Received: June 17, 2023. Revised: August 21, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Securing Privacy: Encrypted Image Retrieval with CNNs and Chaos-Based Visual Cryptography on Cloud Computing

Ziyad Tariq Mustafa Al-Ta'i[1] **Shaima Miteb Sadoon**[1]*

[1] *Department of Computer Science, Science College, Diyala University, Baquba, Iraq*
* Corresponding author's Email: scicompms2212@uodiyala.edu.iq

Abstract: Data security and accessibility in a cloud computing environment are successfully guaranteed by the safe privacy content-based image retrieval (CBIR) approach. However, ineffective cipher text methods and feature-extraction techniques might reduce performance. Therefore, in order to create secure privacy CBIR, it is still necessary to address the need for high-security encryption techniques and the capacity to extract characteristics from cipher text images. To achieve this, we provide a secure privacy image retrieval method based on convolutional neural network features. Initially, a 3D Lotka_Volterra chaotic systems encryption method based on visual cryptography encodes images, and the updated DenseNet-121 model is fine-tuned by employing the encrypted images to construct a feature extractor. There are multi-authorized users set up for the fine-tuning feature extractor. The encrypted features and images are uploaded to a cloud server. Experiments on the Corel10k dataset showed the effectiveness of the proposed model in terms of security and precision compared to previous works, achieving an average search precision of 68.94% and an MSE =12764.27263, PSNR = 7.300480125 dB between the original image and encrypted, and the decrypted image is entirely identical to the original image.

Keywords: Deep learning, Visual cryptography, Lotka_Volterra, Chaotic system, CBIR.

1. Introduction

Daily, innumerable images are taken thanks to the explosive rise of multimedia gadgets. Image retrieval is an attractive technique that enables us to find the needed images rapidly. Image retrieval relies heavily on content-based image retrieval (CBIR). Based on the visual content of a query image, CBIR searches an image archive for related images. This technique has applications in facial recognition [1], online shopping [2], and remote diagnostics [3]. Most data owners use public clouds to store their large image databases and service the image search. due to the high expenses of storage and upkeep. The cloud will then make comparable photos accessible to authorized users. Public clouds lower image owners' costs and raise security concerns [4]. Sensitive information may be present in outsourced photo datasets, making them vulnerable to theft by competitors, hackers, or cloud providers if they are directly outsourced to unreliable cloud servers. Although conventional cryptosystems can stop the leakage of private data, public clouds can no longer offer services image search. Managing massive images securely and effectively is still the most significant issue for CBIR technology.

Feature-encryption CBIR is one of the traditional CBIR techniques that protects privacy by extracting features from unencrypted images and encrypting them. For storage and maintenance, owners upload encrypted features and encrypted images to a cloud server[5, 6]. However, these systems may not scale well for maintaining CBIR services for large image databases and impose significant computational costs on "weak" image owners.

In order to enhance the performance of local computing, CBIR-based image-encryption methods that assign computing duties to the cloud server have been put forth by numerous authors. [7-12], while[7] and [12] use MPEG-7 descriptors, [8]

DOI: 10.22266/ijies2023.1231.17

recommends using hue-saturation value (HSV) descriptors to describe image attributes. It is crucial to emphasize that the majority of lower-level features only extract local data and are unable to capture the semantic correlations required for secure CBIR methods.

High-level semantic features and fused features have gradually replaced CBIR based on low-level features[13-16]. Since the introduction of deep convolutional neural networks (CNN). DenseNet outperforms VGG , GoogLeNet, ResNet , and AlexNet regarding feature use, feature expression, parameters, and computation complexity because of it uses templates from convolution, types of pooling, and other techniques to infer human perception. [16, 17]. On the other hand, creating a specific architecture for a reliable and secure CBIR service is necessary to grant complex semantic elements the property of privacy preservation.

Based on the analysis above, we propose a CNN-based secure image retrieval method that focuses on extracting high-level features. We offer a visual cryptography method based on three-dimension Lotka-Volterra chaos maps to ensure the safety of pictures kept on unreliable cloud services and prevent data breaches. We also offer an image user service that enables the extraction of semantic characteristics from encrypted images using an updated DenseNet-121 model.

To prove the effectiveness of our proposed architecture, our main contribution entails conducting a complete security study and carrying out rigorous assessments. Compared to conventional encryption techniques, this model, which uses visual encryption technology (n,n), provides a higher level of safety for picture retrieval.

CBIR cloud solutions are covered in this section. To solve the problems in the system mentioned above, we suggest a CNN-based, privacy-preserving CBIR framework. Section 2 quickly surveys pertinent literature, the proposed method is fully described in section 3, methodology and strategy. In section 4, coupled with a thorough security analysis, we offer the experimental findings from our study. Section 5 brings our investigation to a close by summarizing the significant conclusions and outlining our next steps.

2. Literature review

The term "content-based image retrieval" (CBIR) refers to extracting the required data from vast collections of multimedia data based on the image's content. In 2021, J. Tang et al. They created a collection of encryption techniques that were JPEG format compatible and did not require JPEG files to be expanded. They first created big blocks by combining many nearby eight-by-eight discrete cosine transform (DCT) coefficient blocks. Then, to safeguard the privacy of the JPEG image, random scrambling and stream encryption are applied to the binary code of DCT coefficients. The cloud server fetches related images and extracts features from encrypted images. The encrypted big blocks extract the group index histograms of the DCT coefficients, and then the bag-of-words (BOW) model is used to create the global vector to represent the JPEG image [18]. It suffers from low-level features for image encryption. In 2022, J. Anju and R. Shreelekshmi, the image owner gathers all the images into clusters and extracts MPEG-7 visual descriptors for indexing. Asymmetric scalar-product preservation (ASPE) encrypts these cluster centres and image characteristics [7]. The problem of safe CBIR, known as the "semantic gap for image-encryption feature," persists between low-level and high-level features despite the extensive efforts put into privacy-preserving CBIR. In 2022, W. Ma et al. They suggested an image retrieval method that protects privacy and is based on deep convolutional network characteristics. Images are first encrypted using a hybrid encryption technique, and then the encrypted images are used to build a feature extractor that is then used to improve the DenseNet model. The fine-tuning feature extractor and encrypted images are then uploaded to a cloud server. The cloud server is currently running a secure CBIR service [19]. Although this proposal achieved security and accuracy in retrieval, it failed in some aspects of the encryption process, including image retrieval after decoding, as the images suffered from data loss during the encryption process.

3. Method terminology

3.1 The proposed CBIR model

The model proposed composes of three entities the image users, cloud server, and image owner as demonstrated in the Fig. 1.

Image owner: An image owner would like to transfer his encrypted local data, which consists of a set of images E_n , to a cloud server while still being able to search through. An image owner encrypts the images and extracts the feature vectors from E_n to create a secure, searchable Index I on features. After that, a cloud stores index I and encrypted image collection E_n Additionally, image owners are responsible for securely authorizing image users.

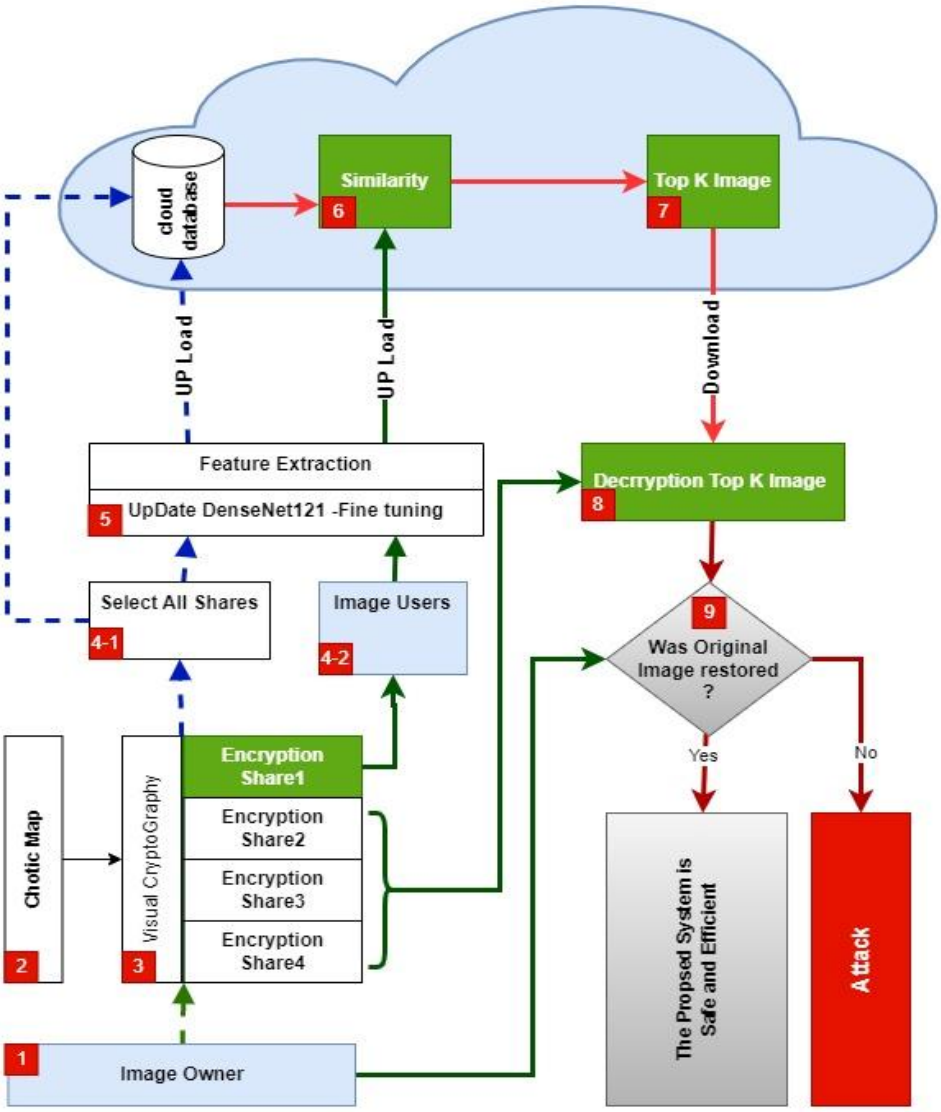


Figure. 1 The system model

An image owner encrypts the images using visual cryptography based on Lotka Volterra chaos map to produce four shares of each image. The four shares and their features are stored on the cloud. Then, it sends the model of feature extraction using CNN to an image users.

Image users: Image user first encrypts the query image into four shares and extracts features using the CNN model, then send the extracted features and share1 to the cloud server. The top K matching results are retrieved by cloud server to the image user. Finally, the image user decrypts the retrieved image as output.

Cloud server: Cloud servers process user query requests and store the image owner's index I. A cloud server correctly complies with protocol specifications, but it will analyze and keep communication history to get sensitive information. Therefore, the privacy of image features and image content must be safeguarded.

Table 1. Notations utilized in our scheme

Notation	Description
E_n	encrypted images dataset
dt	Cotrol parameter
s	Length of generated sequence
$matrix - Share$	Matrices of shares image
$ImgShare$	Share of Image
h	Height of matrix
w	Width of matrix
e_q	encrypted image query
f_q	the feature vector of a query image
f_i	the feature vector of an encrypted image in a cloud

3.2 Notations in this section

In Table 1, the notations utilized in our scheme

```
Algorithm 1: Lotka–Volterra equations
Input: x1,y1,z1, i=1, s
Output: x,y,z
Begin
1: $dt$ = 0.01
2: apply Eqs. (1),(2) and (3).
x(i) = x1*(x1* (1 –x1 - 9 * y1)) * $dt$
y(i) =y1*(-y1 * (1 - 6 * x1 – y1 + 9* z1)) * $dt$
z(i) = z1*(z1 * (1 - 3 * x1 – z1)) * $dt$
3: Swap
        x1=x(i) , y1=y(i) ,   z1=z(i)
4:increase i by one
5:while(i<s) goto step2
6: return x,y,z
End
```

```
Algorithm 2: Shares generation
Input: h, w
Output: $matrix - Share1$, $matrix - Share2$,
$matrix - Share3$, $matrix - Share4$
Begin
1:Determine size of the $matrix - Share$
1-1:Determine the $matrix - Share1$ size to be
generated [h,w]
1-2: Determine the $matrix - Share2$ size to be
generated [h,w]
1-3: Determine the $matrix - Share3$ size to be
generated [h,w]
1-4: Determine the $matrix - Share4$ size to be
generated [h,w]
2: Select an initial value "x1, y1,z1” is chosen to
be any value range 0 and 1..
3: increment i and j by one
4: Compute Lotka–Volterra algorithm to
"x1,y1 ,z1" to obtain the next values in the series.
4-1: byte convert x_byte=$\lfloor x1*10^3 \rfloor$ mod 255 Set
value
    $matrix - Share1$ [i][j]= byte
4-2: byte convert x_byte=$\lfloor y1 * 10^3 \rfloor \, mod \, 255$
Set value $matrix - Share2$ [i][j]= x_byte

4-3: byte convert x_byte= $\lfloor z1 * 10^3 \rfloor \, mod \, 255$
Set value $matrix - Share3$[i][j]= byte

5: Compute Lotka–Volterra algorithm to "x1,y1,
z1" to obtain the next values in the series.

5-1: byte convert x_byte=$\lfloor x1*10^3 \rfloor$ mod 255
            Set value $matrix - Share$ 4[i][j]=
   byte
6:Repeat steps 3-5 with where i < h and j < w.
7: Return $matrix - Share1$, $matrix - Share2$,
$matrix - Share3$, $matrix - Share4$.
End
```

are described.

3.3 Image encryption

Visual cryptography depends on the three-dimension Lotka_Volterra chaos system to obtain an effective image encryption technique. The following equations represent the formula for three dimensions Lotka_Volterra chaos maps used to create four image shares:

$$dx/_{dt} = x * (1 - x - 9 * y) \tag{1}$$

$$dy/_{dt} = -y * (1 - 6 * x - y + 9 * z) \tag{2}$$

$$dz/_{dt} = z * (1 - 3 * x - z) \tag{3}$$

Algorithm 1 demonstrates the three-dimension Lotka_Volterra chaos system. The chaotic sequence created by the 3-Dimension Lotka_Volterra chaos map generates matrix_shares. Algorithm 2 demonstrates the shares creation algorithm.

The values of chaotic sequences float by the Algorithm 1, To take the integer part for this algorithm, it is multiplied by 10^3. The integer part is then modified by mod 255 to fit in the range of (0 - 255). To be used later in a process Xor in the encryption Algorithm 3. The four share matrices produced by an Algorithm 2 are XORed with the original image to produce the four image shares. This is clarified by the share cipher algorithm in the Algorithm 3.

3.4 Feature extraction

DenseNet (DenseNet-121) is a CNN architecture that uses "Dense Blocks," which are direct connections between layers. Fine-tuning is a way to apply or utilize transfer learning. The fine-tuning process involves tuning a trained model to perform a second similar task after it has been trained for the first task. The pre-trained Densenet-121 is imported with weights of the imagenet dataset and excluded the Dense layer of the original pre-trained Densenet-121. Fine-tuning is a method for choosing the model that performs best under given circumstances. Fine-tuning goals are to learn new features and use the newly added data to complement the pre-trained model. This step will deactivate the backward propagation in this scenario. As a result, the features

Algorithm 3: Share cipher algorithm

Input: image, $matrix-Share1$, $matrix-Share2$, $matrix-Share3$, $matrix-Share4$

Output: $ImgShare1$, $ImgShare2$, $ImgShare3$, $ImgShare4$

```
Begin
    1: Define the image size $[h, w]$
    2: for every $(h, w)$ do
    3: gettig value from pixel =image [h, w] called values of color
    4: separated a binary to obtain value $red$ = byte(binary[1,..,8]) $green$ = byte(binary[9,..,16]) $blue$= byte(binary[17,..,24])
    5: Creation Share 1
    5-1: Xor operation
            Red 1= $red \oplus matrix-Share1[h, w]$
            Green 1 = $green \oplus matrix-Share1[h, w]$
            Blue 1 = $blue \oplus matrix-Share1[h, w]$
    5-2: converting to binary.
            C_R 1=binary (Red 1)
            C_G 1=binary (Green 1)
            C_B 1=binary (Blue 1)
    5-3: combine the values of binary values to create the 24 bit RGB image.
            color = [C_R 1, C_G 1, C_B 1]
    5-4: Save the color values in the image's associated cell of the $ImgShare1$
    6: Creation Share 2
    6-1: Xor operation
            Red 2= $red \oplus matrix-Share2[h, w]$
            Green 2= $green \oplus matrix-Share2[h, w]$
            Blue 2= $blue \oplus matrix-Share2[h, w]$
    6-2: converting to binary.
            C_R 2=binary (Red 2)
            C_G 2=binary (Green 2)
            C_B 2=binary (Blue 2)
    6-3: combine the values of binary values to create the 24 bit RGB image.
            color = [C_R 2, C_G 2, C_B 2]
    6-4: Save the color values in the image's associated cell of the $ImgShare2$
    7: Creation Share 3
    7-1: Xor operation
            Red 3= $red \oplus matrix-Share3[h, w]$
            Green 3 = $green \oplus matrix-Share3[h, w]$
            Blue 3 = $blue \oplus matrix-Share3[h, w]$
    7-2: converting to binary.
            C_R3=binary (Red 3)
            C_G3=binary(Green 3)
            C_B3=binary (Blue 3)
    7-3: combine the values of binary values to create the 24 bit RGB image.
            color = [C_R 3, C_G 3, C_B 3]
    7-4: Save the color values in the image's associated cell of the $ImgShare3$
    8: Creation Share 4
    8-1: Xor operation
            Red 4= $red \oplus matrix-Share4[h, w]$
            Green 4= $green \oplus matrix-Share4[h, w]$
            Blue 4= $blue \oplus matrix-Share4[h, w]$
    8-2: converting to binary.
            C_R4=binary (Red 4)
            C_G4=binary (Green 4)
            C_B4=binary (Blue 4)
    8-3: combine the values of binary values to create the 24 bit RGB image.
            color = [C_R 4, C_G 4, C_B 4]
    8-4: Save the color values in the image's associated cell of the $ImgShare 4$
9: Return $ImgShare1$, $ImgShare2$, $ImgShare3$, $ImgShare4$
End
```

are extracted based on the ImageNet dataset. Where the selected data set is split into 80 % for "training" and 10% for each "testing" and "validation" partition, "batch size" =32 and "epoch" =100. Import the pre-trained Densenet-121and remove the output layer adding Flatten layer to the top model and updating weights for the three last layers for this network. Fig. 2 shows the network architecture.

3.5 Image retrieval

The user of the image encrypts the image to

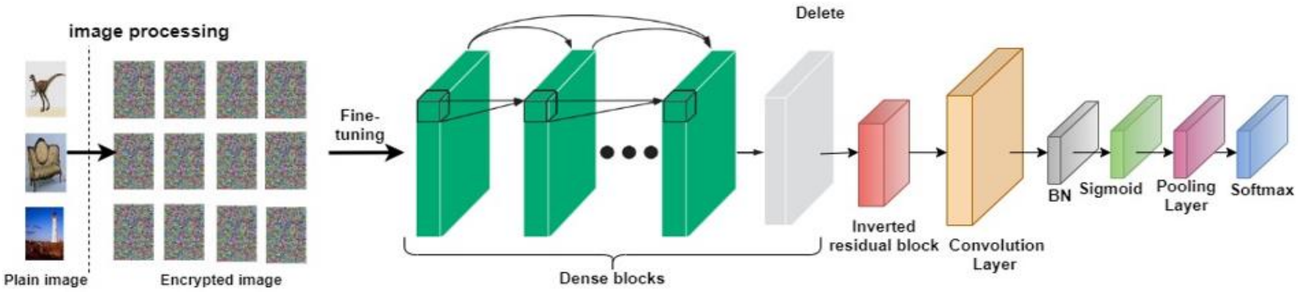
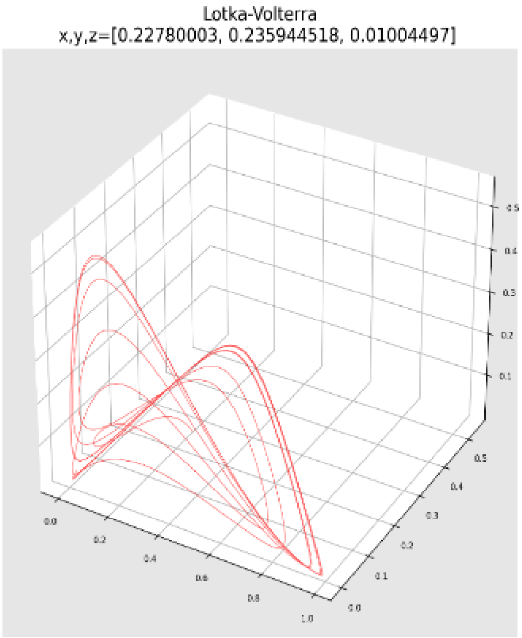
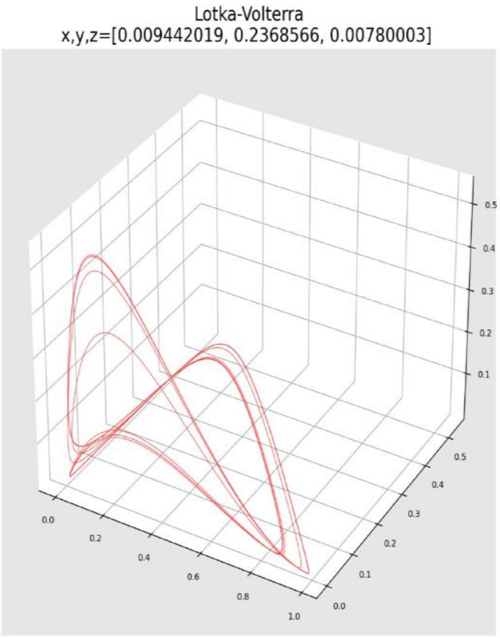


Figure. 2 Network architecture

Table 2. 3D histogram of the Lotka-Volterra chaos map Behavior

Case 1	Case 2
	

retrieve related images from the query image into four shares employing Algorithms 1, 2, and 3 and extracts the features for share 1 using the improved Densenet-121 with fine-tuning. And then sends the share 1 and its features to the cloud. The cloud server takes the feature and compares it with the features of the entire dataset by calculating the Euclidean distance between the feature vector of the query image and the feature vectors of the dataset kept on the cloud according to the following equation:

$$dis\left(e_q \,, E_n\right) = \left\|f_q - f_i\right\|^2 , i = 1,2,3, \dots \dots , n \qquad (4)$$

The result is saved in a list, and the results are arranged in ascending order. Then k images are images based on the mAPs calculates.

4. Results and discussion

The Corel10K dataset was used for the studies.

Table 3. Results of the generation 4 random shares

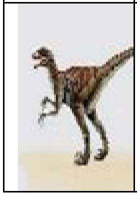
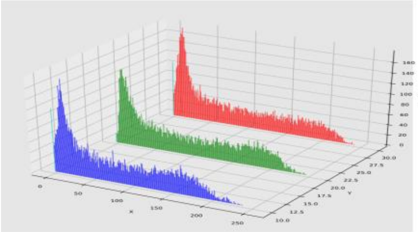
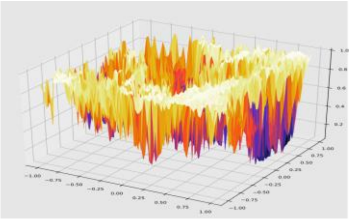
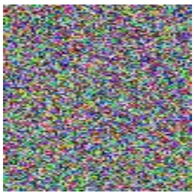
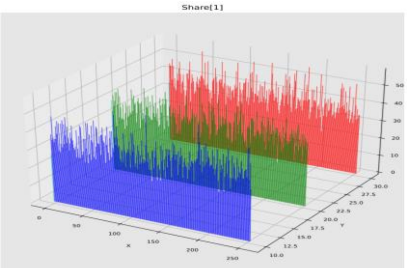
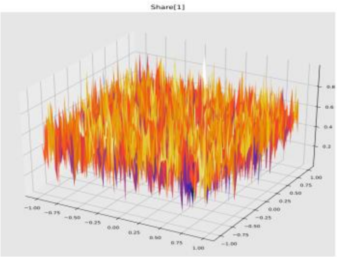
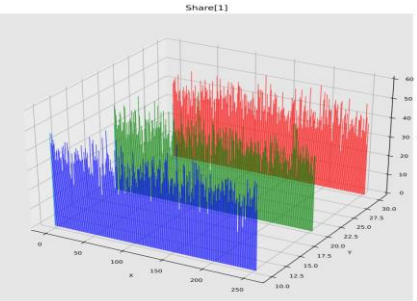
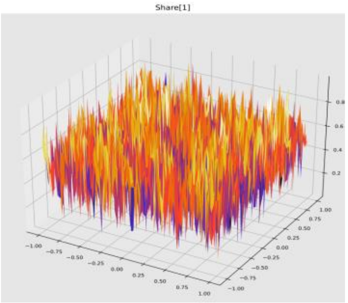
Original	Visual Cryptography			
	Share-1	Share-2	Share-3	Share-4
				

Table 4. 3D and mesh histogram of the original image 2 shares

<table>
<tr><th colspan="3">Original</th><th>3D Hisgram</th><th>Mesh Hisgram</th></tr>
<tr><td colspan="3"> </td><td></td><td></td></tr>
<tr><td></td><td colspan="4">Share[1]</td></tr>
<tr><td rowspan="3">Case1</td><td>X=0.22</td><td rowspan="3"></td><td rowspan="3"> </td><td rowspan="3"> </td></tr>
<tr><td>Y=0.235</td></tr>
<tr><td>Z=.0.01</td></tr>
<tr><td rowspan="3">Case2</td><td>X=0.009</td><td rowspan="3"> </td><td rowspan="3"> </td><td rowspan="3"> </td></tr>
<tr><td>Y=0.236</td></tr>
<tr><td>Z=.0.227</td></tr>
</table>

There are 100 categories in total and 100 related images in each category. Python was used to create the experiment's code, which was run on a Windows 10 computer with an Intel Core i7 running at 3.60 GHz and 16 GB of RAM. 80%, 10%, and 10% of the chosen dataset were divided into three pieces for "training," "testing," and "validation," respectively, in the experiment. The backbone network was decided upon as being DenseNet-121. We employ the train model, which was trained using the ImageNet dataset, in fine-tuning. The learning rate is set to 0.01, the momentum to 0.9, the batch size to 32, and the epochs to 100 for the stochastic gradient descent (SGD) network optimizer. The results of the proposed system are presented in two distinct sections. The first section focuses on image encoding, and the outcomes of this encoding process are detailed in subsection (4.1). The second section pertains to image retrieval based on content, and the retrieval results are showcased in subsection (4.2) and (4.3) Search efficiency.

4.1 Results of the image encryption

The proposed system using visual cryptography based on 3D Lotka_Volterra chaos map for create 4 random share as illustrated in Algorithms 1, 2, and 3. To assess the efficacy and robustness of the encryption algorithm, two distinct case studies are employed, each representing different values for the initial parameters of the 3D Lotka_Volterra chaos map. By generating random shares for each input image based on these identified case studies, the performance of the encryption algorithm is evaluated. A comparative analysis is then conducted to identify the most optimal case studies that yield the best encryption results. The following cases are considered for evaluation:

Case1: values of initial parameters [x=0.22,y=0.235,z=0.0100]

Case2: values of initial parameters [x=0.009,y=0.2368,z=0.0078].

The above study case values were determined through the implementation of the proposed algorithm and careful observation of the results of chaotic map. These two cases study values were found to generate highly random behavior of the 3D Lotka_Volterra chaos map. Table 2 presents the 3D histogram depicting the behavior of the Lotka-Volterra chaos map across two selected case studies of the initial parameters. The purpose of these studies is to generate random numbers that correspond to the dimensions of the input image. The suggested encryption algorithm's security, resilience, speed, and efficiency will be evaluated using encryption image analysis and histogram test, encryption time measurements, and statistical measures. These criteria will reveal the algorithm's performance and encryption efficiency.

4.1.1. Encryption image analysis and histogram test

Visual cryptography and chaotic maps generate four random shares for each input image, matching their sizes. Table 3 shows share generation algorithm results for 2 case studies of initial parameters over three images. As the encrypted image's histogram flattens, the suggested encryption algorithm becomes more resistant to statistical attacks. Table 4 shows the frequency distribution of each color channel for a chosen image. 3D and mesh histograms assess this distribution. The frequency distribution of the two shares created from this image is compared to the original image using two distinct case studies of initial parameters.

4.1.2.

Encryption time

Table 5 shows visual cryptography encryption time results employing 4 cipher shares from 4 images and 2 case studies of initial chaotic parameters. Encryption time is a crucial metric for evaluating the given method. The millisecond (ms) table shows the algorithm's consistent and excellent encryption time.

Case 1 has the lowest average encryption time at 0.1587315ms, followed by Case 2 at 0.159732ms. This comparison shows that Case 1 has the fastest encryption time.

4.1.3. Statistical measures

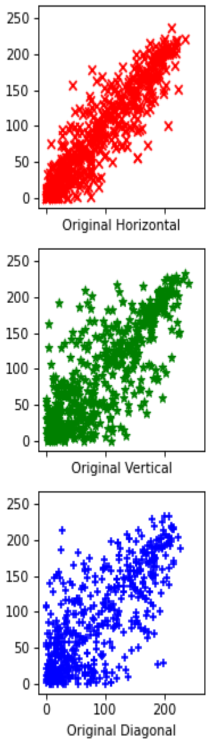
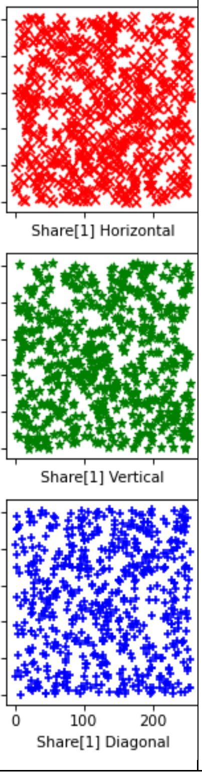
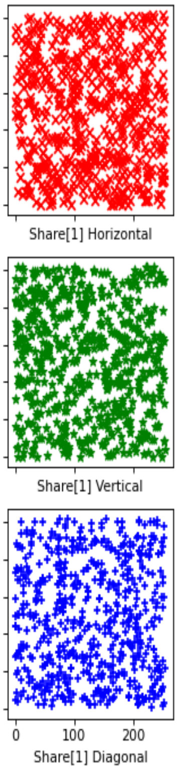
4.1.3.1. Correlation coefficient (CC) test

Table 6 compares original and encrypted images using correlation graphs. The encrypted images are less correlated than the originals. A lower correlation means more security.

Table 5. Encryption time in (ms)

Shares image With size (pixels) $80\times 120 \cong 24KB$	Case1	Case2
Imge1-Share1	0.162531	0.156991
Imge2-Share1	0.157871	0.157199
Imge3-Share1	0.157327	0.172497
Imge4-Share1	0.157197	0.152241
Average	**0.1587315**	0.159732

Table 6. Results of the correlation coefficient (CC) measure

Original	Case1	Case2
		

4.1.3.2. Entropy test

A reliable encryption scheme aims to maximize the level of randomness in the cipher image. The entropy test is used to evaluate its degree of randomness. Fig. 3 shows the entropy values calculated for two cases of initial chaotic parameters. Specifically, the focus was on the entropy of the first share associated with each image. A comprehensive analysis was conducted by individually calculating

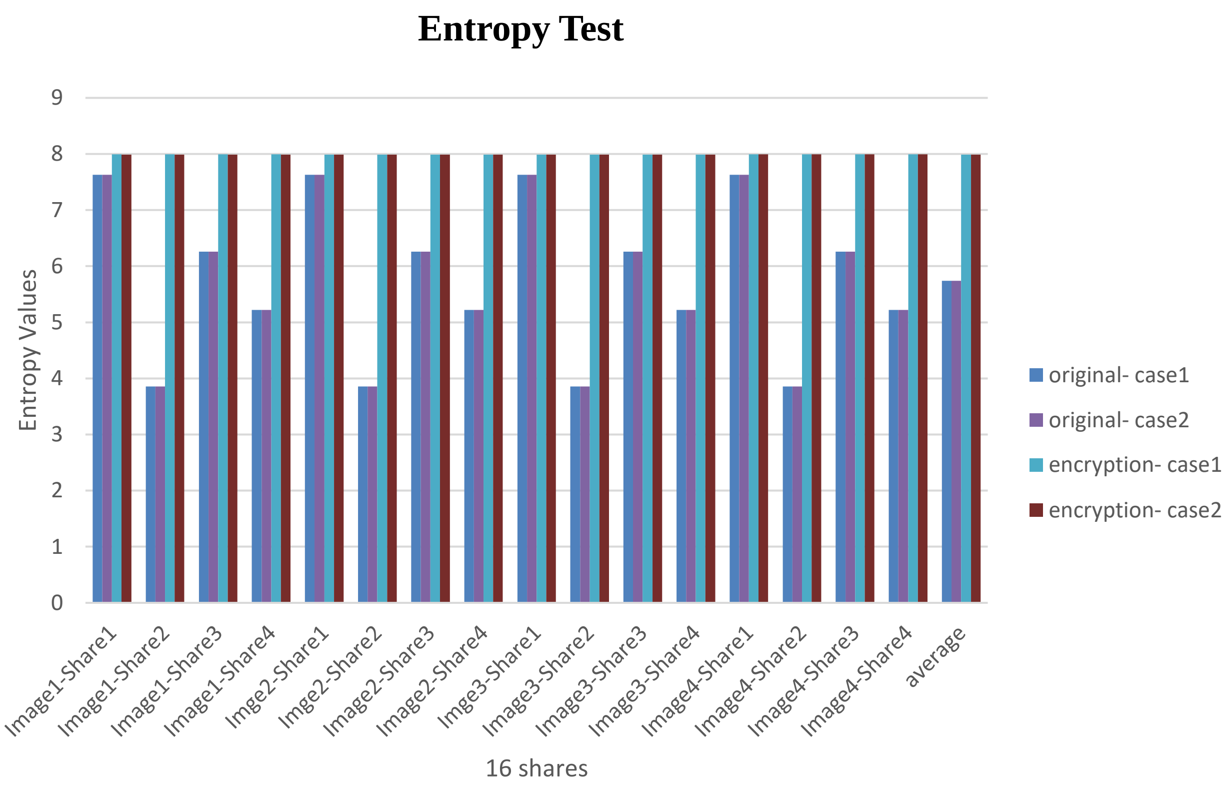


Figure. 3 Results of the entropy test

and comparing the participant entropy with that of the original image. This examination identifies the case study that closely approximates the desired optimal entropy value of 8, shows the entropy results of 16 shares generated from 4 plain images across 2 case studies, and shows higher entropy in the encoded images, indicating the proposed algorithm's efficiency in generating highly random shares. The case studies are arranged based on performance efficiency, with the following order of average entropy values: case1 (7.990869063) and case2 (7.989623688).

4.1.3.3. Mean square error (MSE) test

MSE evaluates cipher image quality by numerically quantifying the difference between the original and shared images. A high MSE value indicates the strong encryption quality of the proposed encryption algorithm, enhancing resistance against attacks Fig. 4 calculates the MSE value between the cipher share and the original image using 16 shares. These excellent values indicate a good cipher quality and demonstrate the robustness of the proposed encryption algorithm, and demonstrate that all case studies exhibited good encryption quality, showcasing varying MSE values among the 16 shares. The determination of the best-case study relies on the average MSE value across the shares. Consequently, in descending order, the highest average MSE values are observed in cases 1 (12764.27263) and 2 (12745.14615). And the recovery of the image after decoding is identical to the original image, that MSE=0 between the plain and decrypted images for 2 cases.

4.1.3.4. Test for "Peak signal-to-noise ratio" (PSNR)

PSNR serves as a means to assess the quality of cipher shares by offering a standardized and measurable indication of the variation between the original and encrypted images. This enables easy interpretation and comparison of image quality, with lower MSE values approaching zero indicating stronger encryption. Fig. 5 calculates the PSNR value between the cipher share and the original image using 16 shares, these excellent values indicate a good cipher quality and demonstrate the robustness of the proposed encryption algorithm, showcasing varying PSNR values among the 16 shares. Notably, the average PSNR values for the two case studies over the 16 shares are very close, indicating the stable and excellent performance of the encryption algorithm regardless of input data quality variations. This is a noteworthy strength of the proposed algorithm. However, there are minor differences among the cases that can be considered for determining the best performance. In order of

Received: June 17, 2023. Revised: August 21, 2023.

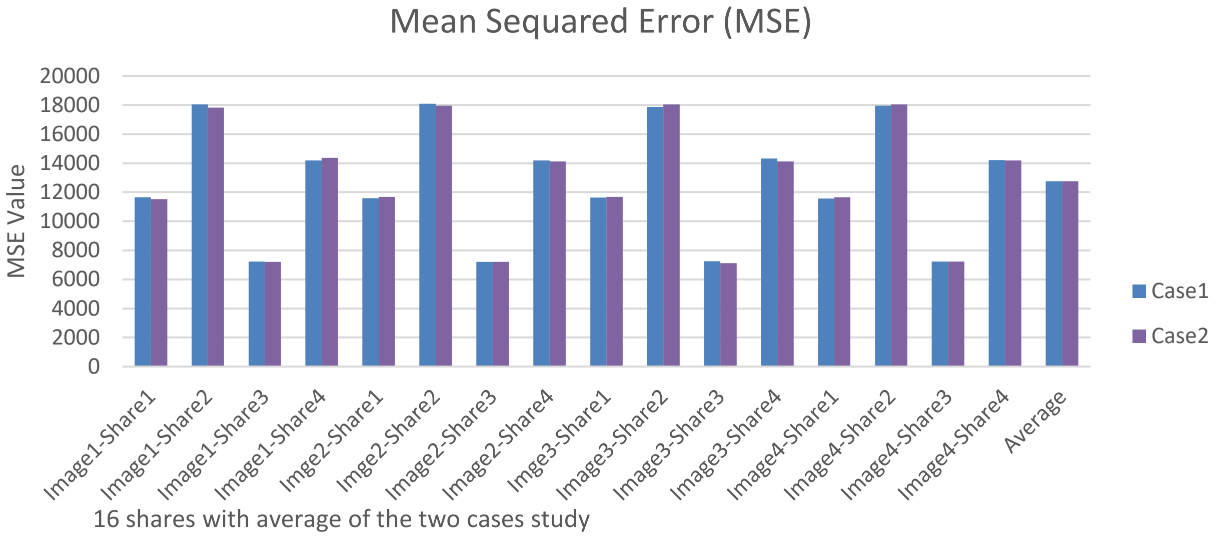


Figure. 4 Results of the MSE

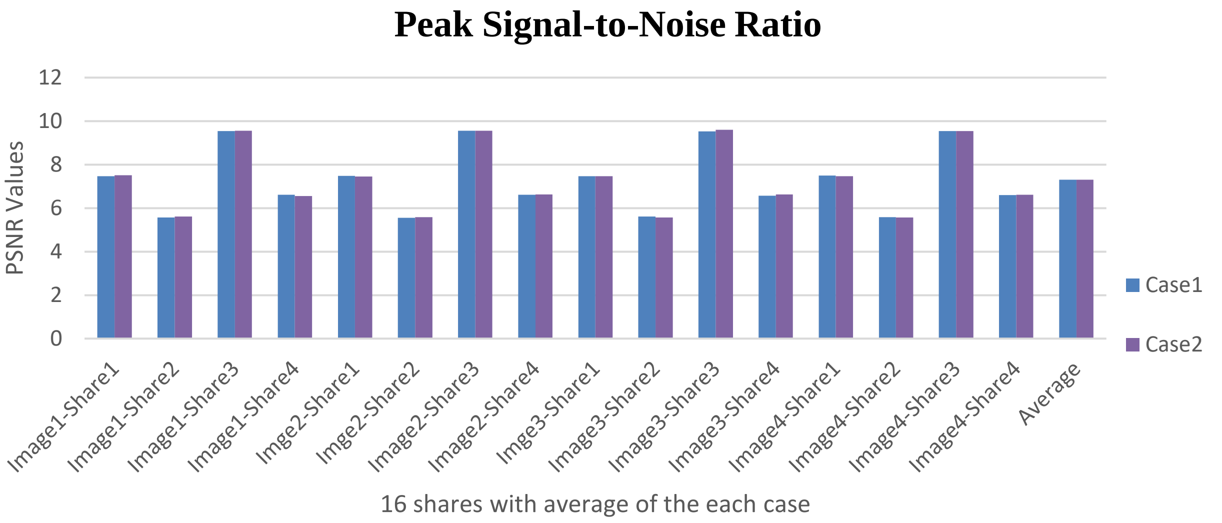


Figure. 5 Results of the PSNR

achievement, the cases are ranked as follows: case 1 (7.300480125), and case 2 (7.30901425).

4.1.3.5. Spatial correlation coefficient (SCC)

SCC evaluates the performance of the visual cryptography encryption algorithm by measuring the similarity between the original image and the ciphered shares. It quantitatively assesses the resemblance of the shares to the original, aiding researchers in gauging the algorithm's effectiveness in generating low-correlation shares. Fig. 6 calculates the SCC values between the cipher shares and the original image, utilising 16 shares for two case studies. By examining the average values of the SCC, one can observe the relative performance of the two case studies. The case with the lowest average SCC value was Case 2 (-0.001123125). Following Case 2, the average SCC values for Case 1 (0.001470688). These findings indicate that Case 2 exhibited the most favourable average SCC value among the two cases.

4.1.3.6. NIST test

The encryption algorithm's output should be more random and unpredictable. National institute of standards and technology (NIST) is one of the ways used to compute randomness. This study utilizes 15 statistical tests from the NIST statistical suite to assess the randomness of the shares generated by the proposed encryption algorithm. Two case studies are conducted, and a significance level of 0.01 is chosen for the probability value (p-

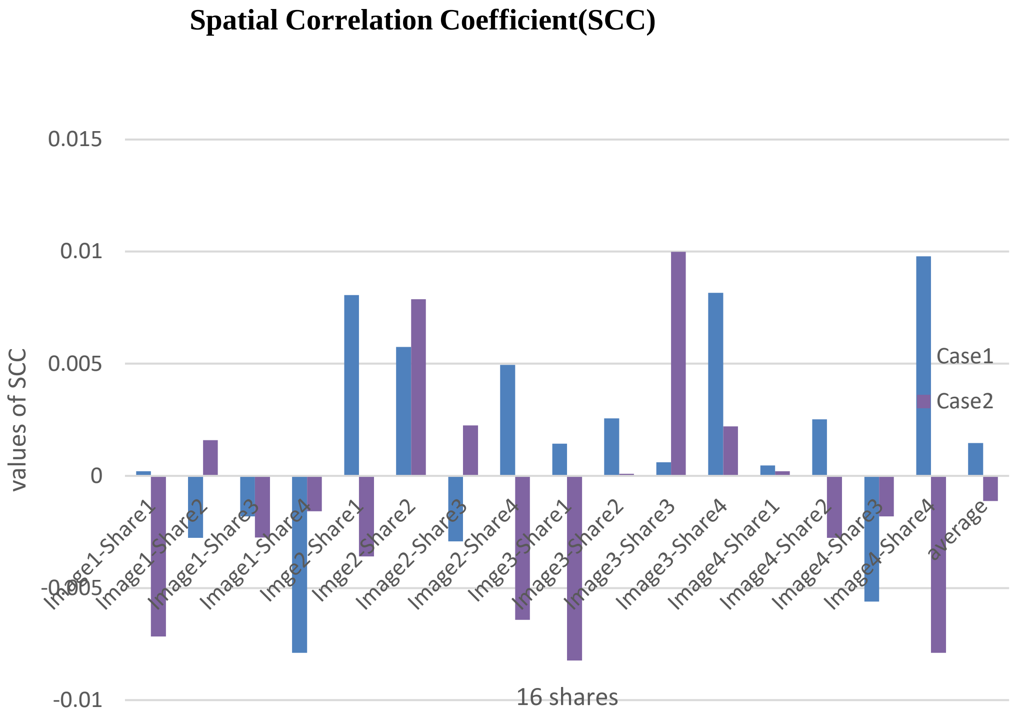


Figure. 6 Results of the SCC

Table 7. Results of running NIST test

NIST Test	Case1	Case2	Average	Status
“Frequency within Block”	0.31753	0.52332	0.44355	P
“Run”	0.55194	0.10573	0.554953	P
“Longest Run”	0.24655	0.59627	0.44998	P
“Binary Matrix Rank”	0.78312	0.41138	0.373745	P
“Discrete Fouri Transform”	0.47226	0.48415	0.52401	P
“Non Overlappin Template”	0.30929	0.99281	0.569078	P
“Overlapping Template”	0.16316	0.17887	0.434045	P
“Linear Complexity”	0.52264	0.41702	0.51759	P
“Serial”	0.27668	0.19714	0.340408	P
“Approximate Entropy”	0.48822	0.56279	0.371048	P
“Cumulative Sums Forward”	0.60963	0.82639	0.591668	P
“Cumulative Sums Reverse”	0.02367	0.63736	0.459773	P
“Random Excursions”	0.52164	0.52332	0.377028	P
“Random ExcursionVariant”	0.47187	0.86685	0.68026	P

value) to determine randomness. A p-value approaching 1 indicates high randomness, while a p-value of zero indicates no randomness. A pass status (P) signifies that the p-value of the tests is greater than 0.001, indicating acceptable output with good randomness. The proposed visual cryptograph encryption algorithm passed all the tests in two case studies as demonstrated in Table7.

4.1.3.7. Comparison and selection best cases study

This section will show case the best-case study chosen from the evaluation tests performed on the proposed encryption algorithm's performance. The selection process will consider the average value obtained from the tests conducted in above. Table 8 provides a comprehensive comparison of the two cases across all evaluation measurements, with the best value for each case indicated by a red line. The tables above illustrates that all study cases successfully achieved optimal values for each measure, with slight variations among them. These differences help determine the best case. The first case outperforms others in terms of entropy, MSE, PSNR. while Case 2 demonstrates strong performance in SCC. As a result, the first case emerges as the most reliable and secure option for generating share images. Table 7 displays the results

Table 8. Comparison of the two cases based on evaluation measurements

Tests	Average Value		Optimal Value
	Case1	Case4	
Entropy	**7.990869063**	**7.989623688**	≅ 8
MSE	**12764.27263**	**12745.14615**	Higher value
PSNR	**7.300480125**	**7.30901425**	Lower value
SCC	**0.001470688**	**-0.00112312**	≅ 0

of comparing 15 NIST tests across two study cases, using the first participant generated from the input image. This comparison serves to evaluate the level of randomization achieved by each study case and determine the best-case study. In general, Table 9 demonstrates that all case studies possess the capability to pass the majority of the 15 NIST tests, thereby validating the security and reliability of the proposed visual cryptograph encryption algorithm with chaotic maps. Specifically, case 1 successfully passes all tests and exhibits superior results compared to other case studies. Therefore, case 1 is selected as the best case, effectively enhancing the performance of the proposed algorithm.

4.2 Results of the image retrieval

In image retrieval, the proposed system utilizes the Improved Densenet-121 with Fine-tuning with 4095-Dim features. The Corel10k dataset will be partitioned into three distinct groups denoted as (k) to extract essential features. These groups will consist of 20, 60, and 100 image collections, respectively. The next step involves calculating the average search precision for the feature extraction model, considering different size values of [20, 60, 100] for each group (k) of [20, 40, 100] images. Fig. 7 compares the results obtained from mAP@k with different sizes [20, 60, 100] and Top-k values [100, 40, 20]. The comparison reveals that the DenseNet-121 with Fine-tuning model achieves the highest performance in terms of mAP@k with a score of (45.95) in the [Top-k=100, Size=100], mAP with a score of (68.94) in the [Top-k=20, Size=20], We note that it is significantly superior to the previous methods [7, 19].

Table 9. Comparison of the two cases based on NIST test

NIST Test	Visual-Cryptography share [1]			
	Case 1		Case 2	
“Frequency within a Block”	0.49953	**P**	0.00601	NP
“Run”	0.31753	**P**	0.01042	**P**
“Longest Run”	0.55194	**P**	0.00917	NP
“Binary Matrix Rank”	0.42052	**P**	0.01042	**P**
“Discrete Fourier Transform”	0.78312	**P**	0.01198	**P**
“Non - Overlapping Template”	0.47226	**P**	0.00917	NP
“Overlapping Template”	0.30929	**P**	0.01042	**P**
“Linear Complexity”	0.16316	**P**	0.01198	**P**
“Serial”	0.52264	**P**	0.00917	NP
“Approximate Entropy”	0.27668	**P**	0.01164	**P**
“Cumulative Sums Forward”	0.48822	**P**	0.00917	NP
“Cumulative Sums Reverse”	0.60963	**P**	0.01198	**P**
“Random Excursions”	0.02367	**P**	0.01198	**P**
“Random Excursions Variant”	0.29687	**P**	0.009171	NP
“Frequency within a Block”	0.47187	**P**	0.86685	**P**

4.3 Search efficiency

Encryption time, feature extraction, and search time will all be used in this section to illustrate how effective our system is.

4.3.1. Encryption time

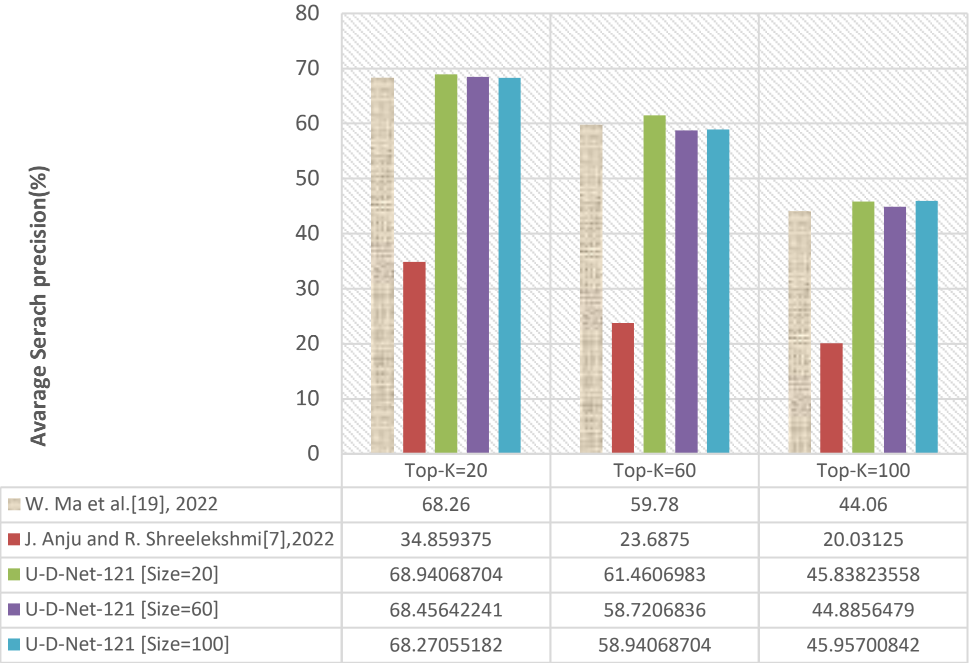
We briefly discussed this subject earlier in the subsection 4.1.2.

4.3.2. Feature extraction

In this study, we extract the semantic features of the images encoded by users. Moreover, the representation of the feature is 1024-dim, 4095-dim which is greater than the representation of all the previous methods, while the feature extraction time is slightly larger than the previous methods. The feature extraction efficiency is 121.25s.

4.3.3. Search time

Although our approach's content-based image



	Top-K=20	Top-K=60	Top-K=100
▨ W. Ma et al.[19], 2022	68.26	59.78	44.06
🟥 J. Anju and R. Shreelekshmi[7],2022	34.859375	23.6875	20.03125
🟩 U-D-Net-121 [Size=20]	68.94068704	61.4606983	45.83823558
🟪 U-D-Net-121 [Size=60]	68.45642241	58.7206836	44.8856479
🟦 U-D-Net-121 [Size=100]	68.27055182	58.94068704	45.95700842

Figure. 7 Comparison between the update dense net-121 with some related works based on the parameter mAp

retrieval service was more secure and accurate than other techniques, which results in a longer recovery time than theirs, this offsets the longer recovery time.

5. Conclusion

This paper uncovers several inferences obtained by implementing and evaluating the suggested method. These conclusions are:

1. The proposed CIBR is built upon the integration of hybrid lightweight and transfer deep learning algorithms, resulting in a robust and efficient framework for image retrieval.

2. This paper ensures a higher level of protection for visual cryptographs, by presenting an innovative (n,n) hybrid visual cryptographic scheme that utilizes the 3D Lotka-Volterra chaos map, which has proven effective in several measures (MSE, PSNR, Entropy, CC, SCC) These measures were not accounted for in the previous methods[7, 19].

3. Tables 2 and 4 proved that the cipher images are uniformly distributed histograms, providing compelling evidence of the algorithm's exceptional effectiveness in concealing plain image information and its robustness against histogram analysis attacks.

4. Table 5 confirmed that the proposed encryption algorithm is fast (0.1587315ms) for 4 shares, rendering it highly suitable for efficient and rapid encryption in cloud environments.

5. Table 6 verified that the condition of the initial parameters [x = 0.22, y = 0.235, z = 0.0100] appears as a more reliable and safe option for generating participation images because it shows superior performance in entropy (7.990869063), MSE (12764.27263), PSNR (7.300480125). And the recovery of the image after decoding is identical to the original image, that MSE=0 between the plain and decrypted images

6. 6. By comparing the proposed system with related works in terms of the mAP parameter as shown in Fig. 7 it is evident that the proposed model possesses the capability to expand the dataset by generating four cipher shares from each input plain image. Remarkably, the proposed model achieved the highest mAP rate of (45.95, size=100) using the proposed DenseNet-121 feature extraction model. This signifies the outstanding and depend able performance of the proposed system in retrieving secure images with high accuracy from the cloud environment. These results establish the superiority and effectiveness of the proposed model in comparison to existing approaches.

For future work, the possibilities of integrating the proposed image retrieval model with other applications or domains, such as e-commerce platforms, social media platforms, or multimedia content management systems, could be explored.

Users may get enhanced functionality and value-added services due to this connection.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Shaima Miteb Sadoon and Ziyad Tariq Mustafa Al-Ta'i; methodology, Shaima Miteb Sadoon; software, Ziyad Tariq Mustafa Al-Ta'i; validation,Shaima Miteb Sadoon,and Ziyad Tariq Mustafa Al-Ta'i; formal analysis, Shaima Miteb Sadoon; investigation, Ziyad Tariq Mustafa Al-Ta'i; resources, Shaima Miteb Sadoon; data curation, Ziyad Tariq Mustafa Al-Ta'i; writing—original draft preparation, Ziyad Tariq Mustafa Al-Ta'i; writing—review and editing, Shaima Miteb Sadoon; visualization, Ziyad Tariq Mustafa Al-Ta'i; supervision, Ziyad Tariq Mustafa Al-Ta'i; project administration, Shaima Miteb Sadoon; funding acquisition, Ziyad Tariq Mustafa Al-Ta'i.

References

[1] Y. Ma, L. Wu, X. Gu, J. He, and Z. Yang, "A secure face-verification scheme based on homomorphic encryption and deep neural networks", *IEEE Access*, Vol. 5, pp. 16532–16538, 2017.

[2] K. Lin, H. F. Yang, J. H. Hsiao, and C. S. Chen, "Deep learning of binary hash codes for fast image retrieval", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Boston, MA, USA, pp. 27–35, 2015.

[3] V. V. Estrela and A. E. Herrmann, "Content-based image retrieval (CBIR) in remote clinical diagnosis and healthcare", In: *Encyclopedia of E-Health and Telemedicine*, IGI Global, pp. 495–520, 2016.

[4] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey", *J. Netw. Comput. Appl.*, Vol. 79, pp. 88–115, 2017.

[5] J. Qin, J. Chen, X. Xiang, Y. Tan, W. Ma, and J. Wang, "A privacy-preserving image retrieval method based on deep learning and adaptive weighted fusion", *Journal of Real-Time Image Processing*, Vol. 17, pp. 161–173, 2020.

[6] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing", *IEEE Trans. Inf. Forensics Secur.*, Vol. 11, No. 11, pp. 2594–2608, 2016.

[7] J. Anju and R. Shreelekshmi, "A faster secure content-based image retrieval using clustering for cloud", *Expert System. Application.*, Vol. 189, p. 116070, 2022.

[8] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos, "Practical privacy-preserving content-based retrieval in cloud image repositories", *IEEE Transaction Cloud Computing*, Vol. 7, No. 3, pp. 784–798, 2017.

[9] Q. Gu, Z. Xia, and X. Sun, "MSPPIR: Multi-source privacy-preserving image retrieval in cloud computing", *Futur. Gener. Computing System*, Vol. 134, pp. 78–92, 2022.

[10] Y. Li, J. Ma, Y. Miao, Y. Wang, X. Liu, and K. K. R. Choo, "Similarity search for encrypted images in secure cloud computing", *IEEE Transactions on Cloud Compuing*, Vol. 10, No. 2, pp. 1142–1155, 2020.

[11] L. Song, Y. Miao, J. Weng, K. K. R. Choo, X. Liu, and R. H. Deng, "Privacy-Preserving threshold-based image retrieval in cloud-assisted internet of things", *IEEE Internet Things Journal*, Vol. 9, No. 15, pp. 13598–13611, 2022.

[12] Z. Xia, N. N. Xiong, A. V Vasilakos, and X. Sun, "EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing", *Information Sciences*, Vol. 387, pp. 195–204, 2017.

[13] S. Gkelios, A. Sophokleous, S. Plakias, Y. Boutalis, and S. A. Chatzichristofis, "Deep convolutional features for image retrieval", *Expert Systems with Applilications*, Vol. 177, p. 114940, 2021.

[14] S. Hussain, M. A. Zia, and W. Arshad, "Additive deep feature optimization for semantic image retrieval", *Expert Systems with Applilications*, Vol. 170, p. 114545, 2021.

[15] Ş. Öztürk, "Stacked auto-encoder based tagging with deep features for content-based medical image retrieval", *Expert Systems with Applilications*, Vol. 161, p. 113693, 2020.

[16] W. Pan, M. Wang, J. Qin, and Z. Zhou, "Improved CNN-Based Hashing for Encrypted Image Retrieval", *Security Communication Networks*, Vol. 2021, 2021.

[17] G. Huang, Z. Liu, G. Pleiss, L. V. D. Maaten, and K. Q. Weinberger, "Convolutional Networks with Dense Connectivity", In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 44, No. 12, pp. 8704–8716, 2019.

[18] J. Tang, Z. Xia, L. Wang, C. Yuan, and X. Zhao, "OPPR: An outsourcing privacy-preserving JPEG image retrieval scheme with local histograms in cloud environment", *Journal of Big Data*, Vol. 3, No. 1, p. 21, 2021.

[19] W. Ma, T. Zhou, J. Qin, X. Xiang, Y. Tan, and Z. Cai, "A privacy-preserving content-based image retrieval method based on deep learning in cloud computing", *Expert Systems with Applilications*, Vol. 203, p. 117508, 2022.

[20] J. Z. Wang, J. Li, and G. Wiederholdy, “Simplicity: Semantics-sensitive integrated matching for picture libraries”, In: *Laurini, R. (eds) Advances in Visual Information Systems*, Berlin, Heidelberg, pp. 360–371, 2000.