Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 7s (2024)

Bi-level feature Classification Approach in Privacy Preserving Cloud Network for High-Performance Content-based Image retrieval Mechanism

[1]**J.Sheeba Selvapattu,** Research Scholar, JAIN Deemed-to-be University, Bangalore, India. s.sheeba@jainuniversity.ac.in

[2]**S.K. Manju bargavi,** Professor, School of Computer Science and IT, JAIN Deemed-to-be University, Bangalore, India. b.manju@jainuniversity.ac.in

Article History:

Received: *01-06-2024*

Revised: *03-07-2024*

Accepted: *29-07-2024*

Abstract:

The enormous storage capacity and easy use of cloud storage have made it extremely popular in recent years. Nevertheless, there's a chance that privacy may be compromised if the picture is uploaded straight to the cloud storage.The content-based image retrieval (CBIR) mechanism proposed in this research maintains the image's privacy content using a Bi-level feature classification approach. This method divides the pixels of the image into two groupssuch aslow and high-level classes. A local encryption technique encrypts the pixels of the low-level class. Two encryption mechanisms are presented in the proposed approachthat uses two different features such as local binary pattern-based position-dependent (LBP-PD) feature and local binary pattern-based position-independent (LBP-PI) feature. The encryption mechanismsare termed asScrambling Without High-Level Component (SWO-HLC) and Scrambling With High-Level Component (SW-HLC). The feature selection factor determines the quantity of high-level features used. The high-level component descriptors from encrypted images were detected by the cloud server and are stored in a storage of the cloud network. The high-level component (HLC) descriptor that are collected from the query encrypted image are compared to the descriptors in the storage of the cloud network to perform the retrieval process. Using the Inria Holidays dataset, the CBIR system was assessed with the criteria including retrieval precision and time complexity. A mean average precision of 62.95% and 67.9% is provided by the SW-HLC and SWO-HLC schemes respectively which is higher than other CBIR schemes used in privacy preserving.

Keywords: CloudStorage,Content-based image retrieval, Feature extraction, Image encryption, Local-Binary Pattern, Privacy-preserving, Security.

1. Introduction

Due to advancements in internet, computer, and smartphone technology, the number of digital images created by users is rising quickly. Large local storage is therefore required for these devices.By transferring photos produced by computers and phones straight to the cloud, cloud storage offers a solution to reduce the amount of physical storage used in these systems. But maintaining the privacy of the image's content is crucial.The 2014 iCloud hack, in which hackers released almost 500 personal images of various celebrities, is one instance of why it is important to preserve the image's content [1]. Furthermore, there's no guarantee that the cloud server won't attempt to access the sensitive information included in photos that users have uploaded to the cloud. Thus, the image's content must be protected before it is uploaded.Before an image is uploaded, the

owner might employ a basic technique called encryption for preserving the privacy content. However, as the encrypted image is kept in a large storage by the cloud server, the process of retrievalis more challenging. There are two categories of content-based image retrieval mechanisms: user-based feature descriptor extractionand cloud-based feature descriptor extraction. Before encrypting the image in the first method, the user must perform feature descriptor extraction, encrypt the images, and then upload both the features and encrypted images. This method places a huge computational strain on the user or owner because it requires them to do tasks like feature extraction, image encryption, and feature extraction. The second method, in which the user simply has to encrypt and upload the image to the cloud, and the cloud server handles feature extraction. This can lessen the computing load since the cloud server extracts the features.

When the user requests a picture for retrieval, the CBIR finds related photos and returns them in a closely matched rating order. There are two modules in the cloud server's CBIR system: one for image retrieval and the other for feature extraction. After extraction of feature, the feature extraction module stores the descriptors in the feature storage. The image retrieval module extract the feature descriptors, which then compares them with the data stored in the cloud. The user will receive the best matching similarity findings. The performance of the CBIR is primarily determined by three processes: (i) the owner’s/user’s encryption and decryption scheme; (ii) the feature descriptor extraction schemeappliedin encrypted images. (iii) theimage descriptor matching scheme yielding results with close matches.

The following outlines the rationale behind the suggested CBIR system:
(i) The Bi-level categorization approach is proposed in this research which divides the pixel blocks into low and high-level blocks that decide the blocks in which encryption is performed and descriptor is extracted.

(ii) The SW-HLC and SWO-HLC encryption techniques are introduced in this paper.
(iii) In addition, this research suggests two low-complexity feature descriptor extraction techniques, LBP-PD and LBP-PI, that are capable of extracting features from high-level classes.
(iv) The Manhattan distance is used to retrieve images, and the k-means approach is utilized to generate visual words.

The structure of the paper is constructed as highlighted below. Some of the relevant works are displayed in Section 2. The suggested CBIR algorithm and the experimental findings are presented in Sections 3 and Section 4, respectively. Section 5 offers the conclusion, at last.

2. Related works

Numerous researchers have presented various CBIR techniques, some of which are included below. The related work section concentrates on feature extraction, feature matching, and encryption/decryption algorithms. Additionally, it displays the two different kinds of CBIR systems, including owner-basedfeature descriptor extraction and cloud-basedfeature descriptor extraction. Under the owner-basedfeature descriptor extraction, the users extract the descriptors and upload the collected features together with the image. The authors Lu et al. [2] encrypted the extracted histograms utilizingmin-hash and order-preserving encryption schemes that was based on a histogram of the visual word characteristics. For retrieval of images, this method used the Jaccard

similarity between the histogram of cloud image and query image. In order to maintain the image's privacy content, the scheme [3] later suggested random projections, along with processes such as random unary encoding and bit plane randomization. This approach uses random unary encoding, random projections with the L_1 distance and bit plane randomization with the Hamming distance.

These methods are not as precise as getting the image in the plain text-domain. Few schemesencrypt the feature vectors using homomorphic encryption [4,5]. The burden of the owner is greatly increased by this strategy, which necessitates cyclic cloud-owner communication for the distance estimation procedure. Weng et al. [6] employ partial encryption, where each image is represented by a hash vector. The predicted hash vector is categorized into two sections: stream cipher is utilized to encrypt one section, while the second is left unencrypted. Xia et al. [7] created secure kNN, which uses a binary vector to split the feature vectors into two vectors and marginally increases search precision. The two vectors are encrypted using two invertible matrices. These techniques lessen the user's load without the need for cloud-user communication.

A search pattern was later employed in which Xia et al. [8] employeda feature descriptor extraction approachutilizing SIFT (scale-invariant feature transform) and a BOW (bag of words), where the earthmover's distance is utilized to estimate how similar the images are. The authors Qin et al. [9] employ a technique called "Speeded-up robust features" (SURF), in which features are encrypted using a chaotic algorithm. To increase retrieval accuracy, it additionally makes use of a local sensitive hash. The author Abduljabbar [10] constructs a tree index for the retrieval procedure; however, this technique raises the image owner's burden.

Cloud-based feature descriptor extraction approaches are introduced, in which the cloudsimplifies feature extraction or index construction process. Xu et al. [11] presented partial encryption, which divides a picture into two parts: an unencrypted component and an encrypted part using advanced encryption standard (AES) encryption. The partial encryption method uses a Gaussian orthogonal matrix to do this. Since the feature extraction algorithm is unable to distinguish between encrypted and unencrypted regions, this method further extracts features from the encrypted regions. Yuan et al. used the cloud server's tree index formation procedure [12] to simplify the owner side's tree index construction procedure. This approach also requires owner-cloud communication though the burden on the user side is reduced in terms of index construction.

In order to maintain privacy, homomorphic encryption is also employed. The wavelet coefficients histogram was utilized by Bellafqira et al. [13] that uses Paillier algorithm.The similarity between the query image and database descriptor is identified by utilizing L_1 distance. Nevertheless, the Paillier algorithm has a significant temporal complexity.Permutation of pixels and row/column substitution technique was employed by the authors Ferreira et al. [14,15] for image encryption. For retrieval, the cloud server uses the Hamming distance and extracts a color histogram. AES encryption was employed by Wang et al. [16] to increase security, however, this minimizes the image retrieval accuracy since random features are utilized which was not as effective.

Cheng et al. [17] suggested the JPEG image-specific privacy-preserving CBIR system, in which the pair of coefficients are constructed from the JPEG bitstreams. The quantization table is constructed by encrypting the DC coefficients using the stream cipher which is robust to attacks [18]. The

authors Xia et al. [19] applied a histogram ofLBPdescriptorson the encrypted image to collect the descriptors. This approach does not yield high-accuracy image retrieval, even though it is more accurate in retrieving a comparable face. In CBIR [20], the Manhattan distance is utilized to evaluate the similarity of the pictures.Discrete cosine transform (DCT), along with the descriptors on the spatial domain are also employed in the retrieval process. The encryption technique uses permutation and substitution, as suggested by Xia et al. [21], that enhances the retrieval accuracy.

Otsu's threshold was employed by Nalini et al. [22] to recover the features from the scrambled image. An instantaneous clustering technique was used to cluster a series of features using the iterative process of Otsu's threshold. This method's retrieval accuracy is primarily dependent on the color attributes. Weng et al. [23] employed a polyalphabetic cipher in conjunction with 3×3 permutation within the context of a secure LBP in a privacy-preserving CBIR system. Secure local binary pattern features can be derived from the encrypted blocks without requiring cloud-owner communication. The bag of words modeland Manhattan distance are utilized in feature construction and feature matching respectively.

The work proposes two different feature extractions namely LBP-PI and LBP-PD that preserve the privacy content by partitioning the blocks into two categories without the need of cloud-user communication. The work also introduces two encryption mechanisms namely SWO-HLC and SW-HLC that encrypt the sensitive regions for privacy preserving.

3. Proposed CBIR method

There are three sub-systems in the suggested CBIR processes: (i) Owner-side (ii). Cloud server (iii) User-side

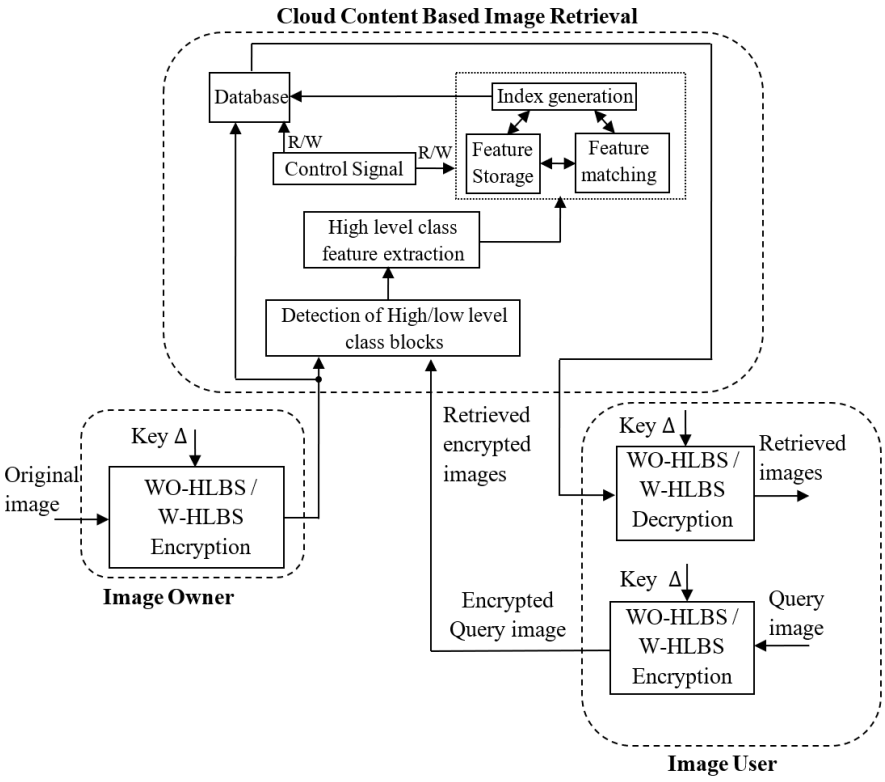

Fig 1: Schematic of Proposed CBIR system

(a) Owner and user side

The person who uploads a picture to the storage of the cloud is the owner of the picture; the person who uploads the picture to the cloud network for the purpose of retrieval is the image user. Fig. 1 illustrates the Schematic of the proposed CBIR's architecture.

(b) Cloud model

The procedure that takes place on the cloud server can be divided into two stages. Storing of the encrypted image to the storage of the cloud constitutes the first part. In this phase, the encrypted image uploaded by the owner is stored, together with the features collected by the cloud network. The user uploads the image to the cloud, and request for retrieving similar contentconstitutes the second stage. In this step, features descriptors are also extracted from the query image that was encrypted and compared to feature vectors that have been stored. The image witha closely matched result is obtained as the retrieval outcome.Like other CBIR systems, this scheme makes the assumption that the cloud storage is an "honest but curious" server. This means that the retrieval and storage services provided by the cloud server are extremely honest. However, the cloud server is curious to access the image's private content.

3.1 **ProposedEncryption and DecryptionProcess**

Let P be the picture that the user (or owner)uploads to the storage of cloud for retrieval (or storage). P_r, P_g and $P_b$resembles the red, green, and blue components of the image, respectively. Let Z represent any one of the channel of the picture$P$. Assume that the size of the picture be$\alpha \times \beta$. Two methods are used in the proposed encryption: (a) scrambling without high-level components (SWO-HLC) and (b) scrambling with high-level components (SW-HLC).

3.1.1 SWO-HLCEncryption

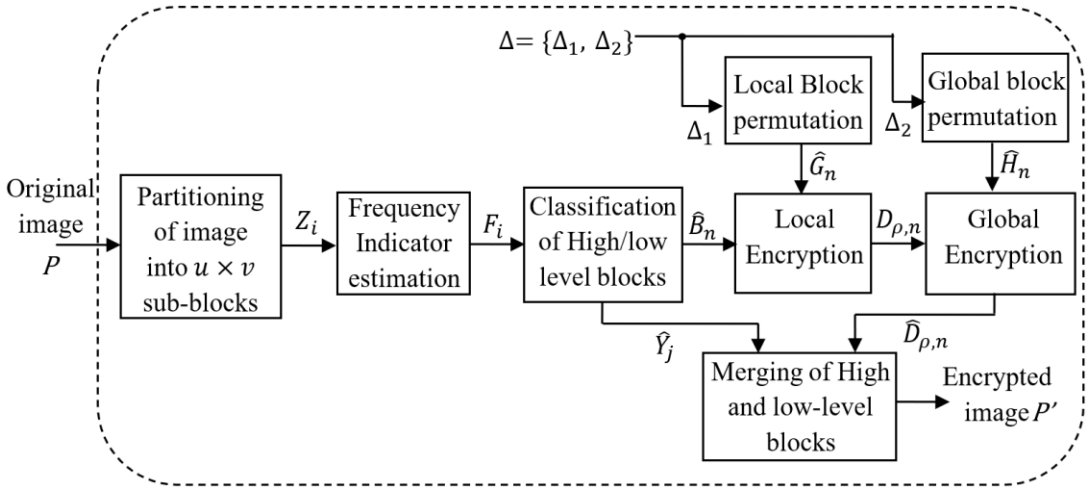

Fig. 2: Architecture of SWO-HLC encryption

Fig. 2 shows the SWO-HLC encryption architecture. The picture is first separated intonon-overlapping blocks, with a smaller size of $u \times v$. As a result$\rho = (\alpha \times \beta)/(u \times v)$gives the total number of blocks per channel. There are two classes into which the ρ sub-blocks $Z_i = [Z_1, Z_2, Z_3 \ldots \ldots Z_\rho]$are classified: as low and high-level classes. Blocks with more information, or

regions that constitute high-frequencyinformation such as edges, and corners, form the high-level category. The low-frequency information region or blocks with redundant or fewer information constitute the low-level category. Thehigh-level categories of blocks can be determined using the below procedure. Let the mean values estimated on each sub-block $Z_i = \{Z_1, Z_2, Z_3 \dots \dots . . Z_\rho\}$ be denoted by$\tau_i = \{\tau_1, \tau_2, \tau_3 \dots \dots . . \tau_\rho\}$. One can estimate the high-frequency information blocks by deducting the average value of $\tau_i$for every sub-block. The high-frequency pixel component is estimated as,

$$\hat{Z}_i(p,q) = Z_i(p,q) - \tau_i \ , p = 1,2, \dots \dots u \text{ and } q = 1,2, \dots \dots v \quad (1)$$

$$\hat{Z}_i(p,q) = Z_i(p,q) - \frac{1}{uv}\sum_{p=1}^{u}\sum_{q=1}^{v} Z_i(p,q) \quad (2)$$

The sub-blocks $\hat{Z}_i(p,q)$contain components that exhibit higher and lower frequency information. Specifically, the larger magnitude of $\hat{Z}_i(p,q)$displays more fluctuating frequency components, whereas the lesser magnitude of $\hat{Z}_i(p,q)$ displays frequency components that are slowly varying. The sub-block frequency indicator can be expressed as,

$$F_i = \sum_{p=1}^{u}\sum_{q=1}^{v}\left|\hat{Z}_i(p,q\right|) \ i = 1,2, \dots . . \rho \quad (3)$$

From minimum to maximumthe frequency indicator $F_i$is sorted. Blocks with a lower frequency indicator value are consideredas low-level classes, whilst those with a higher frequency indicator value are considered as high-level classes. The feature selection factor γ is employed in order to distinguish the frequency indication into two classes. A high-level block is defined as one with an γ number of high frequencies. The sub-blocks $Z_i$are grouped as a low and high-level class using the frequency indicator, denoted by $\hat{B}_n$ and $\hat{Y}_j$, respectively.

$$\hat{Y}_j = argmax(F_i) \ j = 1,2, \dots . . \gamma, i = 1,2 \dots \dots \rho \quad (4)$$

$$\hat{B}_n = argmin(F_i) \ n = 1,2, \dots . . \rho - \gamma, i = 1,2 \dots \dots \rho \quad (5)$$

The low-level class is represented by the indices with the lowest $\rho - \gamma$values of F_i, while the high-level categories are represented by the indices with the greatest γ values. Two encryption techniques are utilized to encrypt the low-level blocks $\hat{B}_n$: local encryption and global encryption. The pair of keys$\Delta = \{\Delta_1, \Delta_2\}$ is utilized to construct the sequence of random numbers.The high-level blocks $\hat{Y}_j$are not encrypted and are retained unaltered.

(a) Local Encryption

Every block's pixel components carry out the local encryption. The key$\Delta_1 = \{\Delta_{1,1}, \Delta_{1,2} \dots \dots . \Delta_{1,\rho-\gamma}\}$, typically represented as $\Delta_{1,n}$is the key that has$\rho - \gamma$ number of sub-keys. Let $\hat{B}_n$be a $u \times v$sized low-class pixel block. Thus, using the associated key $\Delta_{1,n}$, the local blocks are permutedto produce 2D sequence of random numbers$\hat{G}_n$with a size of $u \times v$. The encrypted block $D_{\rho,n}$is then obtained by utilizing the pseudo-random sequence $\hat{G}_n$to scramble the block $\hat{B}_n$. For the entire set of low-level class blocks that need to be encrypted, this process is repeated using its corresponding key.

(b) Global Encryption

Using the key Δ_2, locally encrypted blocks are permuted to perform global encryption. Initially, a 2D random sequence with size $\frac{\alpha}{u} \times \frac{\beta}{v}$is produced by this method. The high-level block locations are then left on the created pseudo-random sequence. Let $\widehat{H}_n$be the sequence of random numbers in 2D form. The global encrypted blocks $\widehat{D}_{\rho,n}$ is constructed, by scramblingthe locally encrypted blocks $D_{\rho,n}$utilizing the random sequence $\widehat{H}_n$. To create the encrypted image P', the high-class blocks $\hat{Y}_j$and the globally encrypted block $\widehat{D}_{\rho,n}$ are combined.

3.1.2 SWO-HLC decryption

Let the encrypted picture be P'. At first, the image which is encrypted is split up into $u \times v$ subblocks. Next, using the equation (3), the frequency indicator $F_i'$is calculated utilizing the subblocks of the image P'. With the parameter γ, the frequency indicator $F_i'$is categorizedas low and high-level categories. The high-level blocks are indicated by the blocks that match the high-frequency indicator values, while the low-level blocks resemble the remaining blocks. Let $D'_{\rho,n}$ and Y_j'represent the low-level and high-level blocks, respectively, where $j = 1,2,3..\gamma$ and $n = 1,2,3 \ldots..\rho - \gamma$. Assume that the decryption key is $\Delta = \{\Delta_1, \Delta_2\}$. The global decryption process utilizes the key Δ_2, whereas local decryption utilizes the key $\Delta_1$as used in encryption.

(a) Global decryption

Let $\frac{\alpha}{u} \times \frac{\beta}{v}$ resembles the size of the 2D random sequence generated utilizing the keyΔ_2. The high-level block locations are then left on the created random sequenceH_n'. To get the globally decrypted blocks $\sigma_{\rho,n}$, the encrypted blocks $D'_{\rho,n}$are once more unscrambled using the pseudo-random sequence H_n'.

(b) Local decryption

For local decryption, the globally decrypted block $\sigma_{\rho,n}$is used. During local decryption, the key $\Delta_1$is used to construct a sequence of keys $\{\Delta_{1,1}, \Delta_{1,2} \ldots\ldots.\Delta_{1,\rho-\gamma}\}$. Let $\sigma_{\rho,n}$ be a global decrypted block (low-class) whose size is$u \times v$. Thus, using the associated key $\Delta_{1,n}$the local blocksare permutedutilizing 2D random sequence G_n'whose size is$u \times v$. The decrypted block is then obtained by unscrambling the block $\sigma_{\rho,n}$using the random sequence G_n'. To retrieve the entire set of decrypted low-level class blocks, this method is applied for each low-level sub-block. To generate the decrypted image, the high-level category blocks Y_j' and low-level category blocks B_n' are merged after decryption.

3.1.3 SW-HLC Encryption

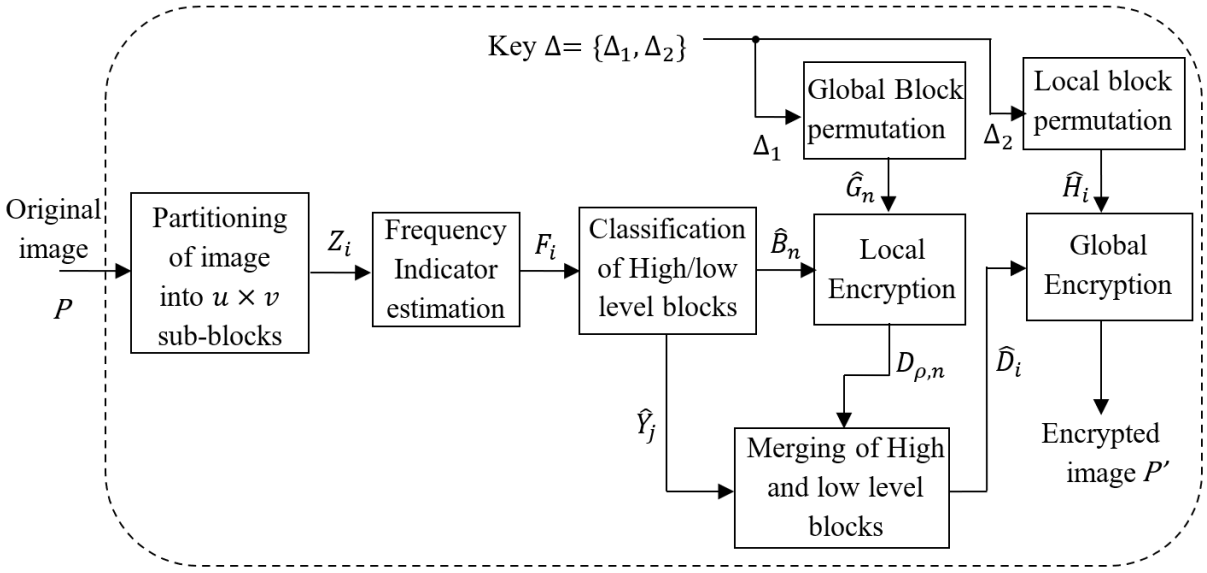

Fig. 3. SW-HLC architecture

The high-level pixels' position in SWO-HLC encryption is constant both before and after encryption. The location of the high-level categories of blocksarescrambled in order to stop privacy leaks from the high-level blocks. Fig.3 shows the SW-HLC architecture. The approach works similar to SWO-HLC encryption, in that it first determines the frequency indicator F_i, based on which the sub-blocks are categorized as low-level blocks $\hat{B}_j$and high-level blocks $\hat{Y}_j$, respectively. The encrypted blocks $D_{\rho,n}$ are obtained by applying local encryption to the low-level blocks $\hat{B}_n$. To create the image $\hat{D}_i$, the encrypted blocks $D_{\rho,n}$ and $\hat{Y}_j$ are combined. To produce the encrypted picture P', the image $\hat{D}_i$is subjected toglobal encryption. The decryption employed in SW-HLC is identical to both the local and global encryptionemployed in the SWO-HLC approach.

3.1.4 SW-HLC Decryption

After the encrypted image P' has been obtained, it is divided into smaller blocks and the frequency indicator is calculated. Using the random sequence produced with the keyΔ_2, the global decryption is performed. The high-level and low-level blocks are then categorized using an estimated frequency indicator. The decrypted image is obtained by applying the local decryption to the low-level categories of blocks and leaving the high-level categories of blocks unaltered.

3.2 Feature extraction

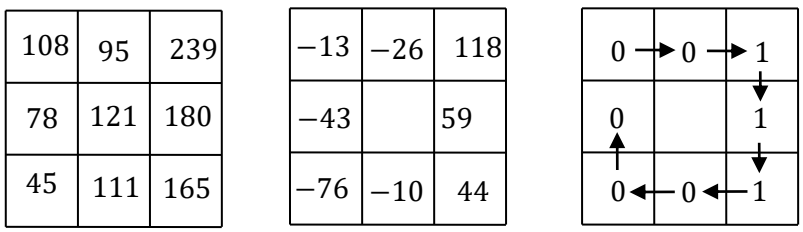

Fig. 4: Representation of LBP feature estimation

The suggested approach makes use of two distinct feature extraction techniques, depending on the type of encryption used. The LBP-based position-independent (LBP-PI) feature extraction and LBP-based position-dependent (LBP-PD) feature extraction procedures are proposed in this research. Fig. 4 shows the procedure for obtaining the LBP descriptor in a 3 × 3 block.

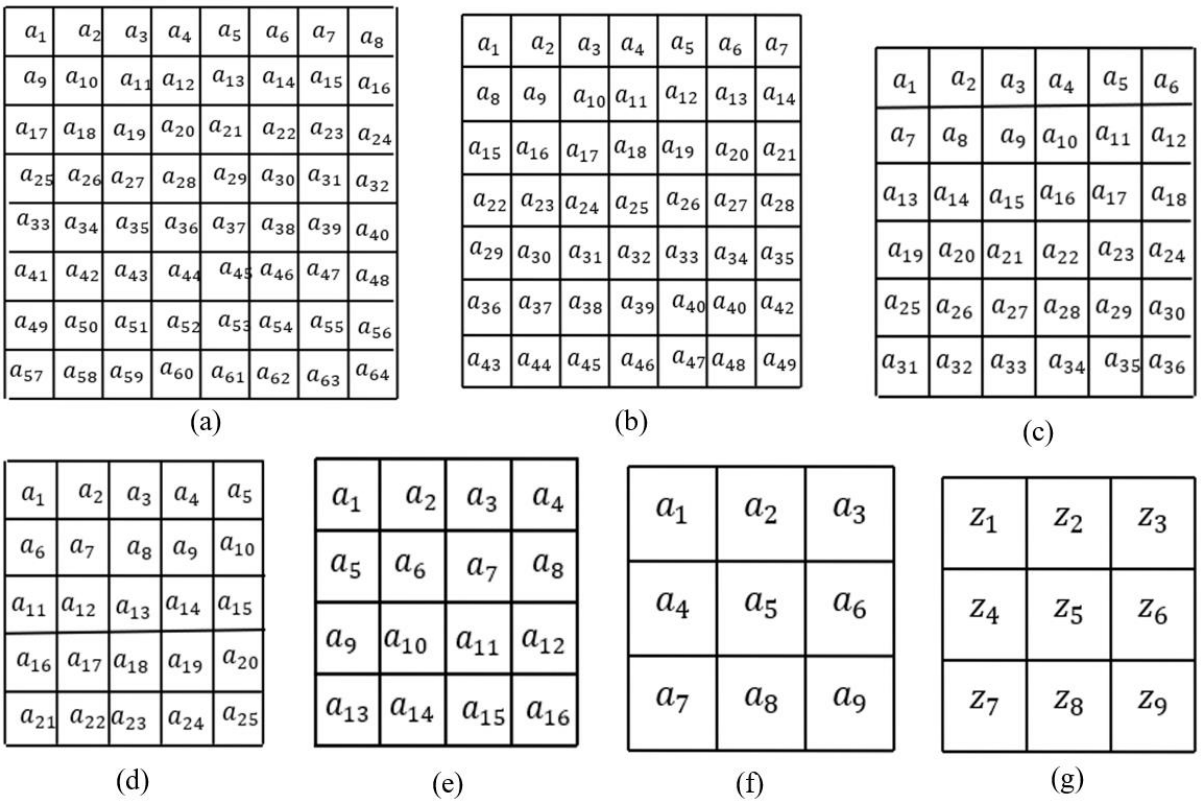

Fig 5: LBP feature extraction represented with varying block sizes (a) 8 × 8 (b) 7 × 7 (c) 6 × 6 (d) 5 × 5 (e) 4 × 4 (f) 3 × 3 (g) Resized image

Table 1:Rule to convert 4 × 4,5 × 5,6 × 6,7 × 7,8 × 8 to 3 × 3size for LBP-PI and LBP-PD feature estimation

	Sub-image size					
	3 × 3	**4 × 4**	**5 × 5**	**6 × 6**	**7 × 7**	**8 × 8**
z_1	a_1	a_1, a_2, a_5	a_1, a_2, a_6, a_7	a_1, a_2, a_7, a_8	$a_1, a_2, a_3, a_8, a_9, a_{10}, a_{15}, a_{16}, a_{17}$	$a_1, a_2, a_3, a_8, a_9, a_{10}, a_{15}, a_{16}, a_{17}$
z_2	a_2	a_2, a_3	$a_2, a_7, a_3, a_8, a_4, a_9$	a_3, a_4, a_9, a_{10}	$a_3, a_4, a_5, a_{10}, a_{11}, a_{12}, a_{17}, a_{18}, a_{19}$	$a_4, a_5, a_{12}, a_{13}, a_{20}, a_{21}$
z_3	a_3	a_3, a_4, a_8	a_4, a_5, a_9, a_{10}	a_5, a_6, a_{11}, a_{12}	$a_5, a_6, a_7, a_{12}, a_{13}, a_{14}, a_{19}, a_{20}, a_{21}$	$a_6, a_7, a_8, a_{14}, a_{15}, a_{16}, a_{22}, a_{23}, a_{24}$
z_4	a_4	a_5, a_9	$a_6, a_7, a_{11}, a_{12}, a_{16}, a_{17}$	$a_{13}, a_{14}, a_{19}, a_{20}$	$a_{15}, a_{16}, a_{17}, a_{22}, a_{23}, a_{24}, a_{29}, a_{30}, a_{31}$	$a_{25}, a_{26}, a_{27}, a_{33}, a_{34}, a_{35}$
z_5	a_5	a_6, a_7, a_{10}, a_{11}	a_{13}	$a_{15}, a_{16}, a_{21}, a_{22}$	$a_{17}, a_{18}, a_{19}, a_{24}, a_{25}, a_{26}, a_{31}, a_{32}, a_{33}$	$a_{28}, a_{29}, a_{36}, a_{37}$
z_6	a_6	a_8, a_{12}	$a_{14}, a_{15}, a_{19}, a_{20}$	$a_{17}, a_{18}, a_{23}, a_{24}$	$a_{19}, a_{20}, a_{21}, a_{26}, a_{27}, a_{28}, a_{33}, a_{34}, a_{35}$	$a_{30}, a_{31}, a_{32}, a_{38}, a_{39}, a_{40}$
z_7	a_7	a_9, a_{13}, a_{14}	$a_{16}, a_{17}, a_{21}, a_{22}$	$a_{25}, a_{26}, a_{31}, a_{32}$	$a_{29}, a_{30}, a_{31}, a_{36}, a_{37}, a_{38}, a_{43}, a_{44}, a_{45}$	$a_{41}, a_{42}, a_{43}, a_{49}, a_{50}, a_{51}, a_{57}, a_{58}, a_{59}$
z_8	a_8	a_{14}, a_{15}	$a_{17}, a_{18}, a_{19}, a_{22}, a_{23}, a_{24}$	$a_{27}, a_{28}, a_{33}, a_{34}$	$a_{31}, a_{32}, a_{33}, a_{38}, a_{39}, a_{40}, a_{45}, a_{46}, a_{47}$	$a_{44}, a_{45}, a_{52}, a_{51}, a_{60}, a_{61}$
z_9	a_9	a_{12}, a_{13}, a_{14}	$a_{19}, a_{20}, a_{24}, a_{25}$	$a_{29}, a_{30}, a_{35}, a_{36}$	$a_{33}, a_{34}, a_{35}, a_{40}, a_{41}, a_{42}, a_{47}, a_{48}, a_{49}$	$a_{46}, a_{47}, a_{48}, a_{54}, a_{55}, a_{56}, a_{62}, a_{63}, a_{64}$

3.2.1 LBP based Position dependent(LBP-PD) feature

The algorithm for extracting LBP-PD features is utilized when the owner and user employs the SWO-HLC encryption. Because the high-level category blocks' positions are fixed, the extracted features in this case are dependent on the feature location. The LBP feature and the location descriptor are first extracted via the LBP-PD feature. The sub-image is resized to 3×3 if the size$u \times v$ (such as 4×4, 5×5, 6×6, 7×7 or 8×8) is greater than 3×3 as illustrated in Table 1. The cloud server classifies the encrypted image blocks as high and low-level categories based on the frequency indicator $F_i$that is estimated from the encrypted image. The high-level blocks are used to extract the LBP-PDdescriptor. Let $\varepsilon_1, \varepsilon_2 \ldots\ldots \varepsilon_\rho$be the LBP features that were obtained from the encrypted image's high-level blocks, commonly represented as ε_i, where, $i = 1,2 \ldots\ldots \rho$. The feature ε_i is positioned at$(s_i, t_i) i = 1,2 \ldots\ldots \rho$. The following relation is used to normalize the position features $(s_i, t_i)$of theimage.

$$\hat{s}_i = s_i - \min(s_i) \quad (6)$$

$$\hat{t}_i = t_i - \min(t_i) \quad (7)$$

The feature $\varepsilon_i$and the position features (s_i, t_i) are organized according to their location from the reference location$(0,0)$. Consequently, the LBP-PD feature can be represented as

$$J_t = \{(\hat{s}_1, \hat{t}_1, \varepsilon_1), (\hat{r}_2, \hat{t}_2, \varepsilon_2) \ldots\ldots\ldots (\hat{s}_L, \hat{t}_L, \varepsilon_\rho)\} \quad (8)$$

where t is the index of the pictureswhich are stored in the cloud and $t = 1,2 \ldots T$. Here the total number of images uploaded to the cloud is T.

3.2.2 LBP based Position independent (LBP-PI)features

The algorithm for extracting LBP-PI feature is utilized when the userandowner employs SW-HLC encryption. Due to the fact that high-level block positions fluctuate throughout the encryption process, the extracted features in this case are independent of feature locations.

The LBP feature and the location data are first extracted via the LBP-PD feature. The sub-image size $u \times v$ is converted to a size of $3 \times 3(z_1, z_2 \ldots\ldots z_9)$ for the extraction of LBP-PI and LBP-PD features utilizing the table shown in Table 1,ifthe size of the sub-image is larger than 3×3 as shown in Fig. 4. Let F_i represent the frequency indicator values to categorize the low and high-level blocks on the encrypted image uploaded to the cloud. The frequency indicator values corresponding to the high-level categories are denoted by F_i. Sort the LBP feature derived from the high-level category blocks according to the frequency indicator values $F_i$ranging from minimum to maximum. Consequently, the features based on LBP-PI extracted from any image can be stated as

$$J_t = \{\varepsilon_1, \varepsilon_2 \ldots\ldots\ldots \varepsilon_\rho\} \quad (9)$$

3.3 CBIR cloud system

The two main functions of the Cloud CBIR system are (i) storing encrypted images and extracting the owner image's features, and (ii) retrieving saved photos in response to query requests.

3.3.1 Extracted features and encrypted image storage

High-level feature extraction and detection are two processes included in the cloud CBIR system. Both the owner-uploaded photos and the features that are extracted are stored using the database.

(a) Generation of visual words

The process of retrieving images from a large database is longer time consumingwhen using one-to-one feature matching. Therefore, using the k-means approach, the features collected from the T images available in the database$J_t$are grouped to ω clusters enabling easy image retrieval from the massive database [24]. Let $e_p = \{e_1, e_2 \dots \dots \dots e_n\}$ be the representation of the features corresponding to the cluster centers, where $n = 1,2 \dots \dots \omega$. If there are a lot of images in the database, the clustering procedure speeds up the retrieval of results. Let$\varphi = 9\rho$ resemble the size of features that were collected from the color image using the LBP-PD algorithm, while $\varphi = 3\rho$ indicates the size of features that were collected from a color picture utilizing the LBP-PI technique. The cloud storage contains the encrypted image, extracted features, and cluster center features.

3.3.1 Image retrieval with Query request

The cloud server divides the encrypted image into subblocks before detecting the high-level blocks as soon as it receives the encrypted query image. The blocks with high class are utilized to extract the LBP-PI and LBP-PD features. Let $C'(l)$. be the features that were taken out of the query image. The cluster centers$e_p$that are present in the database will be matched with the features $C'(l)$via the feature-matching algorithm. The formula for estimating the Manhattan distance is$\tau(p) = \sum_{l=1}^{\varphi} |E_P\,(l) - C'(l)|$. The closest clusters that match the lowest values of $\tau(p)$ are computed from the Manhattan distance resemble the $\hat{k}$ number of retrieved encrypted images. If the user has the authorization and the key Δ, the obtained results can be decrypted.

4. Experimental Results

The Inria Holiday dataset [25] was utilized to assess the suggested CBIR system performance, which was implemented utilizing MATLAB2018a on a Windows 10 with an Intel core I7s CPU running at $3.2GHz$ and $64GB$ of RAM. There are 1491 color photos in 500 categories within the Inria collection. Fig 6 displays sample photographs from the Inria Holiday datasets. The mean average precision (mAP), time ofencryption, query search,feature extraction, and visual word formation were used to evaluate the CBIR system's performance.

Fig 6: Sample query picture from the Inria dataset

The accuracy can be expressed as $\frac{\varphi_1}{\hat{k}}$. Here$\varphi_1$is the number of images that were accurately recovered out of $\hat{k}$ images. The average precision for each category was assessed using the average of $\varphi_3$precision results for each category. This yields the mAP, which is calculated from the average of $\varphi_2$tests that were conducted on various categories.

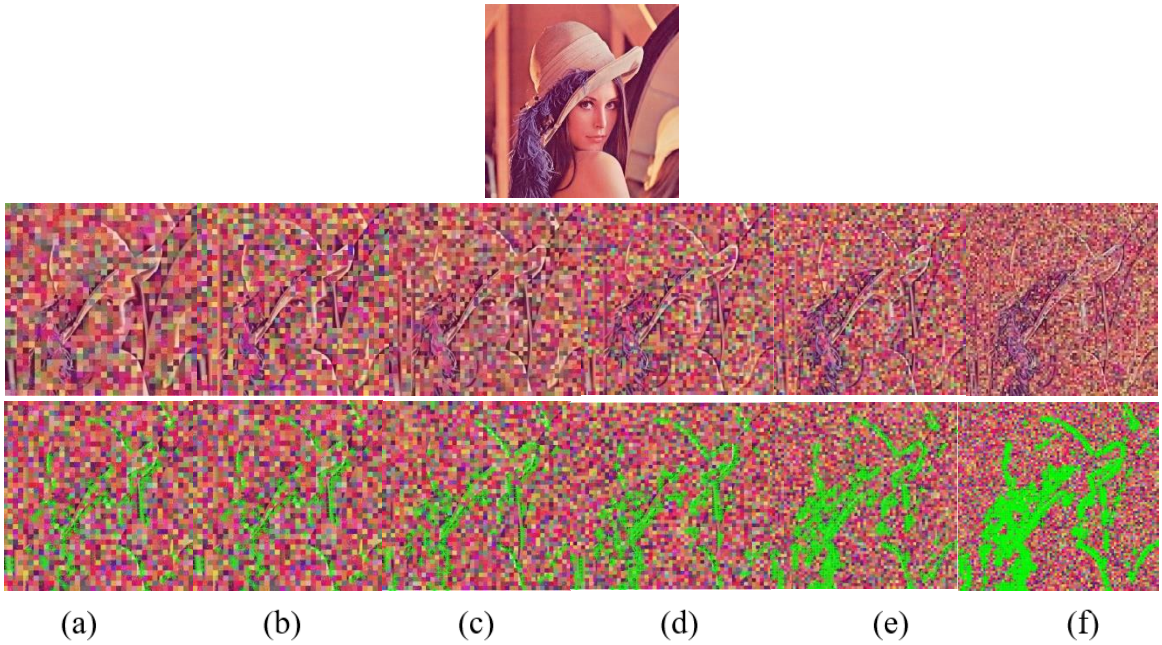

Fig 7: Feature points for SWO-HLC(a) 8 × 8 (b) 7 × 7 (d) 6 × 6 (e) 5 × 5 (f) 4 × 4 (g) 3 × 3(Feature points (row3) obtained on Encrypted image (row2) for Lena input image (row1))

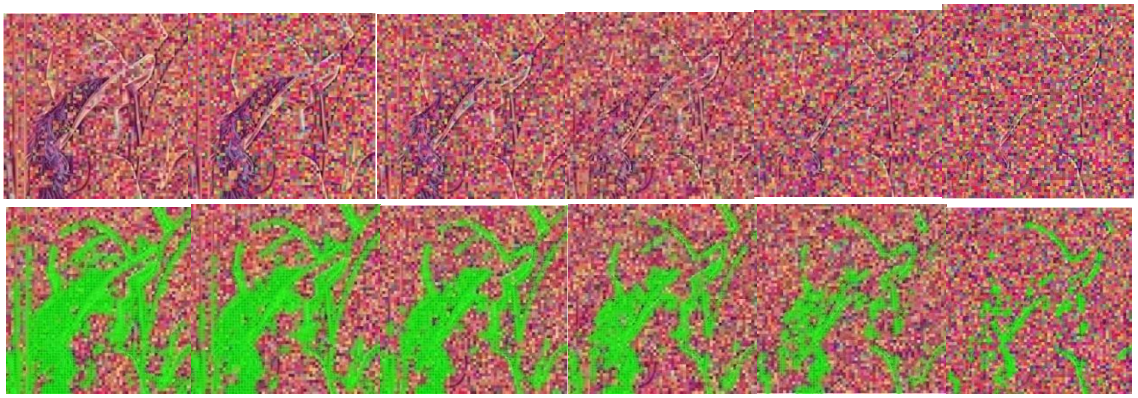

Fig 8: Feature points for SWO-HLC approach for different values of γ (a) $\gamma = 30$ (b) $\gamma = 25$ (c) $\gamma = 20$ (d) $\gamma = 15$ (e) $\gamma = 10$ (f) $\gamma = 5$ (Feature points are illustrated in row2 for the encrypted image illustrated in row1)

The results obtained using the Lena imageis shown in Fig 7, in which the image was encrypted using the SWO-HLC technique for various size of sub-image with $\gamma = 10$. In case of sub-images with larger step-sizes, the high-frequency regions of the image is strongly visible, it is only marginally visible for smaller sub-images. Additionally, encryption with a sub-imagesize of3×3yields more features than the sub-image size of8×8. The encrypted images for various values of γ are shown in Fig. 8, where the size of sub-image is3×3. Just 10% of the regions are utilized for feature descriptor extraction when $\gamma = 10$, whereas 25% of the regionsare utilized when $\gamma = 25$. A higher γvalue reveals moreimage content, whereas a smaller value of γ reveals less image content.

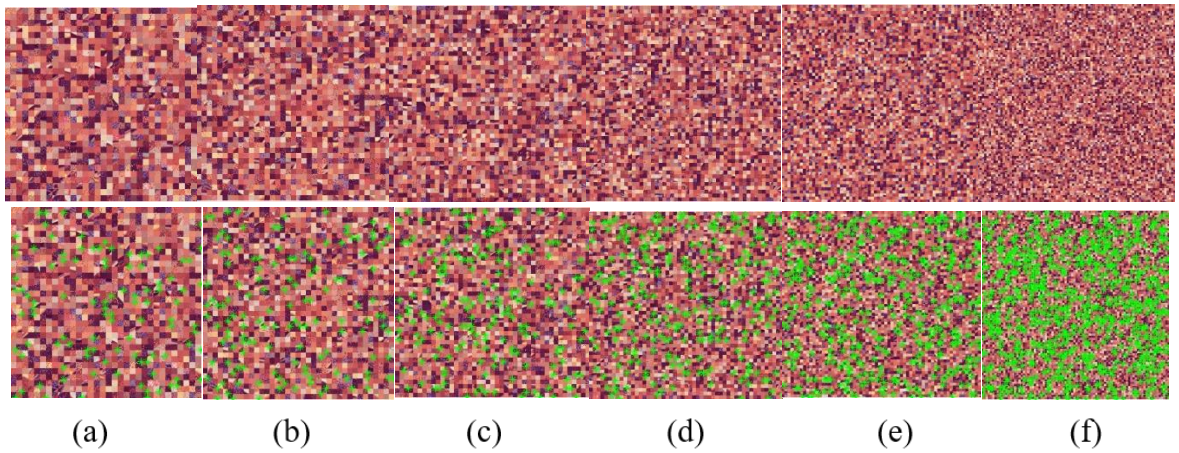

Fig 9: Feature points for SW-HLC (a) 8×8 (b) 7×7 (d) 6×6 (e) 5×5 (f) 4×4 (g) 3×3(Feature points (row2) obtained on Encrypted image (row1))

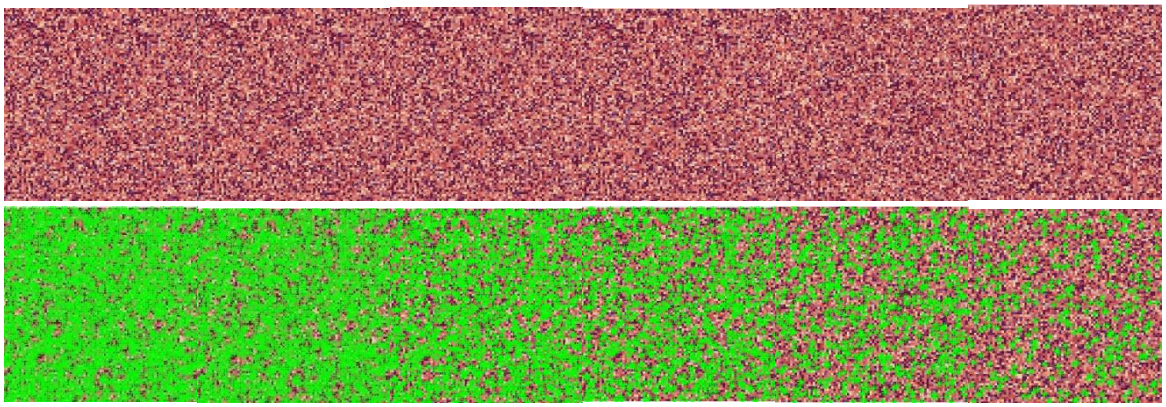

Fig 10: Feature points for SW-HLC approach for different values of γ (a) $\gamma = 30$ (b) $\gamma = 25$ (c) $\gamma = 20$ (d) $\gamma = 15$ (e) $\gamma = 10$ (f) $\gamma = 5$ (Feature points are illustrated in row2 for the encrypted image illustrated in row1)

The results obtained using the Lena imageis shown in Fig 9, in which the image was encrypted using the SW-HLC technique for various size of sub-image with $\gamma = 10$. The content of the image is hidden in smaller sub-images. Because of the highly scrambled high-level component position, it is difficult to discern the true content of large size sub-image, even though the image's content is exposed as blocks. Additionally, encryption with a sub-imagesize of3×3yields more features than the sub-image size of8×8. The encrypted images for various values of γ are shown in Fig. 10, where the size of sub-image is3×3. Just 10% of the regions are utilized for feature descriptor extraction when $\gamma = 10$, whereas 25% of the regions are utilized when $\gamma = 25$. A higher γvalue reveals moreimage content, whereas a smaller value of γ reveals less image content similar to SWO-HLC approach.

Table 2: mAP(%) for the LBP-PD feature extraction with SWO-HLC encryption

Sub-image size	**# of Retrieved images**								**Average**
	100	**500**	**1000**	**2000**	**3000**	**4000**	**5000**	**8000**	
3 × 3	60.10	65.10	65.96	66.08	65.17	64.29	63.40	66.47	64.57
4 × 4	60.55	66.27	66.97	67.16	66.58	65.63	64.72	63.25	65.14
5 × 5	61.71	66.82	67.15	67.70	67.85	66.45	65.54	64.03	65.91
6 × 6	62.91	67.97	68.06	68.51	68.54	66.61	65.65	64.05	66.54
7 × 7	63.03	68.52	68.81	68.85	68.12	67.28	66.77	65.00	67.05
8 × 8	63.10	68.72	69.04	69.09	68.69	67.13	66.21	65.04	67.13
Average	61.90	67.23	67.67	67.90	67.49	66.23	65.38	64.64	66.06

Table 3: mAP(%) for the LBP-PI feature extraction with SW-HLC encryption

Sub-image size	# of Retrieved images								Average
	100	500	1000	2000	3000	4000	5000	8000	
3 × 3	55.16	60.14	61.02	61.1	60.19	59.36	58.48	61.57	59.63
4 × 4	55.64	61.32	61.98	62.21	61.67	60.71	59.77	58.3	60.2
5 × 5	56.73	61.84	62.16	62.79	62.91	61.46	60.58	59.03	60.94
6 × 6	57.93	63.04	63.13	63.56	63.57	61.68	60.75	59.09	61.59
7 × 7	58.04	63.53	63.82	63.95	63.17	62.36	61.79	60.03	62.09
8 × 8	58.17	63.81	64.07	64.1	63.78	62.16	61.28	60.06	62.18
Average	56.95	62.28	62.7	62.95	62.55	61.29	60.44	59.68	61.105

The mAP comparison for the LBP-PD feature extraction-based SWO-HLC encryption is shown in Table 2. With varying values of$\hat{k}$, the mAP increases with various size of sub-image. The mAP value gets increased until the number of retrieved image is$\hat{k} = 2000$, after which they begin to decline. With the suggested SWO-HLC encryptionalong withLBP-PD feature descriptor extraction technique, results in mAP of 67.9% with $\hat{k} = 2000$. The mAP for the LBP-PD feature extraction-based SWO-HLC encryption is shown in Table 3. An mAP of 62.95% is offered by the suggested SW-HLC encryption with LBP-PI feature extraction for $\hat{k} = 2000$.

Table 4: Time complexity (s) forSWO-HLCin Generation of Visual words

Sub-image size	# of Retrieved images							
	100	500	1000	2000	3000	4000	5000	8000
3 × 3	443.99	856.05	873.65	1160.17	1442.5	1695.53	1930.34	2646.43
4 × 4	437.67	702.73	873.26	1118.2	1430.81	1603.95	1928.52	2615.7
5 × 5	419.62	657.01	869.03	1102.53	1361.34	1585.79	1852.08	2552.51
6 × 6	305	603.21	834.36	1097.95	1358.73	1583.61	1824.93	2545.56
7 × 7	222.04	566.29	799.48	998.86	1264.9	1521.65	1720.38	2391.04
8 × 8	79.16	217.39	234.56	313.74	337.37	389.7	443.61	591.44

Table 5: Time complexity (s) forSW-HLCin Generation of Visual words

Sub-image size	# of Retrieved images							
	100	500	1000	2000	3000	4000	5000	8000
3 × 3	405.9	817.53	835.47	1123.09	1404.32	1659.11	1893.4	2609.17
4 × 4	399	666.27	834.62	1079.8	1391.9	1566.85	1891.91	2578.98
5 × 5	381.1	618.11	830.93	1065.08	1324.48	1547.29	1813.46	2516.23
6 × 6	268.4	564.71	795.82	1061.01	1321.89	1546.32	1787.31	2509.27
7 × 7	183.5	528.69	761.16	962.17	1226	1483.4	1683.35	2352.19
8 × 8	41.06	178.87	198.42	277.57	299.14	352.79	406.06	554.97

The time required for visual words formation for the SWO-HLC and SW-HLC schemes is shown in Tables 4 and 5, respectively. Because there are fewer features for larger sub-image sizes, the time

consumption is lower for encryption of smaller sub-image sizes. The time required to generate a visual word rises with the number of clustersω. This increase in visual word generation time is due to the increase in the k-means clustering scheme complexity.

Table 6: Comparison of mAP (%) with traditional CBIR schemes

Schemes	**Secure LBP [23]**	**BOEW [21]**	**Ferreiras scheme [15]**	**Partial-encryption [26]**	**SSE [27]**	**IES [28]**	**Proposed W-HLBS**	**Proposed WO-HLBS**
mAP	51.59	64.24	50.38	56.04	49.07	54.56	62.95	67.9

The suggested method's mAP was compared with schemessuch assecure LBP [23], BOEW [21], Ferreiras [15],partial encryption [26],SSE [27], and IES [28]. As shown in Table 6, the suggested SWO-HLC offers a mAP of 67.9%, which is greater than other recent algorithms, whereas the scheme SW-HLC offers a mAP of 62.95%, that is less than the BOEW approach.

Table 7: Search time (s) for various values of$\hat{k}$

Scheme	**# of Retrieved images**						
	8000	**7000**	**6000**	**5000**	**2000**	**800**	**100**
SW-HLC	0.2552	0.2451	0.2346	0.2201	0.1856	0.1743	0.1683
SWO-HLC	0.2635	0.2532	0.2412	0.2291	0.1987	0.1826	0.1712

Thesearch time for various values of$\hat{k}$is shown in Table 7, where the two methods, SWO-HLC and SW-HLC, have longer search times as the number of clusters grows. For every value of $\hat{k}$, the SW-HLC scheme requires less time of search than the SWO-HLC strategy.

Table 8: Comparison of encryption time

Scheme	**Sub-image size**					
	$\mathbf{8\times8}$	$\mathbf{7\times7}$	$\mathbf{6\times6}$	$\mathbf{5\times5}$	$\mathbf{4\times4}$	$\mathbf{3\times3}$
SW-HLC	214.82	227.86	241.13	259.76	259.76	322.43
SWO-HLC	215.56	228.12	241.87	260.38	249.36	323.17

The image encryption time for different size of sub-image is displayed in Table 8. The amount of high-level blocks decreases with sub-image size, which further cuts down on image encryption time. In comparison to the SWO-HLC scheme, the SW-HLC scheme offers a shorter encryption time.

Table 9: Comparison of feature extraction time

Feature	$\mathbf{8\times8}$	$\mathbf{7\times7}$	$\mathbf{6\times6}$	$\mathbf{5\times5}$	$\mathbf{4\times4}$	$\mathbf{3\times3}$
LBP-PI	1420.3	1473.2	1521.8	1579.4	1634.2	1685.6
LBP-PD	1515.5	1595.7	1657.2	1702.8	1791.4	1856.2

The feature extraction times for the LBP-PD and LBP-PI methods are displayed in Table 9. Compared to the LBP-PD approach, the LBP-PI method requires less time for extraction of features. In both techniques, the feature extraction time decreases with increasing sub-image size.

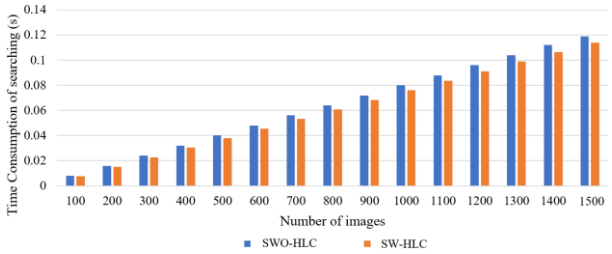

Fig 11: Time of Query search comparison

Fig.11 illustrates the time utilized for Query search comparisonbetween the SWO-HLC and SW-HLC algorithms. The time of visual word formation, encryption, and feature extraction between these two schemes is shown in Fig. 12. The feature extraction phase takes up about 70% of the overall time in both approaches. Roughly 13 percent and 17 percent of the total time is spent on encryption and visual word formation.

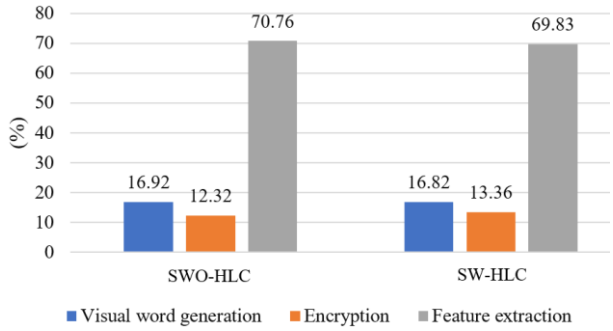

Fig 12: SWO-HLC and SW-HLCschemes time complexity comparison

5. Conclusion

This research presented a CBIR mechanismwhich utilizes low and high-level block classification to preserve privacy content. The work utilizes a local encryption followed by a global encryptionnamed SWO-HLC and SW-HLC that extracts the features using the LBP-PD and LBP-PItechniques. For the purpose of extraction of features, high-level componentsare left unencrypted while low-level components are encrypted. The high-level blocks in the SW-HLC scheme are also scrambled to provide further security, the high-level blocks in the SWO-HLC method are left unencrypted. The Inria Holiday dataset was used for the study, and indicators like time consumption, and mAP were used. SWO-HLC and SW-HLC, the two suggested systems, offer mAP of 62.95% and 67.9% respectively. The amount of time spent on encryption, feature extraction, and visual word synthesis is also used to assess the time consumption. Comparing the SW-HLCscheme against the SWO-HLC method, the scheme SW-HLC reveals less computational time. The suggested SW-HLC technique with LBP-PIfeature works better than other privacy-preserving CBIR systems, in terms of mAP and time consumption.

References

[1] N. Reda, “25 celebrities who were victims of nude photo leaks,” [Online]. Available: https://popcrush.com/celebrities-nude-photo-leaks/

[2] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, “Enabling search over encrypted multimedia databases,” in Media Forensics Secur., vol. 7254, p. 725418, 2009.

[3] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, “Secure image retrieval through feature protection,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2009, pp. 1533–1536.

[4] L. Zhang et al., “Pic: Enable large-scale privacy preserving contentbased image search on cloud,” IEEE Trans. Parallel Distrib. Syst., vol. 28, no. 11, pp. 3258–3271, 2017.

[5] W. Lu, A. L. Varna, and M. Wu, “Confidentiality-preserving image search: A comparative study between homomorphic encryption and distance-preserving randomization,” IEEE Access, vol. 2, pp. 125–141, 2014.

[6] L. Weng, L. Amsaleg, and T. Furon, “Privacy-preserving outsourced media search,” IEEE Trans. Knowl. Data Eng., vol. 28, no. 10, pp. 2738–2751, Oct. 2016.

[7] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, “A privacypreserving and copy-deterrence content-based image retrieval scheme in cloud computing,” IEEE Trans. Inf. Forensics Secur., vol. 11, no. 11, pp. 2594–2608, Nov. 2016.

[8] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, “Towards privacy-preserving content-based image retrieval in cloud computing,” IEEE Trans. Cloud Comput., vol. 6, no. 1, pp. 276–286, Jan.–Mar. 2015.

[9] J. Qin et al., “An encrypted image retrieval method based on harris corner optimization and lsh in cloud computing,” IEEE Access, vol. 7, pp. 24 626–24 633, 2019.

[10] Z. A. Abduljabbar, A. Ibrahim, M. A. Hussain, Z. A. Hussien, M. A. Al Sibahee, and S. Lu, “Eeiri: Efficient encrypted image retrieval in IoT-cloud,” KSII Trans. Internet Inf. Syst., vol. 13, no. 11, pp. 5692– 5716, 2019.

[11] Y. Xu, J. Gong, L. Xiong, Z. Xu, J. Wang, and Y.-Q. Shi, “A privacypreserving content-based image retrieval method in cloud environment,” J. Vis. Commun. Image Representation, vol. 43, pp. 164–172, 2017.

[12] J. Yuan, S. Yu, and L. Guo, “SEISA: Secure and efficient encrypted image search with access control,” in Proc. IEEE Conf. Comput. Commun., 2015, pp. 2083–2091.

[13] R. Bellafqira, G. Coatrieux, D. Bouslimi, and G. Quellec, “Contentbased image retrieval in homomorphic encryption domain,” in Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., 2015, pp. 2944– 2947.

[14] B. Ferreira, J. Rodrigues, J. Leito, and H. Domingos, “Privacy-preserving content-based image retrieval in the cloud,” in Proc. IEEE 34th Symp. Reliable Distrib. Syst., 2015, pp. 11–20.

[15] B. Ferreira, J. Rodrigues, J. Leito, and H. Domingos, “Practical privacypreserving content-based retrieval in cloud image repositories,” IEEE Trans. Cloud Comput., vol. 7, no. 3, pp. 784–798, Jul.–Sep. 2019.

[16] H. Wang, Z. Xia, J. Fei, and F. Xiao, “An aes-based secure image retrieval scheme using random mapping and bow in cloud computing,” IEEE Access, vol. 8, pp. 61 138–61 147, 2020.

[17] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, “Encrypted jpeg image retrieval using block-wise feature comparison,” J. Vis. Commun. Image Representation, vol. 40, pp. 111–117, 2016.

[18]Shijin, Kumar PS, and Dhas D. Edwin. "Simulated attack based feature region selection for efficient digital image watermarking." 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET). IEEE, 2012.

[19] Z. Xia, L. Jiang, X. Ma, W. Yang, P. Ji, and N. N. Xiong, “A privacypreserving outsourcing scheme for image local binary pattern in secure industrial Internet of Things,” IEEE Trans. Ind. Informat., vol. 16, no. 1, pp. 629–638, Jan. 2020.

[20] Z. Xia, L. Lu, T. Qiu, H. Shim, X. Chen, and B. Jeon, “A privacy-preserving image retrieval based on ac-coefficients and color histograms in cloud environment,” Comput., Mater. Conti., vol. 58, no. 1, pp. 27–44, 2019.

[21] Z. Xia, L. Jiang, D. Liu, L. Lu, and B. Jeon, “BOEW: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing,” IEEE Trans. Serv. Comput., pp. 1–1, 2019.

[22] Bel, K. Nalini, and I. Shatheesh Sam. "OT-Feature Extraction on Scrambled Images with Instantaneous Clustering for CBIR Scheme in Cloud Computing." *The ISC International Journal of Information Security* 13.1 (2021): 1-17.

[23] Xia, Zhihua, et al. "A Privacy-Preserving Image Retrieval Scheme Using Secure Local Binary Pattern in Cloud Computing." *IEEE Transactions on Network Science and Engineering* 8.1 (2020): 318-330.

[24] Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. "The global k-means clustering algorithm." *Pattern recognition* 36.2 (2003): 451-461.

[25] H. Jegou, M. Douze, and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” Computer Vision–ECCV 2008, pp. 304–317, 2008.

[26] Y. Xu, J. Gong, L. Xiong, Z. Xu, J. Wang, and Y. qing Shi, “A privacy-preserving content-based image retrieval method in cloud environment,” Journal of Visual Communication and Image Representation, vol. 43, pp. 164 – 172, 2017. [Online]. Available: http://www.sciencedirect.com/ science/article/pii/S104732031730007X

[27] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, “Enabling search over encrypted multimedia databases,” Proceedings of SPIE The International Society for Optical Engineering, vol. 7254, pp. 725 418– 11, 2009.

[28] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos, “Practical privacy-preserving content-based retrieval in cloud image repositories,” IEEE Transactions on Cloud Computing, vol. PP, no. 99, pp. 1–1, 2017.