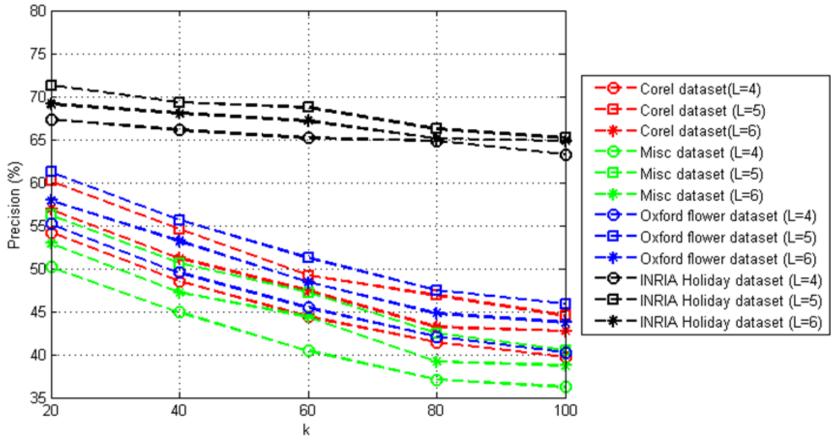


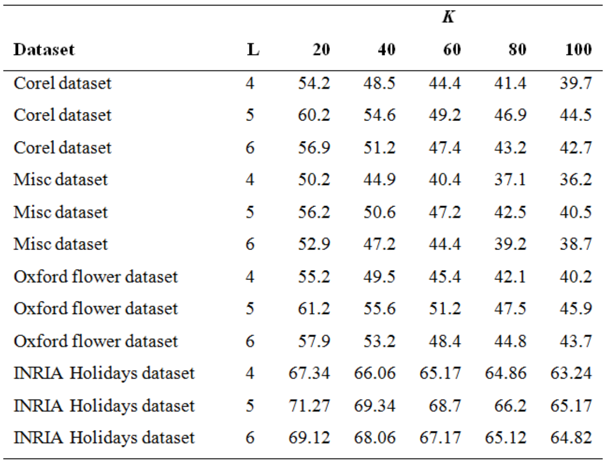
Figure 10. Comparison of precision for the proposed method from different datasets

and traditional feature extraction is shown in Fig. (13). The average time of index construction depends on the number of image in the dataset, and it is around 500 ms for the database having 10,000 images (average time of index construction is calculated using Corel 10K image dataset and Misc dataset). As the value of L increases, the time of index construction also increases then the number of features is also getting increased. While comparing the average time of index construction for the proposed and traditional methods, the time of index construction is less for $L = 4$ and $L = 5$, than the traditional methods. For $L = 6$, the time of index construction is greater than the traditional EHD descriptor in [30] . Table 4 shows the average time consumption for index construction comparison for proposed and traditional methods such as BoEW [38], PPP-CBIR [29], PP-CBIR [32], PP-CD [30] . The time of index construction for the proposed method with $L = 4$ and $L = 5$ is estimated as 471ms and 780ms respectively which is less than the traditional methods. Fig. (14) shows the graphical comparison of average time consumption for index construction.

3.3 Time of Search

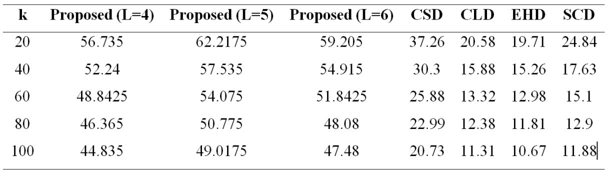
As soon as the proposed CBIR server receives the scrambled query image, it extracts the features from

Table 1. Precision comparison for four different datasets with different values of k



		K				
Dataset	**L**	**20**	**40**	**60**	**80**	**100**
Corel dataset	4	54.2	48.5	44.4	41.4	39.7
Corel dataset	5	60.2	54.6	49.2	46.9	44.5
Corel dataset	6	56.9	51.2	47.4	43.2	42.7
Misc dataset	4	50.2	44.9	40.4	37.1	36.2
Misc dataset	5	56.2	50.6	47.2	42.5	40.5
Misc dataset	6	52.9	47.2	44.4	39.2	38.7
Oxford flower dataset	4	55.2	49.5	45.4	42.1	40.2
Oxford flower dataset	5	61.2	55.6	51.2	47.5	45.9
Oxford flower dataset	6	57.9	53.2	48.4	44.8	43.7
INRIA Holidays dataset	4	67.34	66.06	65.17	64.86	63.24
INRIA Holidays dataset	5	71.27	69.34	68.7	66.2	65.17
INRIA Holidays dataset	6	69.12	68.06	67.17	65.12	64.82

Table 2. Average precision comparison for the proposed method with descriptor CSD, CLD, EHD and SCD in [30]



k	Proposed (L=4)	Proposed (L=5)	Proposed (L=6)	CSD	CLD	EHD	SCD
20	56.735	62.2175	59.205	37.26	20.58	19.71	24.84
40	52.24	57.535	54.915	30.3	15.88	15.26	17.63
60	48.8425	54.075	51.8425	25.88	13.32	12.98	15.1
80	46.365	50.775	48.08	22.99	12.38	11.81	12.9
100	44.835	49.0175	47.48	20.73	11.31	10.67	11.88

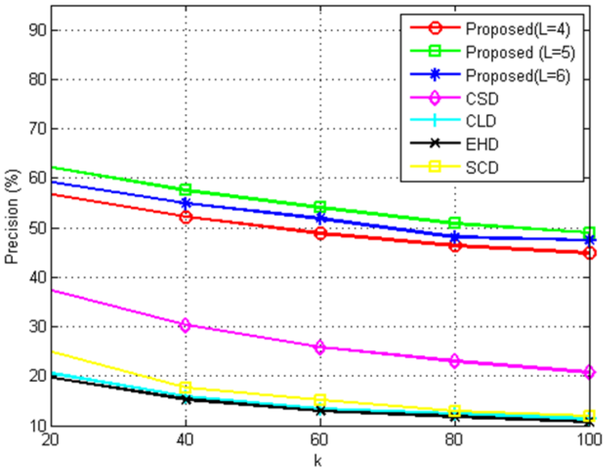
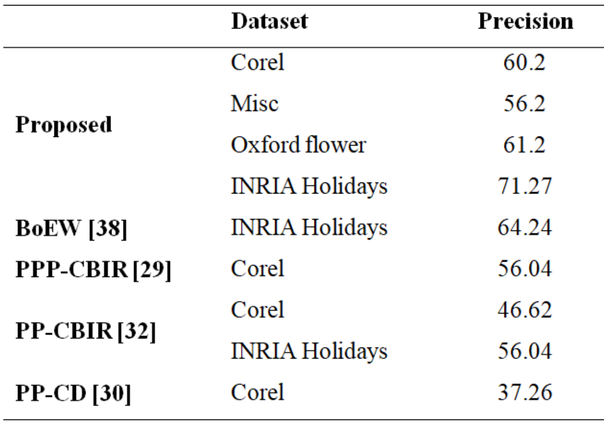


Figure 11. Average Retrieval Precision comparison of proposed method with the descriptors CSD, CLD, EHD and SCD

the scrambled query image. The features extracted from the scrambled query image are being matched with the nearest cluster center. The scrambled images corresponding to the top k match of the selected cluster is returned to the user. The time consumption for performing the search operation increases as the size of the database increases. The search time of the proposed CBIR is less when compared to the traditional CBIR system because it produces only less number of features from an image as shown in Table 8. The time consumption of search depends on finding the top k minimum Euclidean distance. The time consumption of search depends on the number of iteration L and the number of images outsourced to the database n. As the value of L increases the time consumption of search increases. Fig. (15) depicts the comparison of the average search time of the

Table 3. Average precision comparison for the proposed method with traditional methods



	Dataset	Precision
Proposed	Corel	60.2
	Misc	56.2
	Oxford flower	61.2
	INRIA Holidays	71.27
BoEW [38]	INRIA Holidays	64.24
PPP-CBIR [29]	Corel	56.04
PP-CBIR [32]	Corel	46.62
	INRIA Holidays	56.04
PP-CD [30]	Corel	37.26

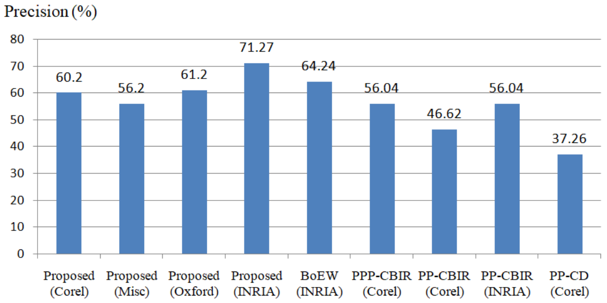


Figure 12. Average Retrieval Precision comparison of proposed method with the traditional methods

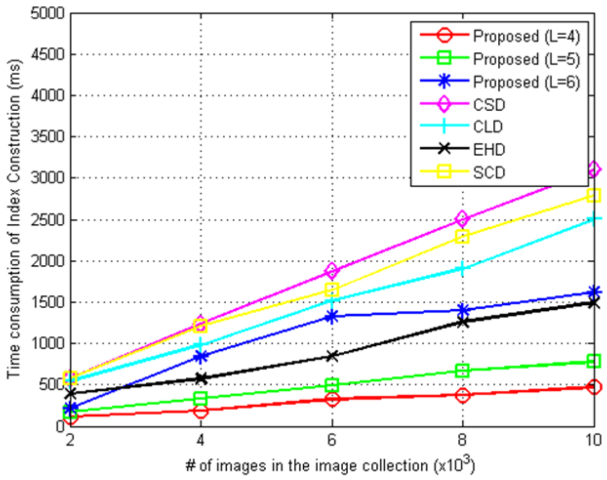


Figure 13. Average Time consumption for Index Construction comparison of the proposed method with CSD, CLD, EHD and SCD descriptor of [30]

Table 4. Average time consumption for index construction

Method	Proposed (L=4)	Proposed (L=5)	Proposed (L=6)	BOEW	PPP-CBIR	PP-CBIR	PP-CD
Time (ms)	471	780	1618	5390	950	1478	3000

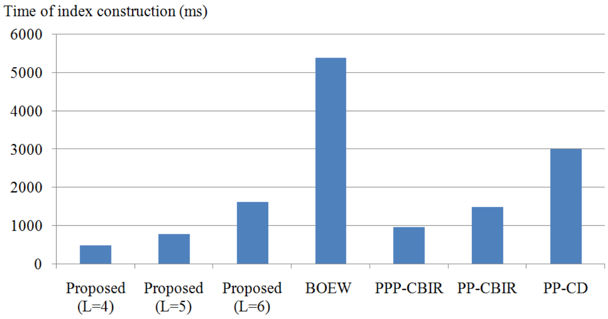


Figure 14. Comparison of Average time consumption for index construction

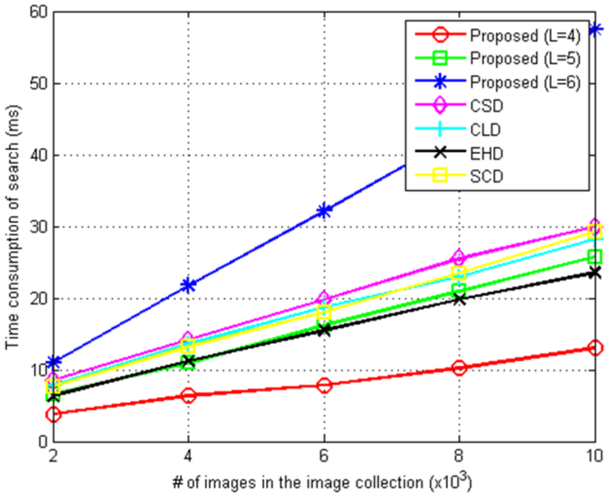


Figure 15. Average search time comparison of proposed and traditional methods

proposed method and traditional descriptors such as CSD, CLD, EHD and SCD shown in [30]. The search time is less for $L = 4$. For a database having 10,000 images, the average search time is around 13ms (average search time is calculated using Corel 10K image dataset, Misc dataset, Oxford flower dataset and INRIA Holidays dataset). The search time $L = 5$ is slightly greater than the EHD descriptor which is around 26ms. Table 5 shows the time of search comparison for proposed method with traditional method such as BoEW [38], PPP-CBIR [29] , PP-CBIR [32], PP-CD [30]. The proposed method has a less time of search than the existing methods for $L = 4$ and $L = 5$ which is estimated as 13ms and 25.75ms respectively. The graphical comparison is depicted in Fig. (16).

Table 5. Average time of search comparison for proposed and traditional methods

Method	Proposed (L=4)	Proposed (L=5)	Proposed (L=6)	BOEW	PPP-CBIR	PP-CBIR	PP-CD
Time (ms)	13	25.75	57.45	70.4	44	28.3	30

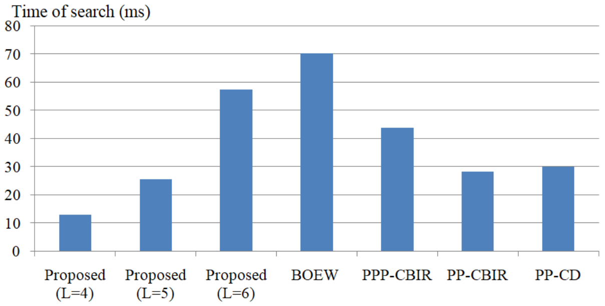


Figure 16. Average time of search comparison for proposed and traditional methods

3.4 Time of Feature Extraction

The time of feature extraction depends on the number of iterations L and the number of clusters formed during instantaneous clustering. As the iteration increases, the number of features almost doubles. The number of features generated for different values of L is depicted in Table 8. The time of feature extraction for different values of L for different dataset size is shown in Table 6. For $L = 4$, with the dataset size 10,000, the average time consumption of feature extraction is 1986 s while it is 5568 s for $L = 5$.

Table 6. Average Time consumption of feature extraction from Corel 10K and Misc dataset

Number of	Number of images in the Dataset				
Iteration	2000	4000	6000	8000	10000
$L = 4$	458 s	899 s	1349 s	1713 s	1986 s
$L = 5$	639 s	1079 s	1665 s	2509 s	2790 s
$L = 6$	1233 s	2569 s	3264 s	4292 s	5568 s

3.5 Storage Consumption of Index

The storage consumption of the index depends on the number of iteration L and the number of images outsourced to the cloud n. The number of features extracted in iteration is shown in Table 6. Table 7 shows the average storage consumption of index for different values of L for different dataset size. For a database having a size of 10,000 images, the storage consumption is around 700 KB for $L = 4$ and it is about 3000 KB for $L = 6$. The proposed method produces less number of features during feature extraction. For $L = 5$, the proposed method produces

Table 7. Average Storage consumption of index for different size of Dataset

Number of	Number of images in the Dataset				
Iteration	2000	4000	6000	8000	10000
$L = 4$	140 KB	286 KB	428 KB	569 KB	721 KB
$L = 5$	264 KB	531 KB	786 KB	1049 KB	1331 KB
$L = 6$	628 KB	1232 KB	1795 KB	2396 KB	3025 KB

Table 8. Number of features extracted for different values of L

Iteration	Number of features from a single image ($3 \times N_f$)
$L = 2$	18
$L = 3$	42
$L = 4$	90
$L = 5$	186
$L = 6$	378
$L = 7$	762

only 186 features from a single image. This is very less when compared to feature extraction algorithms used in other CBIR systems.

3.6 Security Analysis

The scrambling algorithm here uses a 48 bit binary key, where the first 16 bit is used to generate the random sequence to scramble the rows. After scrambling the rows, the next 16 bits are used to generate the random sequence to scramble the columns. Finally using the remaining 16 bits, a random sequence is generated to scramble a channel of the image. The rows of the channel in an image can be scrambled in 2^{16} ways. Similarly the columns of the channel in an image can be scrambled in 2^{16} ways. The complete row and column scrambled pixels can be scrambled in 2^{16} ways. Therefore the number of combinations in which the pixels can be scrambled is 2^{48}. In order to check the robustness of the algorithm to attacks, we have used the attacks such as JPEG compression, brightening and darkening on scrambled Query image. Fig. (17) shows the descrambled images after applying attacks to the scrambled Query image. Even though the scrambled images are subjected to attacks such as JPEG compression, brightening and darkening, there is only slight reduction in precision. This shows that the retrieval is possible if the scrambled query image is subjected to attack. The image user only needs to have the exact key shared by the image owner. Table 9 shows the Precision changes when the scrambled Query image is subjected to attacks such as JPEG compression, Brightening and darkening.

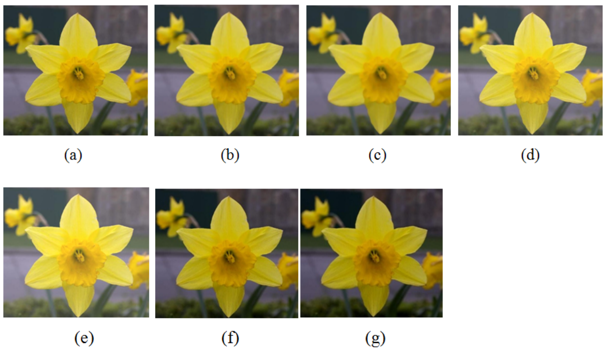


Figure 17. The distortion of image under different attack (a) Original Query image (b) JPEG compression 70% (c) JPEG compression 90% (d) Brightening by 20 (e) Brightening by 40 (f) Darkening by 20 (g) Darkening by 40

Table 9. Precision with Scrambled images which are subjected to attacks

Attack	Corel dataset	Misc dataset	Oxford flower dataset	INRIA Holidays dataset
No attack	60.2	56.2	61.2	71.27
JPEG compression 70%	58.3	54.3	59.57	69.37
JPEG compression 90%	57.4	53.4	58.16	68.47
Brightening by 20	59.4	55.4	59.82	70.47
Brightening by 40	58.6	54.6	59.04	69.67
Darkening by 20	59.2	55.2	59.82	70.27
Darkening by 40	58.32	54.32	58.66	69.39

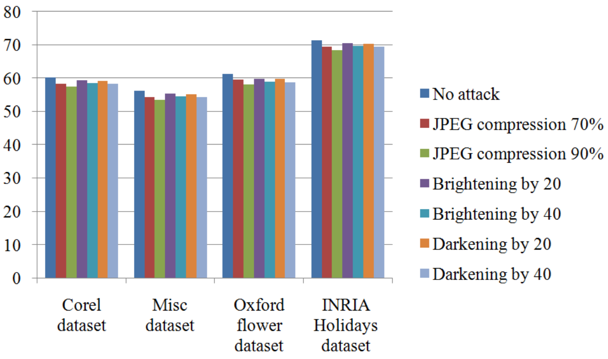


Figure 18. Precision comparison with scrambled images which are subjected to attacks

Fig. (18) shows the graphical comparison for precision when the scrambled images are subjected to attacks. The proposed method shows a good performance for $L = 5$. The next section shows the conclusion of the proposed work.

4 Conclusion

This paper proposed a feature extraction algorithm on scrambled images using Otsu's threshold and instantaneous clustering for content-based image retrieval on cloud computing. The image owner scrambles the image using position scrambling. The cloud server extracts the OT-features from the scrambled image iteratively by estimating the Otsu's threshold. The iteration is repeated L times to extract the features. During the image retrieval, the cloud server receives the scrambled image which is scrambled by the user and extracts the features using the Otsu's threshold. This paper also proposed an instantaneous clustering (IC-CBIR) approach which moves the image into a cluster as soon as the image was uploaded by its owner. The experimental verification is performed on datasets such as Corel 10K, Misc, Oxford flower and INRIA holidays dataset using randomly selected test query images. The proposed method outperforms the traditional feature extraction descriptors CSD, CLD, EHD and SCD shown in [30] and other traditional methods such as BoEW [38], PPP-CBIR [29], PP-CBIR [32], PP-CD [30] in terms of precision and efficiency such as time of search and time of index construction. While comparing the precision and efficiency for $L = 5$ it shows a better performance than the traditional methods.

References

[1] John Eakins and Margaret Graham. Content-based image retrieval. 1999.

[2] Ishwar K Sethi, Ioana L Coman, and Daniela Stan. Mining association rules between low-level image features and high-level concepts. In *Data Mining and Knowledge Discovery: Theory, Tools, and Technology III*, volume 4384, pages 279–290. International Society for Optics and Photonics, 2001.

[3] Shi-Kuo Chang and Shao-Hung Liu. Picture indexing and abstraction techniques for pictorial databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):475–484, 1984.

[4] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic, and William Equitz. Efficient and effective querying by image content. *Journal of intelligent information systems*, 3(3-4):231–262, 1994.

[5] Alex Pentland, Rosalind W Picard, and Stan Sclaroff. Photobook: Content-based manipulation of image databases. *International journal of computer vision*, 18(3):233–254, 1996.

[6] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Communications of the ACM*, 40(5):70–79, 1997.

[7] John R Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98, 1997.

[8] Fuhui Long, Hongjiang Zhang, and David Dagan Feng. Fundamentals of content-based image retrieval. In *Multimedia information retrieval and management*, pages 1–26. Springer, 2003.

[9] Quist-Aphetsi Kester. Image encryption based on the rgb pixel transposition and shuffling. *International Journal of Computer Network and Information Security*, 5(7):43, 2013.

[10] Musheer Ahmad and M Shamsher Alam. A new algorithm of encryption and decryption of images using chaotic mapping. *International Journal on computer science and engineering*, 2(1):46–50, 2009.

[11] Peng Yanguo, Cui Jiangtao, Peng Changgen, and Ying Zuobin. Certificateless public key encryption with keyword search. *China Communications*, 11(11):100–113, 2014.

[12] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*, pages 44–55. IEEE, 2000.

[13] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*, pages 44–55. IEEE, 2000.

[14] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. *Journal of Computer Security*, 19(5):895–934, 2011.

[15] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *International symposium on privacy enhancing technologies symposium*, pages 235–253. Springer, 2009.

[16] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999.

[17] Chao-Yung Hsu, Chun-Shien Lu, and Soo-Chang Pei. Secure and robust sift. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 637–640, 2009.

[18] Janez Križaj, Vitomir Štruc, and Nikola Pavešić. Adaptation of sift features for robust face recognition. In *International Conference Image Analysis and Recognition*, pages 394–404. Springer, 2010.

[19] Unsang Park, Sharath Pankanti, and Anil K Jain. Fingerprint verification using sift features. In *Biometric Technology for Human Identification V*, volume 6944, page 69440K. International Society for Optics and Photonics, 2008.

[20] Chao-Yung Hsu, Chun-Shien Lu, and Soo-Chang Pei. Image feature extraction in encrypted domain with privacy-preserving sift. *IEEE transactions on image processing*, 21(11):4593–4607,

2012.

[21] Peijia Zheng and Jiwu Huang. Implementation of the discrete wavelet transform and multiresolution analysis in the encrypted domain. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 413–422, 2011.

[22] Tiziano Bianchi, Alessandro Piva, and Mauro Barni. On the implementation of the discrete fourier transform in the encrypted domain. *IEEE Transactions on Information Forensics and Security*, 4(1):86–97, 2009.

[23] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[24] Wenjun Lu, Ashwin Swaminathan, Avinash L Varna, and Min Wu. Enabling search over encrypted multimedia databases. In *Media Forensics and Security*, volume 7254, page 725418. International Society for Optics and Photonics, 2009.

[25] Zhihua Xia, Yi Zhu, Xingming Sun, Zhan Qin, and Kui Ren. Towards privacy-preserving content-based image retrieval in cloud computing. *IEEE Transactions on Cloud Computing*, 6(1):276–286, 2015.

[26] Hang Cheng, Xinpeng Zhang, Jiang Yu, and Fengyong Li. Markov process-based retrieval for encrypted jpeg images. *EURASIP Journal on Information Security*, 2016(1):1, 2016.

[27] Degang Xu, Hongtao Xie, and Chenggang Yan. Triple-bit quantization with asymmetric distance for image content security. *Machine Vision and Applications*, 28(7):771–779, 2017.

[28] Reda Bellafqira, Gouenou Coatrieux, Dalel Bouslimi, and Gwénolé Quellec. Content-based image retrieval in homomorphic encryption domain. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2944–2947. IEEE, 2015.

[29] Bernardo Ferreira, Joao Rodrigues, Joao Leitao, and Henrique Domingos. Practical privacy-preserving content-based retrieval in cloud image repositories. *IEEE Transactions on Cloud Computing*, 2017.

[30] Zhihua Xia, Xinhui Wang, Liangao Zhang, Zhan Qin, Xingming Sun, and Kui Ren. A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. *IEEE transactions on information forensics and security*, 11(11):2594–2608, 2016.

[31] Zhihua Xia, Neal N Xiong, Athanasios V Vasilakos, and Xingming Sun. Epcbir: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing. *Information Sciences*, 387:195–204, 2017.

[32] Yanyan Xu, Jiaying Gong, Lizhi Xiong, Zhengquan Xu, Jinwei Wang, and Yun-qing Shi. A privacy-preserving content-based image retrieval method in cloud environment. *Journal of Visual Communication and Image Representation*, 43:164–172, 2017.

[33] Jiaying Gong, Yanyan Xu, and Xiao Zhao. A privacy-preserving image retrieval method based on improved bovw model in cloud environment. *IETE Technical Review*, 35(sup1):76–84, 2018.

[34] Meng Shen, Guohua Cheng, Liehuang Zhu, Xiaojiang Du, and Jiankun Hu. Content-based multi-source encrypted image retrieval in clouds with privacy preservation. *Future Generation Computer Systems*, 109:621–632, 2020.

[35] Chuan Zhang, Liehuang Zhu, Chang Xu, and Rongxing Lu. Ppdp: An efficient and privacy-preserving disease prediction scheme in cloud-based e-healthcare system. *Future Generation Computer Systems*, 79:16–25, 2018.

[36] Nasir Rahim, Jamil Ahmad, Khan Muhammad, Arun Kumar Sangaiah, and Sung Wook Baik. Privacy-preserving image retrieval for mobile devices with deep features on the cloud. *Computer Communications*, 127:75–85, 2018.

[37] Salahuddin Unar, Xingyuan Wang, Chunpeng Wang, and Yu Wang. A decisive content based image retrieval approach for feature fusion in visual and textual images. *Knowledge-Based Systems*, 179:8–20, 2019.

[38] Zhihua Xia, Leqi Jiang, Dandan Liu, Lihua Lu, and Byeungwoo Jeon. Boew: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing. *IEEE Transactions on Services Computing*, 2019.

[39] Priyanka Singh and Hany Farid. Robust homomorphic image hashing. In *CVPR Workshops*, pages 11–18, 2019.

[40] Dongmei Li, Xiaolei Dong, Zhenfu Cao, and Haijiang Wang. Privacy-preserving outsourced image feature extraction. *Journal of Information Security and Applications*, 47:59–64, 2019.

[41] Intedhar Shakir Nasir. A new approach for content based image retrieval using statistical metrics. *Jour of Adv. Research in Published By: Blue Eyes Intelligence Engineering.*

[42] James Ze Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on pattern analysis and machine intelligence*, 23(9):947–963, 2001.

[43] Marco La Cascia, Saratendu Sethi, and Stan Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *Proceedings. IEEE Workshop on Content-Based Access of Image and Video Li-*

braries (Cat. No. 98EX173), pages 24–28. IEEE, 1998.

[44] Ning Chen. Ci-snf: Exploiting contextual information to improve snf based information retrieval. *Information Fusion*, 52:175–186, 2019.

[45] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, pages 304–317. Springer, 2008.

K. Nalini Sujantha Bel received her M. Sc degree from Bharathiar University, Tamil Nadu in 2007 and M.Phil degree from Manonmaniam Sundaranar University, Tamil Nadu in 2010. At present, she is pursuing research (Reg.No.18123112162001) in the Department of Computer Science, Nesamony Memorial Christian College, Marthandam, affiliated to Manonmaniam Sundaranar University, Abishekapatti,Tirunelveli, India. Her research interests include image processing, network security and multimedia.

I. Shatheesh Sam received Master of Computer Science and Engineering from Sathyabama University, India, in 2006. He received his Ph.D in Information and Communication Engineering at Anna University Chennai, India in 2012. Currently, he is an Associate Professor in the Department of PG Computer Science, Nesamony Memorial Christian College affiliated to Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli. His research interests include multimedia security, network security and image processing. He is life member of CSI, ISTE, IETE and member of IEEE.