# PRIVACY-PRESERVING CBIR SYSTEM USING SIAMESE TWIN NETWORK WITH SEGNET ARCHITECTURE-BASED HIGH-LEVEL REGION DETECTION

**J. Sheeba Selvapattu[1] & Dr. S.K Manju Bargavi[2]**

*[#1]Research Scholar, JAIN Deemed-to-be-University, Bangalore, India Email: s.sheeba@jainuniversity.ac.in*
*[2]Professor, JAIN Deemed-to-be-University, Bangalore, India Email: b.manju@jainuniversity.ac.in*

**KEYWORDS**
Content-based image retrieval,
SegNet architecture,
Privacy-preserving,
encryption,
Siamese twin network,
Cloud environment

**ABSTRACT:**
The work introduces a Content-based image retrieval (CBIR) approach that can preserve the privacy content of the picture using two deep learning architectures namely Siamese twin network and SegNet architectures. In this approach, the pictures that are uploaded to the cloud are initially separated into two level components namely low and high. The low-level components are encrypted using a block-based permutation approach to preserve the picture privacy content. The resultant image is uploaded to the cloud, where the cloud environment uses a SegNet architecture to segment the high-level components. The high-level component and the low-level encrypted regions are merged to extract features. The SegNet architecture results in a segmentation accuracy, recall, and precision of 98.14%, 96.74%, and 97.63% respectively when evaluated using the Corel-10k dataset. The descriptors are then collected from the merged image and clustered utilizing a recursive tuneable clustering approach. During the retrieval process, the Siamese network is utilized to match the selected leader and followers estimated by the clustering algorithm. The recursive tunable clustering approach reduces the complexity during the retrieval process. The suggested CBIR system was evaluated utilizing the scale such as time complexity and mean average precision (mAP) with the datasets namely Corel-10k and Inria Holiday databases. The proposed CBIR system results in a mAP of 69.27% and 64.53% when evaluated using the Corel-10k and Inria Holiday dataset respectively which is higher than similar recent CBIR systems.

## 1. INTRODUCTION

Image retrieval [1] is an active research domain in the field of computer vision due to its increased practical value. The image retrieval process aims to identify the pictures which are similar to query pictures from a database that has a huge collection of images. Due to the development of cloud storage, the user prefers utilizing the cloud instead of using local storage. The cloud environment [2] has the advantage of accessing the cloud data at any time from any location. Though it adds a huge advantage to the user, the leakage of image privacy content is a threat to untrustworthy cloud. To address this threat, the common approach is that the user needs to transform the image to cipher utilizing an encryption algorithm [3] or hiding the image in media like video or images [4]. A data hiding approach [5,6] was used to preserve the privacy content of the picture by embedding the image in another cover image, which highly increases the workload of user. Therefore, an encryption-based approach is preferred to preserve the privacy content. Encrypting the complete image makes the retrieval process more challenging. Encrypting the majority of content, and leaving few contents (partial encryption) helps to improve the performance in image retrieval. Therefore, it is essential to develop a retrieval algorithm that matches the features from the image which was partially encrypted. The extraction of

descriptors is difficult for the cloud server from the cipher image which was completely encrypted by the encryption algorithm [7], [8].

Numerous CBIR approaches [9, 10] that preserves the privacy content of the image was proposed by various researchers. Deep learning approaches play an important role in different classification problems [11, 12]. A vision transformer was utilized in extracting discriminative features from cipher images [13]. In this scheme features like Huffman code-based frequency descriptors and local length multi-level sequences are extracted in the first level. The vision transformer is further utilized to perform an actual retrieval process. For a multi-user multi-owner environment, a dynamic retrieval mechanism was introduced by Chenyang et al. [14] in which the Chameleon function was used to perform dynamic authentication. Also, a polynomial-based retrieval mechanism was finally utilized to perform retrieval. A transfer learning approach was introduced by Khan et al. [15] for retrieving the image in the cloud environment. This approach retrieves the images in two phases, where the features are collected in the first phase utilizing a pre-trained CNN model. A second phase focuses on the transfer of the feature vector that corresponds to the query image for retrieval.

Ma et al. [16] used deep convolutional network descriptors for retrieving the images from the cloud server. In this process, the pictures are initially encrypted utilizing a hybrid encryption algorithm, and the picture descriptors are collected by the improved DenseNet network. The authors reported that the usage of the improved DenseNet model which used depth-wise convolution provides 8 to 9 times lower parameters and computational complexity than basic DenseNet architecture. An adaptive fusion approach with a deep learning mechanism was used in image retrieval [17], in which descriptors like semantic high-level descriptors, bag of words descriptors, and edge histogram features was utilized in feature matching. Principle component analysis is then utilized to decrease the size of semantic descriptors. The resultant features that are extracted by different approaches are fused adaptively. To minimize the search time, a pre-filter table is constructed using a locality-sensitive hashing.

A deep auto-encoder-based image retrieval mechanism was proposed by Rahim et al. [18] that uses a low complex encryption approach which is suitable for mobile devices. In this scheme, a deep auto-encoder transforms the extracted descriptors to compact binary sequences. During the retrieval process, the mobile device that request for retrieval query will receive the hash codes that are already recorded in a hash table. To reduce the search time a nearest neighbor searching mechanism was utilized. Li et al. [19] used vector homomorphic encryption along with deep convolutional neural network (DCNN). In this approach hash algorithms, and DCNN are used in feature extraction that increases the accuracy during retrieval process. The complexity during the search process is minimized by the construction of a tree that holds the encrypted indices, where the encrypted indices are generated using a k-means clustering approach in combination with vector homomorphic encryption.

The authors Lu et al. [20] used the JPEG compression mechanism to perform the encryption process in which $16 \times 16$ blocks are transformed using discrete cosine transform (DCT) and the $8 \times 8$ blocks of the DCT coefficients are permutated to attain the encryption process. During the phase of retrieval, the DCT histograms are extracted from the query encrypted image that acts as query image descriptors which are fed to the trained model to retrieve similar pictures. For reducing the time of picture retrieval, Zhang et al. [21] introduced an index structure generation process using hash codes. The extracted hash codes and deep visual descriptors are learned using a triplet DCNN network. For the extraction of high-level descriptors, the authors Chen et al. [22] introduced a deep hash technique. The index is further encrypted using a secure k-nearest neighbor algorithm. However, the computational burden of this approach during the retrieval process is higher. ResNet v2 with inception architecture was utilized for the retrieval of pictures in a cloud environment [23]. In this approach, the descriptors are collected by the ResNet v2 architecture. To preserve the picture's privacy content double chaotic map is used as an encryption process.

Most of the schemes that are discussed above does not uses a trained deep learning-based segmentation algorithm to segment the unencrypted regions of the image to extract the retrieval features. Therefore, the proposed approach uses a modified SegNet algorithm in the cloud to detect the location of the unencrypted region. The proposed approach also preserves the privacy content of the image by encrypting the low-level regions of the picture by the owner or user. Thus, the contributing parts of the work are highlighted as follows.

(i) The work introduces an image retrieval algorithm that uses a Siamese twin network for extracting the feature from the high-level region and matches it with the images of the dataset.

(ii) The proposed approach utilizes an encryption algorithm that initially separates the image into two regions such as the low and high-level regions. The encryption algorithm encrypts the low-level region leaving the high-level region for extraction of features by the cloud server.

(iii) The work also proposes a modified SegNet-based segmentation algorithm that helps the cloud server to detect the high-level region for feature extraction.

(iv) The algorithm also proposes a recursive tunable K-means clustering algorithm which was derived from the K-means algorithm that tunes the K-means in different recursive stages.

(v) Finally, the suggested CBIR system was evaluated utilizing the datasets namely Corel-10k and Inria Holidays with scales namely mean average precision and computational complexity. The performance of the modified SegNet architecture in detecting high-level regions was evaluated by utilizing measures namely accuracy, recall, precision, specificity, and F1-score.

The framework of the paper is framed as below. Section 2 discusses on the suggested CBIR system, while the analysis of the CBIR system is discussed in Section 3. Finally, the brief conclusion of the work with the key results is highlighted in Section 4.

## 2. PROPOSED METHODOLOGY

The structure of the suggested CBIR system is illustrated in Fig. 1. The proposed CBIR system includes three sections such as (a) Owner section (b) Cloud section and (c) User section. The major process involved in the image retrieval system includes (a) Encryption/Decryption process (b) High-level region detection (c) Recursive tunable K-means clustering (d) Siamese twin network for feature extraction and (e) Retrieval process.

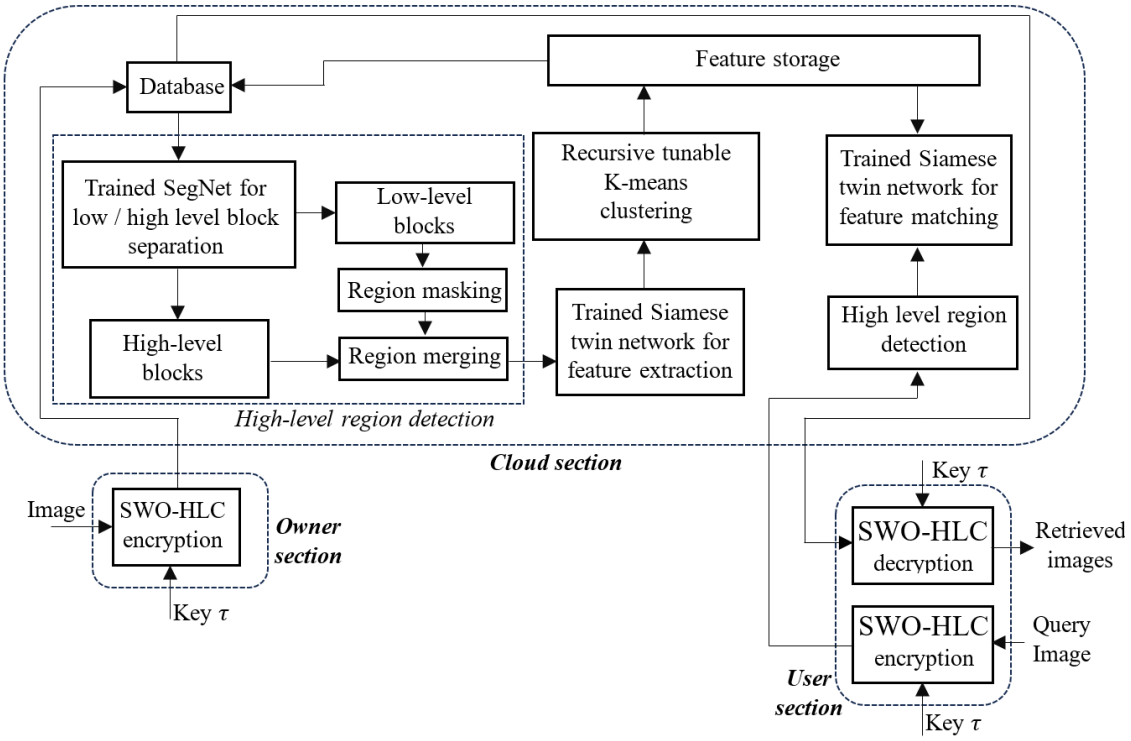



Fig. 1: Layout representation of the suggested CBIR system

### 2.1 Encryption/decryption process

The proposed approach uses scrambling without high-level component (SWO-HLC) [24] encryption to encrypt the plain image. In the SWO-HLC approach, the input image channels are initially grouped into non-overlapping blocks where each block has a constant dimension of $L_1 \times L_2$. Thus the original image totally has $B$ number of blocks.

$$B = \frac{N_1 \times N_2}{L_1 \times L_2} \quad (1)$$

Here $N_1 \times N_2$ resembles the color image size. The sub-divided block is then categorized into either one of the groups as a low-level group, or a high-level group resembling the low, and high-frequency components respectively. Let the low, and high-level blocks be denoted as $R_i^{(L)}$, and $R_i^{(H)}$ respectively. Let the low-level blocks be represented as $R_i^{(L)} = \{R_1^{(L)}, R_2^{(L)}, R_3^{(L)}, \ldots\ldots R_F^{(L)}\}$, and the high-level blocks be expressed as $R_i^{(H)}$=$\{R_1^{(H)}, R_2^{(H)}, R_3^{(H)}, \ldots\ldots R_{B-F}^{(H)}\}$. Here $F$ resemble the number of low-level blocks. The smoothness component is utilized to categorize the low, and high-level components. The smoothness component is estimated as,

$$\hat{R}_i(x,y) = R_i(x,y) = \frac{1}{L_1 \times L_2} \sum_{x=1}^{L_1} \sum_{y=1}^{L_2} R_i(x,y) \quad (2)$$

Using the smoothness components, the frequency indicator is estimated for each sub-block as,

$$J_i = \sum_{x=1}^{L_1} \sum_{y=1}^{L_2} \left| \hat{R}_i(x,y) \right| \quad (3)$$

$$J_i = \sum_{x=1}^{L_1} \sum_{y=1}^{L_2} \left| R_i(x,y) - \frac{1}{L_1 \times L_2} \sum_{x=1}^{L_1} \sum_{y=1}^{L_2} R_i(x,y) \right| \quad i = 1,2,\ldots.B \quad (4)$$

The frequency indicator is sorted based on its increasing value, and the blocks that correspond to the least $F$ values are grouped as low-frequency blocks $R_i^{(L)}$, and other blocks are grouped as high-frequency block $R_i^{(H)}$. The high-frequency block is left unencrypted, while the low-level blocks are encrypted by local, and global block permutation. Using key $\tau$, two intermediate key $\tau_1$, and $\tau_2$ is derived $\tau \in [\tau_1, \tau_2]$ which is used in local, and global block permutation respectively. The key $\tau_1$ is utilized to generate $F$ pseudo-random sequences each having a length of $L_1 \times L_2$. Utilizing the pseudo-random sequence, the elements in the blocks are permutated to complete the local encryption. Let the local encryption be represented as $C_i^{(L)} = local\_encrypt\ (R_i^{(L)}, \tau_1)$. The inverse operation is done to complete the local decryption process represented as, $R_i^{(L)} = local\_decrypt\ (C_i^{(L)}, \tau_1)$. The key $\tau_2$ is utilized to create a pseudo-random sequence having a length of $F$. The position of the blocks $C_i^{(L)}$ is permuted utilizing the pseudo-random sequence created with the key $\tau_2$. Thus global encryption is represented as $D_i^{(L)} = global\_encrypt\ (C_i^{(L)}, \tau_2)$. The inverse operation can be done to complete the global decryption process represented as $C_i^{(L)} = global\_decrypt\ (D_i^{(L)}, \tau_2)$. The algorithm for SWO-HLC encryption and decryption are summarized in algorithms 1, and 2 respectively.

**Algorithm 1:** ***SWO-HLC encryption***

**Input:** Secret key $\tau$, Plain RGB image $I_k$
**Output:** Cipher image $\hat{I}_k$
***Step 1***: Compute $B$ using equation (1).
***Step 2***: Subdivide the image into non-overlapping blocks $R_i$.

***Step 3***: Estimate the smoothness component $\hat{R}_i(x, y)$ using equation (2).
***Step 4***: Estimate the frequency indicator $J_i$ using equation (3).
***Step 5***: Separate the low-level $R_i^{(L)}$, and high-level blocks $R_i^{(H)}$ using the relations
$R_i^{(L)} = \underset{1 \le i \le F}{argmin}\,(J_i)$ and $R_i^{(H)} = \underset{1 \le i \le B-F}{argmax}\,(J_i)$.
***Step 6***: Generate intermediate key $\tau_1$, and $\tau_2$ using the key $\tau$.
***Step 7*** : Perform local encryption on the high-level components as
$C_i^{(L)} \leftarrow local\_encrypt\ (R_i^{(L)}, \tau_1)$
***Step 8***: Perform global encryption on $C_i^{(L)}$ as $D_i^{(L)} \leftarrow global\_encrypt\ (C_i^{(L)}, \tau_2)$
***Step 9***: Merge the encrypted low-frequency component $D_i^{(L)}$ with high-frequency component $R_i^{(H)}$ to obtain the encrypted image $\hat{I}_k$.
***Step 10***: Repeat steps 2 to 9 for R, G, and B channels.

**Algorithm 2: *SWO-HLC decryption***

**Input:** Secret key $\tau$, Cipher RGB image $\hat{I}_k$
**Output:** Plain image $I_k$
***Step 1***: Estimate the number of blocks *B*
***Step 2***: Subdivide $\hat{I}_k$ to non-overlapping blocks.
***Step 3***: Estimate frequency indicator $J_i$ using equation (4) for the image $\hat{I}_k$.
***Step 4***: Using $J_i$ separate the blocks into low and high-level blocks as
$D_i^{(L)} = \underset{1 \le i \le F}{argmin}\,(J_i)$ and $D_i^{(H)} = \underset{1 \le i \le B-F}{argmax}\,(J_i)$.
***Step5***: Using the encryption key $\tau_1$, generate the intermediate keys.
***Step 6***: Perform global decryption followed by local decryption on the low-level blocks
$C_i^{(L)} \leftarrow global_decrypt\left(D_i^{(L)}, \tau_1\right)$and $R_i^{(L)} \leftarrow local\_decrypt\ (C_i^{(L)}, \tau_2)$
***Step7***: Merge the decrypted low-level component $R_i^{(L)}$ with high-level components $D_i^{(H)}$ to obtain the decrypt image $I_k$.
***Step 8***: Repeat steps 2 to 7 for R, G, and B channels.

### 2.2 High-level region detection

The high-level regions are utilized to extract the features and to match the features during the retrieval process by the Siamese twin network. Thus, the two processes involved in high-level class region detection are modified adaptive SegNet architecture and the formation of high-level regions.

*(a) Modified adaptive SegNet architecture*

The SegNet is a deep learning architecture used for pixel-wise classification tasks such as image segmentation. The modified SegNet structure is derived from the traditional SegNet architecture [25] that has a decoder network followed by an encoder network as illustrated in Fig. 2. The modified SegNet architecture is utilized to segment the high-level region. The proposed adaptive SegNet structure uses a convolutional kernel followed by an adaptive convolutional kernel.

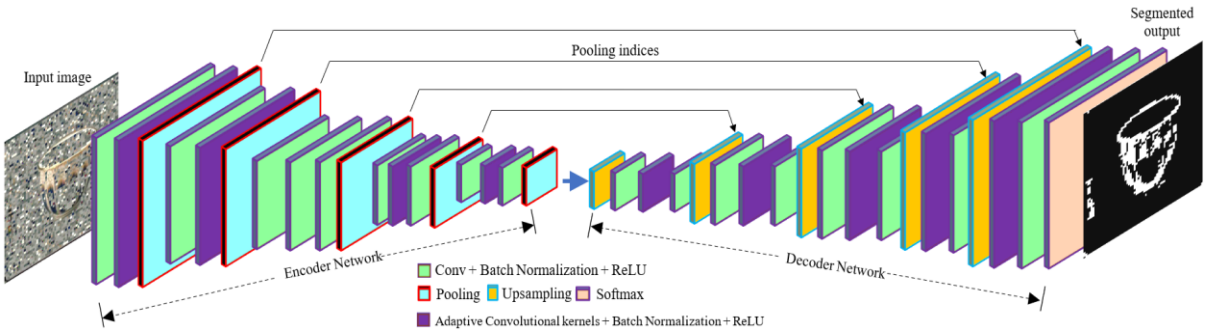



Fig.2: Adaptive SegNet architecture

The convolutional kernel updates its weight during the backward pass based on the loss function

$$\beta_0 = -\sum_{k=1}^{2} y_k \,.\log P(y_k) \tag{5}$$

Here $y_k$ resembles the segmented result. Let the input to the softmax layer that is present as the last layer of the decoder path be $h_j$, then the output of the softmax layer be

$$\hat{g}(h_j) = \frac{\exp(h_j)}{\sum_{c=1}^{2} \exp(h_c)} \tag{6}$$

Here $c$ resembles the segmented classes background and foreground respectively. The convolutional layer updates its weight $\hat{c}(m)$ using the gradient of the loss function

$$\hat{c}(m+1) = \hat{c}(m) - \tau \frac{\partial \beta_0}{\partial \hat{c}} \tag{7}$$

Here $\frac{\partial \beta_0}{\partial \hat{c}} = \frac{\partial \beta_0}{\partial h_j} \times \frac{\partial h_j}{\partial \hat{c}}$ and $m$ resembles the iteration number and $\tau$ resembles the learning rate. The difference between the convolutional kernel and the adaptive convolutional kernel is the weight update of the convolutional kernel depends only on the loss function, but, the weight update of the adaptive kernel also depends on the mean square error. The weight-updated equation of the proposed adaptive kernels can be expressed as,

$$\hat{c}_1(m+1) = \frac{1}{2}[\hat{c}(m+1) + \hat{c}_1(m)] + \mu_0 \times \varepsilon_0 \tag{8}$$

Where $\mu_0 = 0.05$ resembles the step-size, and $\varepsilon_0$ resembles the mean square error between the ground truth $s(x,y)$, and the segmented result $y(x,y)$ estimated as

$$\varepsilon_0 = \frac{1}{L_1 \times L_2} \sum_{x=1}^{L_1} \sum_{y=1}^{L_2} (s(x,y) - y(x,y))^2 \tag{9}$$

Substituting $\hat{c}(m+1)$ from equation (7) in equation (8), the update of adaptive weight can be expressed as,

$$\hat{c}(m+1) = \frac{1}{2}\left[\hat{c}(m) - \eta \frac{\partial \beta_0}{\partial \hat{c}} + \hat{c}_1(m) + \mu_0 \times \varepsilon_0\right] \tag{10}$$

*(b) Formation of high-level class region*

The trained modified SegNet architecture is utilized in the cloud which was trained using the segmented labels generated by the SWO-HLC encryption. The high-level region generated by the SWO-HLC encryption resembles the foreground region, while the low-level region generated by the SWO-HLC encryption resembles the background region. The trained model can able to detect the high-level region for extracting the feature descriptor, instead of using the frequency indicator-based high-level class region detection which was used in the encryption/decryption algorithm. The trained SegNet model will detect the high-level class region which is represented by binary '1'. The segmented result is again refined based on the subblocks. i.e if a sub-block has more than $(L_1 \times L_1)/2$ number of foreground pixels (logic '1'), then, all the pixels in the sub-blocks are assigned as foreground pixels (logic '1'). Reversely, if a sub-block has less than $\frac{L_1 \times L_1}{2} + 1$ number of foreground pixels, then all the pixels in the corresponding sub-blocks are assigned as background pixels (logic '0').

**2.3 Recursive tunable K-means clustering**

Let $P_1, P_2, \ldots \ldots P_G$ represents the feature vector for the pictures in the database which is represented as $Y_1, Y_2, \ldots \ldots Y_G$. The extraction of features is done using the trained Siamese twin network which was provided in Section 2.4. The suggested clustering process was derived from the traditional k-means clustering algorithm [26]. The k-means clustering initializes $K$ number of centroids for K-clusters. The distance between each feature vector $P_i$, and the cluster center $e_j$ is computed using the relation

$$z(P_i, e_j) = \|P_i - e_j\| \tag{11}$$

Here $\|\cdot\|$ resembles the Euclidean norm. The clustering algorithm then assigns each feature vector $P_i$ to a cluster, in which its centroid is closest as

$$u_i = \underset{j}{argmin}\ z(P_i, e_j) \tag{12}$$

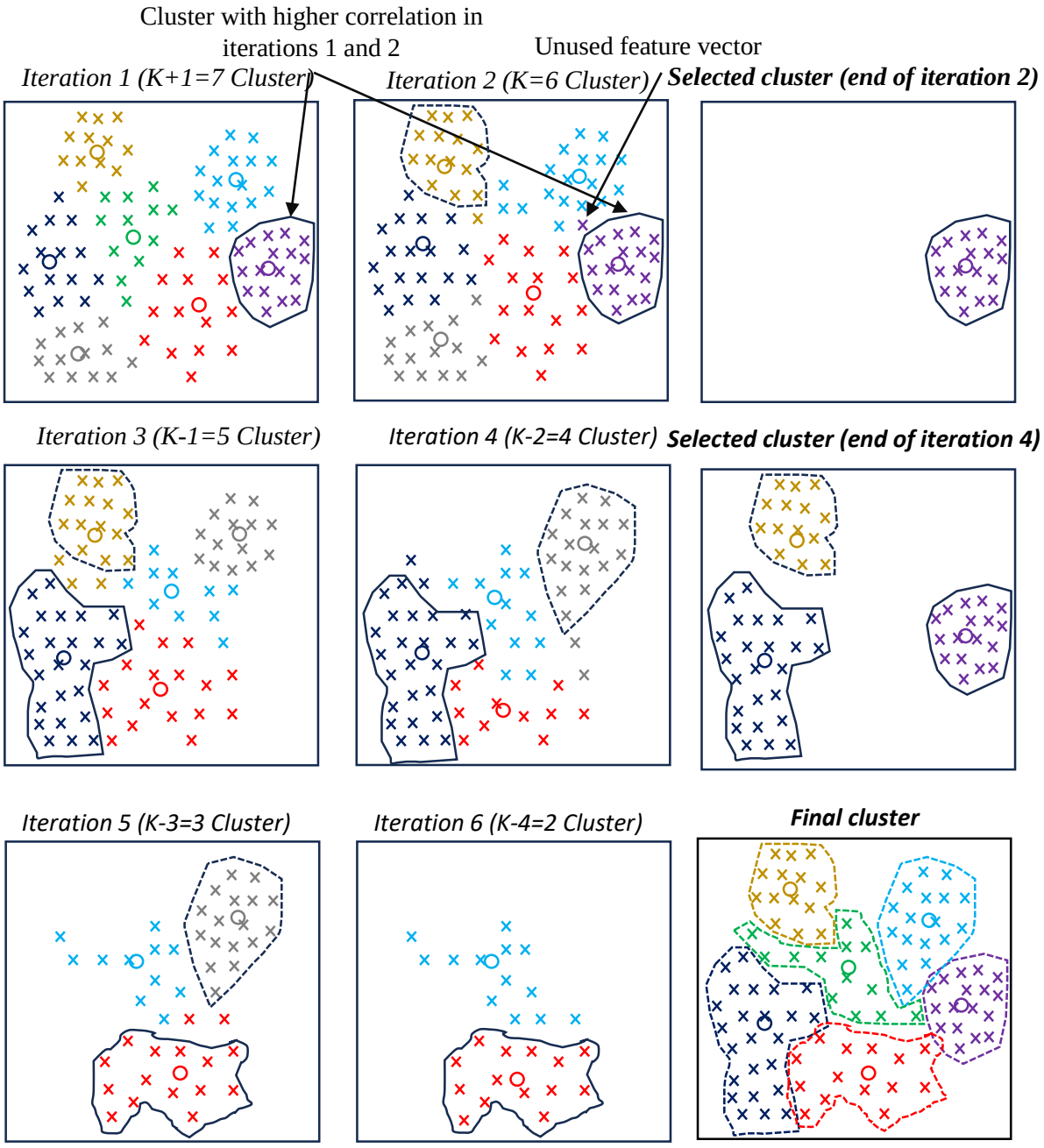



Fig. 3: Representation of recursive tunable clustering process

Here $u_i$ resembles the cluster index in which the feature vector $P_i$ is assigned, while $j$ represent the centroid index. The centroid $e_j$ is updated using the mean feature vector assigned to the corresponding cluster using the relation

$$e_j = \frac{1}{\rho_j} \sum_{P_i \in U_j} P_i \tag{13}$$

Here $U_j$ resembles the feature points belonging to the j[th] cluster, and $\rho_j$ resembles the number of feature vectors clustered in j[th] cluster. Repeat the distance estimation process, clustering, and centroid updation till the stopping condition is attained. The iteration is stopped if the variation between centroids of successive iterations is less than a minimum threshold. The objective function of k-means clustering is to reduce the overall within-cluster variance expressed as,

$$T = \sum_{j=1}^{K} \sum_{P_i \in U_j} \left\| P_i - e_j \right\|^2 \quad (14)$$

Let the K-means algorithm be represented as $(U_j, e_j) \leftarrow k\_means\ (P_i, K)$. The suggested recursive tunable clustering is derived utilizing the k-means clustering in which the proposed approach uses a recursive threshold $\alpha_0$. Instead of performing the clustering process once, the proposed clustering reiterates $\alpha_0$. times. If $K$ be the cluster size, the $K$-means method is reiterated with the number of clusters as steps $K+1, K, k-1, \ldots . K-\alpha_0$. In the first step, the cluster $U_j$, and cluster centroids $e_j$ is computed for $K+1$ cluster. In the second step the cluster $U_j$, and cluster centroids $e_j$ is computed with $k$ clusters. The cluster centers in step $K+1$, and step $K$ that provides a minimum distance variation (high correlation) is computed as,

$$r_j = argmin\ (\|e_{k+1} - e_k\|) \quad k = K, K-1, \ldots . K-\alpha_0 \quad (15)$$

The common cluster elements that correspond to the centers $e_{k+1}$, and $e_k$ is then computed as,

$$C_k = U_{k+1} \cap U_k \quad (16)$$

This $C_k$ is one of the final clusters. The process is repeated for step $K-1, K-2, \ldots . K-\alpha_0$, in which each step computes each cluster of elements $C_k$, and cluster center $e_k$. The clustering process performed in each iteration is illustrated in Fig. 3. Finally, using the elements in the cluster $C_n$ the cluster center is recomputed as $E_n$ using equation (13) by the average of cluster elements. The feature vector that is close to the cluster center in each cluster $C_n$ is chosen as a leader as,

$$q_n = argmin\left(\left\| C_n^{(j)} - E_n \right\|\right) \quad (17)$$

While the remaining elements in the cluster other than the leader are termed as followers. Let the leader and followers of a cluster $n$ be $\lambda_{l,n}$ and $\lambda_{f,n}$ respectively. The algorithm for the proposed recursive tunable K-means clustering is summarized in algorithm 3.

| **Algorithm 3:** | ***Recursive tunable K-means Clustering*** |
|---|---|
| **Input:** | Feature vector $P_i$, Number of clusters $K$, Recursive threshold $\alpha_0$ |
| **Output:** | Cluster leader $\lambda_{l,n}$ and Cluster followers $\lambda_{f,n}$ |
| **1.** | **for** $j = k+1, k, k-1, \ldots . k-\alpha_0$ **do** |
| **2.** | $(U_j, e_j) = k\_means\ (P_i, j)$ |

**3.** **if** $j < K + 1$
**4.** $r_j = argmin\ (\|U_{j+1} - u_j\|)$
**5.** $C_k = (U_{k+1} \cap U_k)$ for cluster index $r_j$
**6.** $C_n \leftarrow C_k$
**7.** Eliminate the feature vector $C_k$, and cluster elements $C_n$ from $P_i$
**8.** $P_i \leftarrow setdiff\ (P_i - C_n)$
**9.** **end if**
**10.** **end for**
**11.** Estimate cluster center $E_n$ using cluster elements $C_n$
**12.** **for** $n = 1$ to $K$ **do**
**13.** Estimate the cluster leaders $\lambda_{l,n}$ and followers $\lambda_{f,n}$ using nearest distance in each cluster using $q_n = argmin\left(\left\|C_n^{(j)} - E_n\right\|\right)$
**14.** **end for**

### 2.4 Siamese twin network

Siamese twin network [27] architecture is a type of neural network structure that has two sub-networks termed as sub-network 1, and sub-network 2 for performing similarity comparison as illustrated in Fig. 4. The two subnetworks share their weights. The principle of the Siamese network is to update the weights in such a way that the generated vector space must be close in the two sub-networks if the two input images are of similar type.

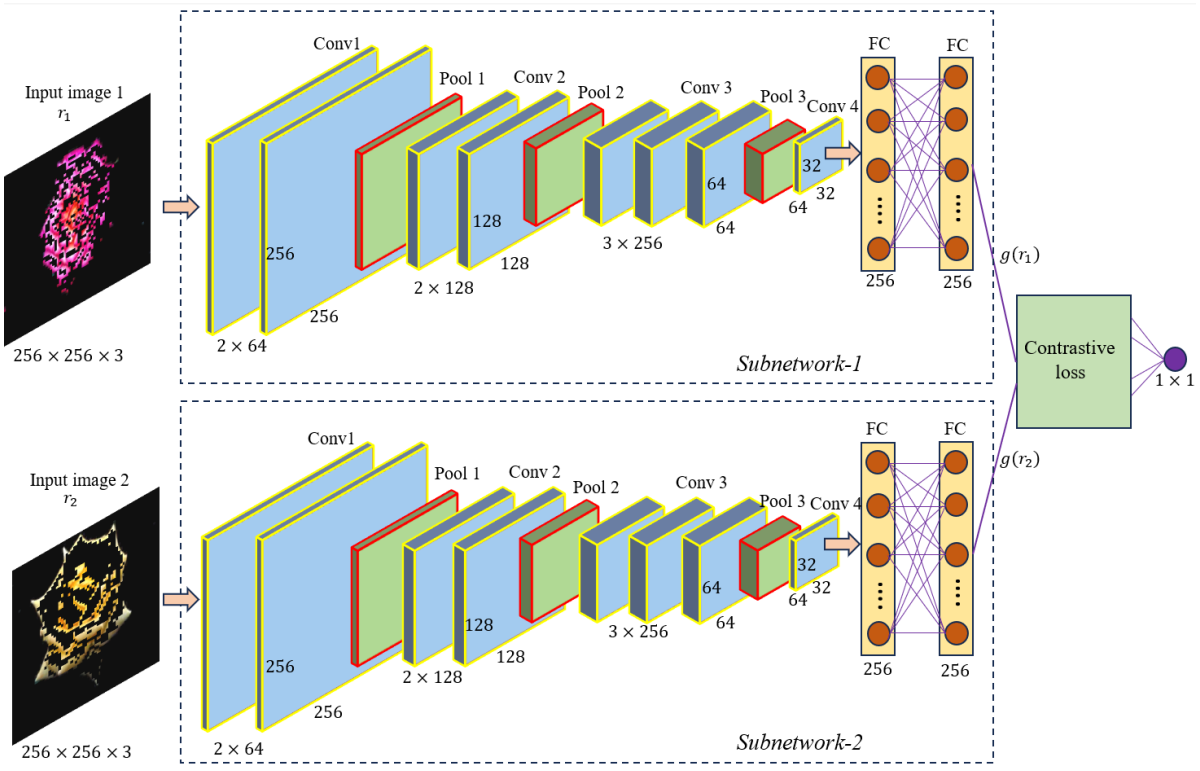



Fig. 4: Structure of Siamese twin network used in proposed CBIR system

Also, the weights are updated to generate a vector space which are far if the two input images belongs to different category. Similar architectures are present in the two sub-networks, with the same number of layers. Each sub-network contains a convolutional, normalization, and pooling layer. Let $r_1$, and $r_2$ represent the input to the sub-networks 1, and 2 respectively. The sub-network 1, and 2 generate the feature vector $g(r_1)$, and $g(r_2)$ respectively. The distance between generated feature vectors $g(r_1)$, and $g(r_2)$ is computed using Euclidean distance as,

$$\alpha = \| g(r_2) - g(r_1) \| \tag{18}$$

The contractive loss function is estimated based on the calculated Euclidean distance as

$$\beta = \frac{\alpha^2}{2}(1-\gamma) + \frac{((\max(0,\Delta-\alpha))^2}{2} \times \gamma \tag{19}$$

Here $\gamma$ represents the class label $\gamma \in [0,1]$. For similar image type $\gamma = 1$, and for different image type $\gamma = 0$. The loss function can be expressed as.

$$\beta = \begin{cases} \frac{\alpha^2}{2} & if\ \gamma = 0 \\ \frac{(\max(0,\Delta-\alpha))^2}{2} & if\ \gamma = 1 \end{cases} \tag{20}$$

$\Delta$ resembles the margin. For similar images, the two networks share similar weights, such that the distance $\alpha$ is reduced. For images of different categories, the network shares different weights, such that distance $\alpha$ is maximized. During the forward pass the distance $\alpha$, and loss $\beta$ is computed. During the backward pass, the gradient is computed with respect to the loss function $\beta$, and weights $\varepsilon$ as,

$$\nabla_{\beta}^{(\varepsilon)} = \frac{\partial \beta}{\partial g(r_1)} \cdot \frac{\partial g(r_1)}{\partial \varepsilon} \cdot \frac{\partial \beta}{\partial g(2)} \cdot \frac{\partial g(r_2)}{\partial \omega} \tag{21}$$

Using the gradient $\nabla_{\beta}^{(\varepsilon)}$, the weight $(\varepsilon)$ is updated by the expression as,

$$\varepsilon(b+1) = \varepsilon(b) - \tau \nabla_{\beta}^{(\varepsilon)} \tag{22}$$

$b$ resemble the iteration number, and $\tau$ resembles the learning rate. During the testing phase, a similarity score is generated by the Siamese network from which the two input images $r_1$, and $r_2$ can be identified as similar or different.

### 2.5 Retrieval process

The cosine similarity between any two-feature vector $P_i$, and $P_j$ can be estimated as,

$$S_{ij} = \frac{P_i \cdot P_j}{\|P_i\| \|P_j\|} \tag{23}$$

The cosine similarity $S_{ij}$ close to $-1$ resembles that the feature vector $P_i$, and $P_j$ are very dissimilar, while cosine similarity $S_{ij}$ close to 1 resembles $P_i$, and $P_j$are more similar. The recursive tunable K-means clustering approach selects $K$ leaders from $G$ data points $P_1, P_2, \dots \dots P_G$. The proposed approach uses the trained Siamese network (either subnetwork 1 or subnetwork 2) to extract the feature (output of subnetwork). As the picture owner uploads the encrypted picture, the cloud server will detect the high-level components and mask the low-level region. The resultant image is used to extract the features with the use of a trained subnetwork present in the Siamese twin network. These features are represented as $P_1, P_2, \dots \dots P_G$. The recursive tunable K-means clustering algorithm will cluster the feature vectors to $K$ clusters, and store the feature vector along with the cluster leader $\lambda_{l,n}$, and follower $\lambda_{f,n}$ information. During the retrieval process, the cloud will detect the high-level components from the encrypted query picture, and mask the low-level region. The resultant image is utilized to collect the query image descriptors with the use of a trained subnetwork present in the Siamese twin network. The query picture descriptors are matched with the descriptor vector of the $K$ cluster leaders. The cluster leader that was closely matched with the query feature vector is chosen as the retrieval cluster. The image corresponding to the cluster leader, and followers of the selected retrieval cluster is then matched using the Siamese twin network, and $\widehat{K}$ images that provide the highest similarity score (cosine similarity using equation (23) of Siamese twin network) are retrieved to the user for decryption.

## 3. EXPERIMENTAL RESULTS

The datasets namely Corel-10k [28] and Inria Holiday [29] are utilized for the evaluation of the suggested scheme. The Corel 10k dataset has 10,000 images which includes images of vehicles, paintings, nature, objects, etc. as illustrated in Fig. 5(a). The Inria holiday dataset has 500 query images and a total of 1491 images that are acquired using mobile cameras from different places in different countries. This dataset contains images of nature such as flowers and water, man-made buildings, shops, vehicles, etc with different dimensions as illustrated in Fig. 5(b). The evaluation of high-level region segmentation algorithm was evaluated using scales such as precision (Pr), recall (Re), F1-score (F1), accuracy (Ac), and specificity (Sp) with the following relations.

$$Precision\ (Pr) = \frac{\delta_{tp}}{\delta_{tp}+\delta_{fp}} \tag{24}$$

$$Recall\ (Re) = \frac{\delta_{tp}}{\delta_{tp}+\delta_{fn}} \tag{25}$$

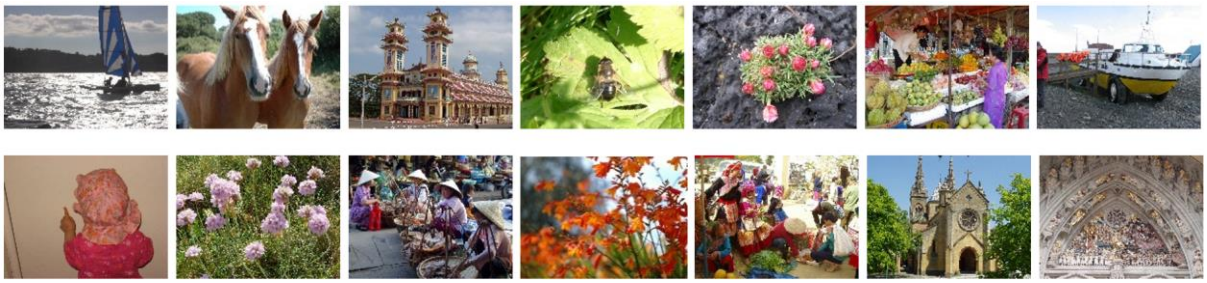
$$F1-score\ (F1) = \frac{2\times Re\times Pr}{Re+Pr} \tag{26}$$

$$Accuracy\ (Ac) = \frac{\delta_{tn}+\delta_{tp}}{\delta_{tp}+\delta_{fp}+\delta_{tn}+\delta_{fn}} \tag{27}$$

$$Specificity\ (Sp) = \frac{\delta_{tn}}{\delta_{tn}+\delta_{fp}} \tag{28}$$

Here $\delta_{fp}$, $\delta_{fn}$, $\delta_{tn}$, and $\delta_{tp}$ resembles false positive, false negative, true negative, and true positive results obtained by the modified SegNet segmentation.

(a)



(b)

Fig.5: Sample images used for analysis (a) Corel10k dataset (b) Inria holiday dataset

The performance of the suggested CBIR system was analyzed utilizing the measures such as time complexity, and mean average precision (mAP). In the case of the Corel-10k dataset, the images are not augmented, thus 7000 images are utilized for training the modified SegNet, and Siamese twin network. The same 7000 images are stored in the cloud for the retrieval process. The remaining 3000 images are used to test the modified SegNet, and proposed CBIR system (3000 query images). In the case of the Inria Holiday database, the images are augmented by modifications such as rotation by 90°, 180°, 270°, brightening by 50, and darkening by 50. Thus the Inria dataset after augmentation has 8946 images. In this 6262 images are utilized for training the modified SegNet, and Siamese twin network. The same 6262 images are stored in the cloud for the retrieval process. The remaining 2684 images are used to test the modified SegNet, and proposed CBIR system (2684 query images).

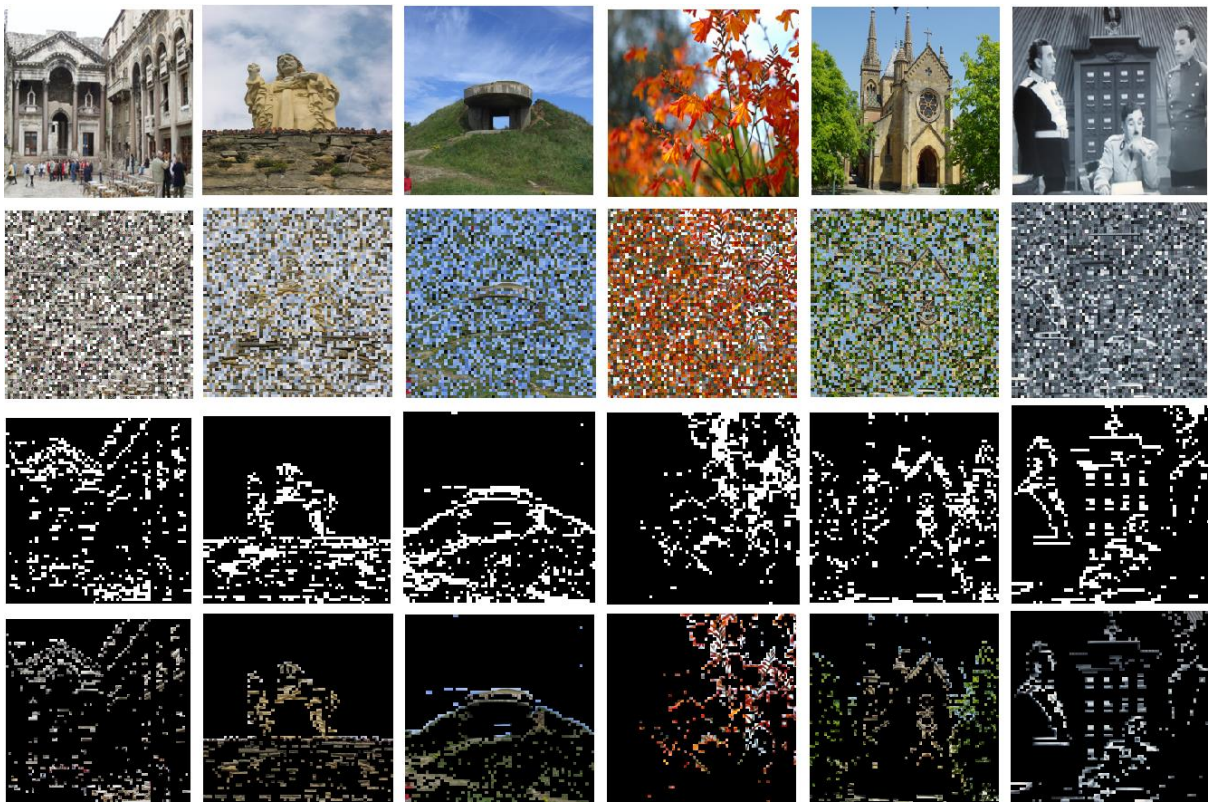


Fig.6: Segmentation result provided by the SegNet architecture (row1) Original image (row2) Encrypted image (row3) Segmented high-level and low-level region (1's represents the high-level region, while 0's represents the low-level region) (row4) Segmented high-level region.

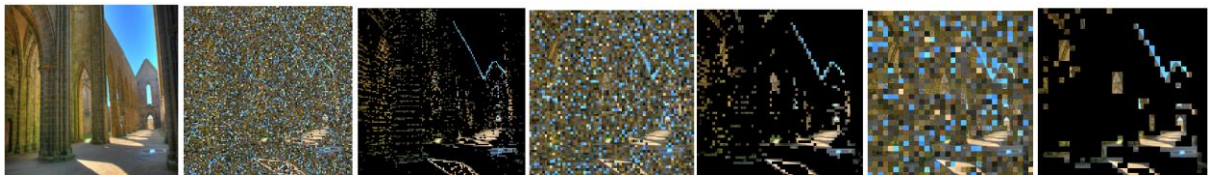
Fig. 6 depicts the segmentation result obtained by the proposed SegNet architecture. The proposed SegNet architecture can able to detect the High-level components that are utilized to extract the features. The encryption was performed with a block size of $L_1 \times L_2 = 4 \times 4$ by utilizing 85% of blocks as low-level blocks i.e. $F = 0.85B$. The high-level components detected by SWO-HLC are used as foreground ground truth, while the low-level components are used as background ground truth to train the modified SegNet. The SegNet uses a learning rate of $\tau = 10^{-4}$. The query image high-level features are used to match the features with the high-level descriptors of the leaders of the clusters selected by the recursive tunable K-means clustering algorithm.

Table I: Segmentation result obtained by the proposed SegNet architecture in separating the High-level region

| Dataset | Ac (%) | Re (%) | Pr (%) | Sp (%) | F1 (%) |
|---|---|---|---|---|---|
| **Corel-10k** | 98.14 | 96.74 | 97.63 | 98.05 | 97.53 |
| **Inria Holiday** | 97.26 | 96.17 | 96.89 | 97.33 | 96.71 |

Table I depicts the comparison of segmentation performance obtained by the proposed SegNet architecture when evaluated using the Corel-10k and Inria Holiday datasets. The suggested SegNet architecture results in a segmentation accuracy of 98.14% and 97.26% when evaluated using Corel-10k

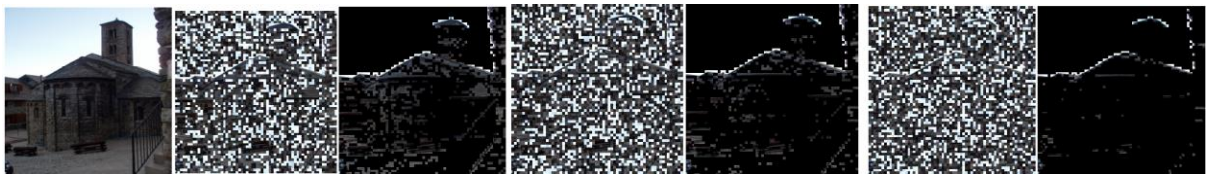
and Inria Holiday datasets respectively. The segmentation performance obtained on the Corel-10k dataset is higher than the performance obtained in the Inria Holiday dataset. The proposed segmentation results in an average accuracy, recall, precision, specificity, and F1-score of 97.7%, 96.45%, 97.26%, 97.69%, and 97.12% respectively.



(a) (b) (c) (d)

Fig.7: Segmentation result provided by the proposed SegNet approach for different values of block size $L = L_1 = L_2$ (a) Plain image (b) Cipher image and segmented image with $L = 2$ (c) Cipher image and segmented image with $L = 4$ and (d) Cipher image and segmented image with $L = 8$

Fig. 7 provides the segmentation results obtained by the suggested modified adaptive SegNet architecture for different values of block size $L = 2$, $L = 4$ and $L = 8$. The proposed SegNet approach can able to segment the high-level blocks into both lower and higher values of $L$. The block size $L = 2$, preserves more privacy content, while increasing the complexity. Conversely the higher block size, reduces the complexity, while exposes the image content. Thus, it is preferred to use the modified SegNet architecture with a block size of $L = 4$.



(a) (b) (c) (d)

Fig.8: Segmentation result provided by the suggested approach for various values of low-level threshold $F$ (a) Plain image (b) Cipher and segmented image with $F = 0.75B$ (c) Cipher and segmented image with $F = 0.85B$ and (d) Cipher and segmented image with $F = 0.95B$

Fig. 8 depicts the segmentation result obtained by the modified SegNet approach for various values of threshold $F$. Using a low-level threshold $F$, exposes more content of the image than the higher level threshold. However, using a higher-level threshold exposes less content of the image, while reducing the effectiveness of descriptors collected from the picture, while reducing the performance of the CBIR system.

Table II: Performance variation of mAP (%) with respect to the recursive threshold $\alpha_0$

| | **Recursive threshold $\alpha_0$** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| **Corel-10k** | 54.97 | 58.36 | 61.73 | 64.75 | 66.43 | 67.98 | 68.76 | 69.27 | 68.96 | 68.67 |
| **Inria dataset** | 50.57 | 54.76 | 58.12 | 61.06 | 62.91 | 63.46 | 64.12 | 64.53 | 64.38 | 64.01 |

Table II shows the variation of mAP performance for different values of the recursive threshold $\alpha_0$. The experiment was done with a number of clusters $K = 26$. As the recursive threshold is varied from $\alpha_0 = 2$ to $\alpha_0 = 20$, the mAP increases and attains a maximum at $\alpha_0 = 16$. For further increases in the recursive threshold, the mAP gradually reduces. Thus, the proposed CBIR system provides a maximum performance with a recursive threshold of $\alpha_0 = 16$.

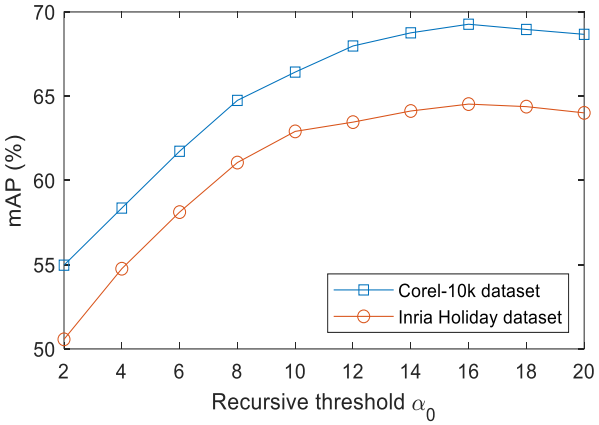



Fig.9: Variation of mAP (%) for different values of recursive threshold

The graphical comparison between mAP for various recursive threshold $\alpha_0$ is plotted in Fig. 9, where the CBIR system shows a maximum performance with $\alpha_0 = 16$. The proposed CBIR system was implemented utilizing MATLAB 2023b installed on a PC with 64GB RAM, an I7 processor @ 3.2GHz, and a Windows 10 operating system.

Table III: Comparison of mAP (%) between the proposed CBIR system and similar approaches

| **Dataset** | **IES [30]** | **Partial Encryption [31]** | **Attention Networks [32]** | **BOEW [8]** | **EViT [13]** | **SWO-HLC [b6]** | **Proposed** |
|---|---|---|---|---|---|---|---|
| **Corel-10k** | 54.56 | 56.04 | 62.51 | 64.24 | 65.73 | 67.95 | 69.27 |
| **Inria dataset** | 48.34 | 50.13 | 56.96 | 58.81 | 59.19 | 61.42 | 64.53 |

The performance of the suggested CBIR system was compared with similar CBIR approaches namely IES [30], Partial encryption [31], attention network [32], BOEW [8], EViT [13], and SWO-HLC [24]

when evaluated using the Corel-10k and Inria dataset. The proposed approach resulted in an mAP of 69.27% and 64.53% when evaluated using the Corel-10k and Inria datasets. The proposed approach results in increase in mAP of 1.32% and 3.11% than the SWO-HLC approach when evaluated using the Corel-10k and Inria datasets respectively. Also, the proposed approach results in an increase in mAP of 3.54% and 5.34% than the EViT scheme when evaluated using the Corel-10k and Inria datasets respectively as illustrated in Table III.

Table IV: Computational complexity during the retrieval process

| Dataset | Time consumption (in seconds) | | | | |
|---|---|---|---|---|---|
| | $t_e$ | $t_h$ | $t_f$ | $t_s$ | $t_{tot}$ |
| **Corel-10k** | 0.24 | 0.36 | 0.392 | 1.176 | 2.168 |
| **Inria Holiday** | 0.26 | 0.38 | 0.413 | 1.15 | 2.203 |

Let, $t_e$, $t_h$, $t_f$, and $t_s$ resembles the time of encryption, high-level region detection, feature extraction, and query search respectively. Thus, the total time utilized during the retrieval is represented as

$$t_{tot} = t_e + t_h + t_f + t_s \tag{29}$$

The comparison of time utilization at different stages such as encryption, high-level region detection, feature extraction, and query search when evaluated using Corel-10K, and the Inria Holiday dataset is illustrated in Table IV. The time of encryption, high-level region detection, feature extraction, and query search in the Corel-10k dataset was estimated as 0.24s, 0.36s, 0.392s, and 1.176s respectively totally utilizing 2.168s to retrieve $\widehat{K} = 100$ images. Similarly, the retrieval time in the Inria Holiday dataset was estimated as 2.203 which is slightly higher than the retrieval time estimated on the Corel-10k dataset.

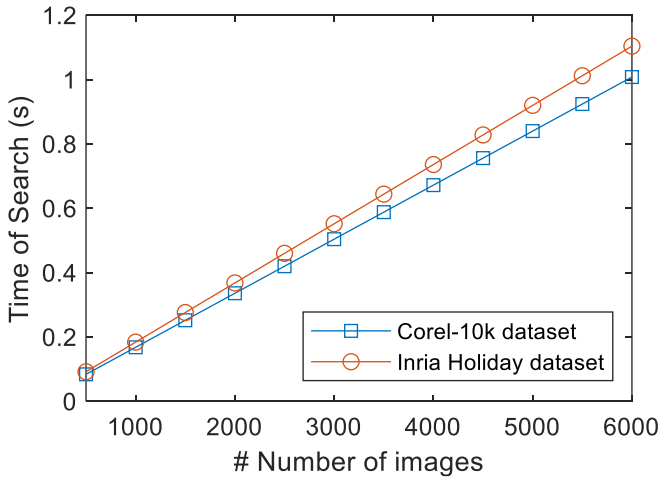



Fig.10: Comparison of time of search for the suggested approach for two datasets

Fig. 10 shows the time of search comparison for the proposed approach with different numbers of images in the database. As the database size increases, the search time also rises linearly. This is due to the increase in number of images in each cluster of the recursive tuned K-means algorithm.

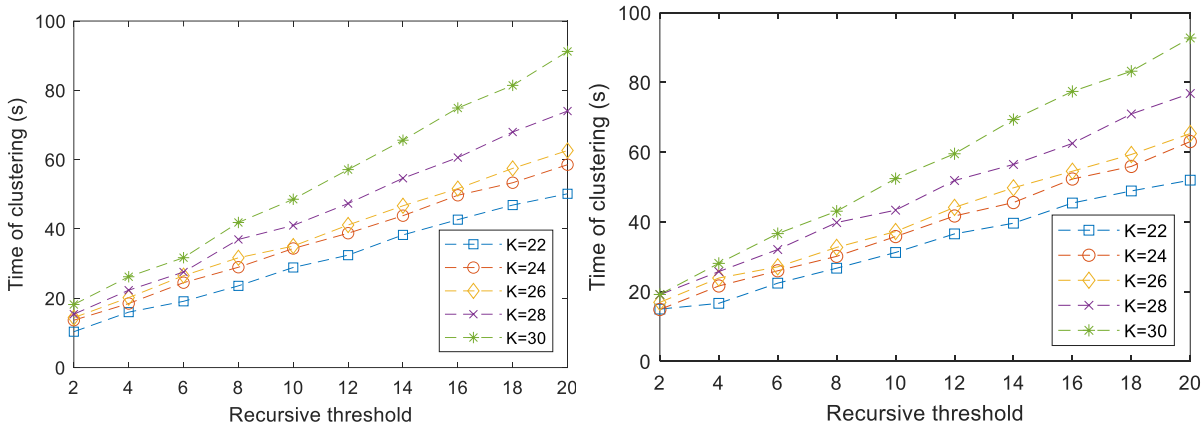



Fig.11: Comparison of time of clustering for the suggested approach for different recursive thresholds and number of clusters (a) Corel-10k dataset (b) Inria Holiday dataset

Fig. 11 shows the time of clustering utilized by the proposed recursive tunable K-means clustering for various values of recursive threshold $\alpha_0$ and the cluster size $K$. The experiment was done with various values of $K$ such as $K = 22,24,26,28$ and 30. The recursive threshold was varied between $\alpha_0 = 2$ to 20. As the recursive threshold increases the number of stages in the clustering process rises, which increases the time complexity of the clustering process. For a recursive threshold, an increase in number of cluster $K$ also increases the complexity and this is due to the increase in the number of clusters. The proposed approach yields a maximum performance with $\alpha_0 = 16$ and $K = 26$.

## 4. CONCLUSION

This work proposed a cloud-CBIR system using deep learning architectures such as Siamese twin network and SegNet architectures which can preserve the privacy of the picture. This approach uses the SWO-HLC block permutation approach to encrypt the low-level component of the image while leaving the high-level components. The SWO-HLC encrypted image which is uploaded to the cloud is separated into two components namely low and high-level regions using a trained SegNet architecture. The detected low-level region is then masked and the resultant high-level region is utilized to extract the features. The query picture which was uploaded to the cloud also undergoes a similar process and the high-level region which was detected by the SegNet architectures is used to extract the features. The matching was done using a Siamese twin network between the high-level regions of query and dataset images. To reduce the time of search a clustering approach based on a recursive tunable K-means clustering approach is proposed. Two datasets namely Corel-10k and Inria Holiday dataset are utilized

to evaluate the suggested retrieval system. The proposed CBIR structure results in a mAP of 69.27% and 64.53% when evaluated using the Corel-10k and Inria Holiday datasets respectively. The evaluation result in terms of mAP and time complexity emphasizes that the suggested approach can be utilized in different cloud CBIR platforms.

**References**

[1]. Hameed, I. M., Abdulhussain, S. H., & Mahmmod, B. M. (2021). Content-based image retrieval: A review of recent trends. Cogent Engineering, 8(1), 1927469.

[2]. Noor, J., Shanto, M. N. H., Mondal, J. J., Hossain, M. G., Chellappan, S., & Al Islam, A. A. (2022). Orchestrating image retrieval and storage over a cloud system. IEEE Transactions on Cloud Computing, 11(2), 1794-1806.

[3]. Shen, M., Cheng, G., Zhu, L., Du, X., & Hu, J. (2020). Content-based multi-source encrypted image retrieval in clouds with privacy preservation. Future Generation Computer Systems, 109, 621-632.

[4]. Xu, S., Horng, J. H., Chang, C. C., & Chang, C. C. (2022). Reversible data hiding with hierarchical block variable length coding for cloud security. IEEE transactions on dependable and secure computing, 20(5), 4199-4213.

[5]. Shijin, K. P., & Edwin, D. D. (2012, March). Simulated attack based feature region selection for efficient digital image watermarking. In 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET) (pp. 1128-1132). IEEE.

[6]. Shen, W., Qin, J., Yu, J., Hao, R., & Hu, J. (2018). Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage. IEEE Transactions on Information Forensics and Security, 14(2), 331-346.

[7]. Xia, Z., Wang, L., Tang, J., Xiong, N. N., & Weng, J. (2020). A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing. IEEE Transactions on Network Science and Engineering, 8(1), 318-330.

[8]. Xia, Z., Jiang, L., Liu, D., Lu, L., & Jeon, B. (2019). BOEW: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing. IEEE Transactions on Services Computing, 15(1), 202-214.

[9]. Anju, J., & Shreelekshmi, R. (2022). A faster secure content-based image retrieval using clustering for cloud. Expert Systems with Applications, 189, 116070.

[10]. Wang, Yu, Liquan Chen, Ge Wu, Kunliang Yu, and Tianyu Lu. "Efficient and secure content-based image retrieval with deep neural networks in the mobile cloud computing." Computers & Security 128 (2023): 103163.

[11]. Krishnan, S. H., Vishwa, C., Suchetha, M., Raman, A., Raman, R., Sehastrajit, S., & Dhas, D. E. (2023). Comparative performance of deep learning architectures in classification of diabetic retinopathy. International Journal of Ad Hoc and Ubiquitous Computing, 44(1), 23-35.

[12]. Sathvika, V. B. T., Anmisha, N., Thanmayi, V., Suchetha, M., Dhas, D. E., Sehastrajit, S., & Aakur, S. N. (2024). Pipelined Structure in the Classification of Skin Lesions Based on Alexnet CNN and SVM Model With Bi-Sectional Texture Features. IEEE Access.

[13] Feng, Q., Li, P., Lu, Z., Li, C., Wang, Z., Liu, Z., ... & Weng, J. (2024). Evit: Privacy-preserving image retrieval via encrypted vision transformer in cloud computing. IEEE Transactions on Circuits and Systems for Video Technology.

[14] Mao, C., Shen, Z., Chen, K., Liu, Y., Meng, Q., & Wang, F. (2024). DCIRM: Dynamic and Controllable Image Retrieval Scheme in Multi-owner Multi-user Settings. IEEE Transactions on Services Computing.

[15] Khan, S., Abbas, H., & Iqbal, W. (2024). Verifiable Privacy-Preserving Image Retrieval in Multi-Owner Multi-User Settings. IEEE Transactions on Emerging Topics in Computational Intelligence.

[16] Ma, W., Zhou, T., Qin, J., Xiang, X., Tan, Y., & Cai, Z. (2022). A privacy-preserving content-based image retrieval method based on deep learning in cloud computing. Expert Systems with Applications, 203, 117508.

[17] Qin, J., Chen, J., Xiang, X., Tan, Y., Ma, W., & Wang, J. (2020). A privacy-preserving image retrieval method based on deep learning and adaptive weighted fusion. Journal of Real-Time Image Processing, 17(1), 161-173.

[18] Rahim, N., Ahmad, J., Muhammad, K., Sangaiah, A. K., & Baik, S. W. (2018). Privacy-preserving image retrieval for mobile devices with deep features on the cloud. Computer Communications, 127, 75-85.

[19] Li, S., Wu, L., Meng, W., Xu, Z., Qin, C., & Wang, H. (2022). DVPPIR: privacy-preserving image retrieval based on DCNN and VHE. Neural Computing and Applications, 34(17), 14355-14371.

[20] Lu, Z., Feng, Q., Li, P., Lo, K. T., & Huang, F. (2023). A privacy-preserving image retrieval scheme based on 16× 16 DCT and deep learning. IEEE Transactions on Cloud Computing, 11(3), 3314-3325.

[21] Zhang, C., Zhu, L., Zhang, S., & Yu, W. (2020). TDHPPIR: An efficient deep hashing based privacy-preserving image retrieval method. Neurocomputing, 406, 386-398.

[22] Cheng, S. L., Wang, L. J., Huang, G., & Du, A. Y. (2021). A privacy-preserving image retrieval scheme based secure kNN, DNA coding and deep hashing. Multimedia Tools and Applications, 80, 22733-22755.

[23] Punithavathi, R., Ramalingam, A., Kurangi, C., Reddy, A. S. K., & Uthayakumar, J. (2021). Secure content based image retrieval system using deep learning with multi share creation scheme in cloud environment. Multimedia Tools and Applications, 80(17), 26889-26910.

[24]. Selvapattu, J. S. (2024). Bi-level feature Classification Approach in Privacy Preserving Cloud Network for High-Performance Content-based Image retrieval Mechanism. Communications on Applied Nonlinear Analysis, 31(7s), 138-156.

[25]. Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), 2481-2495.

[26]. Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. IEEE access, 8, 80716-80727.

[27]. Bao, H., Shu, P., Zhang, H., & Liu, X. (2022). Siamese-based twin attention network for visual tracking. IEEE Transactions on Circuits and Systems for Video Technology, 33(2), 847-860.

[28]. J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 9, pp. 1075–1088, Sep. 2003.

[29]. 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.

[30]. 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.

[31]. 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

[32]. Feng, Q., Li, P., Lu, Z., Zhou, Z., Wu, Y., Weng, J., & Huang, F. (2023). DHAN: Encrypted JPEG image retrieval via DCT histograms-based attention networks. Applied Soft Computing, 133, 109935.