mathematics

MDPI

Article

Fine-Grained Encrypted Image Retrieval in Cloud Environment

Yi-Hui Chen [1,2,*,†] and Min-Chun Huang [1,†]

[1] Department of Information Management, Chang Gung University, Taoyuan City 33302, Taiwan; mandy871024374@gmail.com
[2] Kawasaki Disease Center, Kaohsiung Chang Gung Memorial Hospital, Kaohsiung 83301, Taiwan
* Correspondence: cyh@mail.cgu.edu.tw; Tel.: +886-3-211-8800 (ext. 5866)
† These authors contributed equally to this work.

Abstract: With the growing emphasis on privacy awareness, there is an increasing demand for privacy-preserving encrypted image retrieval and secure image storage on cloud servers. Nonetheless, existing solutions exhibit certain shortcomings regarding retrieval accuracy, the capacity to search large images from smaller ones, and the implementation of fine-grained access control. Consequently, to rectify these issues, the YOLOv5 technique is employed for object detection within the image, capturing them as localized images. A trained convolutional neural network (CNN) model extracts the feature vectors from the localized images. To safeguard the encrypted image rules from easy accessibility by third parties, the image is encrypted using ElGamal. In contrast, the feature vectors are encrypted using the skNN method to achieve ciphertext retrieval and then upload this to the cloud. In pursuit of fine-grained access control, a role-based multinomial access control technique is implemented to bestow access rights to local graphs, thereby achieving more nuanced permission management and heightened security. The proposed scheme introduces a comprehensive cryptographic image retrieval and secure access solution, encompassing fine-grained access control techniques to bolster security. Ultimately, the experiments are conducted to validate the proposed solution's feasibility, security, and accuracy. The solution's performance across various facets is evaluated through these experiments.

Keywords: encrypted image retrieval; fine-grained access control; YOLOv5; ElGamal; convolutional neural network (CNN); secure k-nearest neighbor (skNN)

MSC: 68U01

Citation: Chen, Y.-H.; Huang, M.-C. Fine-Grained Encrypted Image Retrieval in Cloud Environment. *Mathematics* **2024**, *12*, 114. https://doi.org/10.3390/math12010114

Academic Editor: Lingfeng Liu

Received: 24 November 2023
Revised: 18 December 2023
Accepted: 27 December 2023
Published: 28 December 2023

1. Introduction

Due to rapid technological advancements and improvements in hardware and software, various aspects of information hiding, including retrieval, storage, and transmission, have greatly improved in convenience. The emergence of cloud technology has revolutionized data processing, offering novel methods for managing information. However, this progress has also ushered in elevated risks to information security, driven by the growing demand for network connectivity. Notable security incidents encompass phishing schemes aimed at stealing personal data, malicious apps designed to abscond with vital phone information, ransomware viruses targeting operating systems and cloud users, and instances of database exposure orchestrated by hackers.

According to the 2022 Mid-Year Network Security Statistics Report [1] from the Smart Protection Network (SPN), the count of threats detected and thwarted by Emotet exhibited a significant upsurge compared to the corresponding period in 2021. The concept of information privacy originated from Reidenberg's work in 1999. In our country, protective measures have been taken, including the enactment of the Personal Data Protection Law, the Medical Law, and other associated regulations, all aimed at safeguarding users against privacy violations. The Personal Data Protection and Healthcare Act provisions secure users' privacy. However, contemporary concerns about privacy breaches persist. For

instance, the incident involving the unauthorized release of private photos of Jennie, a member of a Korean girl group, by hackers has ignited a widespread public outcry. This incident vividly underscores the ongoing inadequacy in effectively addressing the challenge of personal privacy breaches.

Hence, when dealing with sensitive information, particularly during storage or transmission, adopting encryption techniques becomes imperative to counter information leakage. It is important to note that encrypted data often appear convoluted, raising the question of ensuring the triumvirate of confidentiality, integrity, and information availability, as Neumann highlighted in 1977 [2]. This stands as a significant contemporary challenge.

As various fields witness the expansion of large-scale image databases, the associated computational and storage demands on systems have grown considerably. Consequently, an emerging trend involves outsourcing images to cloud platforms. This shift is driven by the ample computational prowess and storage capabilities of the cloud, which are particularly beneficial for resource-constrained users. Nonetheless, apprehensions regarding privacy and security have consistently posed significant hurdles for users contemplating adopting cloud services. For instance, there is concern that unscrupulous cloud service providers (CSPs) might illicitly acquire information for unauthorized use or manipulate data during its transfer.

To effectively safeguard privacy, users' personal information requires transmission, computation, storage, and retrieval in ciphertext, ensuring robust security. Consequently, this paper explores encrypted image retrieval within the cloud environment. Moreover, the traditional approach to content-based image retrieval (CBIR) typically involves retrieving an entire image rather than "searching a small image within a larger one". In contrast, the proposed method focuses on retrieving specific objects or details within images, demanding precise techniques for extracting image features.

A subsequent aspect under investigation pertains to the requirements of fine-grained access control. This exploration commences with the clarification of coarse-grained versus fine-grained distinctions. In the context of image retrieval, coarse-grained access control typically involves authorization based on entire sets of images. This grants users access to complete image collections based on their roles or privileges. While simpler, this approach cannot effectively manage and differentiate image content intricacies. On the other hand, fine-grained access control pertains to the meticulous regulation of access to detailed image attributes. This entails granting access authorization based on image labels and content characteristics. The absence of fine-grained access control can result in administrators having limited control over image resource access, potentially elevating the privacy risks due to the inadequate oversight of users' interaction with sensitive images.

This study performs a comprehensive analysis and synthesis of literature authored by diverse scholars. The subsequent section enumerates the focal points of the research, as outlined in Table 1.

Table 1. Comparison of previous methods.

Methods	Data Type	Feature Extraction	Retrieval Methods	Access Control	Fine-Grained	Accuracy
[3]	Image	CNN	skNN	None	No	0.7
[4]	Image	CNN	skNN	Symmetric polynomial	No	0.8323
[5]	Image	CNN	skNN	Symmetric polynomial	No	0.82
[6]	Image	CNN	DT-PKC	None	No	0.7
[7]	Image	Fisher vector	skNN	Symmetric polynomial	No	0.61

(1) Low retrieval accuracy: The retrieval of standard global features like texture, color, and brightness is feasible, yet accurately describing the image content becomes challenging under varying conditions such as lighting, occlusion, cropping, and more. Images can significantly differ under diverse lighting and angles, leading to diminished retrieval accuracy, as indicated by Shen et al. [8]. While scholars have attempted to utilize methods like the edge histogram descriptor (EHD) [9] and color layout descriptor (CLD) [10] to extract feature vectors from images, these approaches fall short of providing a comprehensive image characterization, thus resulting in low retrieval accuracy. Efforts to establish secure image retrieval schemes have incorporated the scale-invariant feature transform (SIFT) for extracting local image features [11–13]. Although SIFT is resilient against scaling, rotation, and brightness variations, it struggles to accurately capture features in images with smooth edges. Another technique, the Fisher vector [14], employs k-means clustering to aggregate SIFT feature vectors into a global feature vector. However, an analysis of the data presented in Table 1 reveals that the image retrieval accuracy achieved through CNN-based feature extraction methods, as highlighted in the works of Li et al. [4,5], significantly surpasses that of the Fisher vector method. Recent years have witnessed a surge in studies proposing the utilization of convolutional neural networks (CNNs) for extracting image feature vectors, as demonstrated by Li et al. [4,15–17]. The adoption of CNNs in image feature extraction has notably enhanced retrieval accuracy. This improvement has been experimentally proven to outperform SIFT, CLD, and EHD methods. The strength of CNNs lies in their ability to simulate the biological visual system, enabling them to recognize images through unsupervised learning and capture intricate and abstract image information.

(2) Lack of multi-object retrieval: Retrieving multi-object images from a specific image. In contrast to the conventional CBIR approach discussed in Table 1, CBIR schemes typically use single-object images to locate similar counterparts within the database. However, real-world scenarios often involve query images that only encompass a small fraction of the entire scene, with numerous other objects in the picture. This characteristic adds complexity to the retrieval process. While methods like R-CNN can be employed to detect object boundaries, define regions of interest (RoIs), and enhance convolutional neural network features for individual regions to facilitate classification, as explored by Amitha et al. [18], there is a shortage of research on retrieving single-object images (small images) within encrypted environments. Moreover, no existing research tackles the challenge of retrieving multi-object images (large images) from single-object images (small images) while under encryption.

(3) Lack of granularity: Yingying et al.'s research employs the secure k-nearest neighbor (skNN) encryption method to establish a framework for coarse-grained access control [3]. This symmetric method restricts access to outsourced images solely to users possessing the corresponding search key. Users lacking the requisite key cannot gain entry, while authorized users can seamlessly access the entire repository of outsourced images. Given the substantial security risks associated with unauthorized access, implementing fine-grained access control techniques becomes imperative to regulate image access per user. The solutions presented for fine-grained access control in Table 1 predominantly revolve around images autonomously determining their specific authorization criteria. However, a notable gap exists in the research landscape, as no study thus far has proposed a mechanism for controlling multi-object images by delineating authorization criteria for individual objects within the image.

The structure of this paper is as follows. Section 2 presents the detailed methods applied to encryption image retrieval. Section 3 proposes a supporting fine-grained retrieval for encrypted images. Section 4 presents the experimental results to compare with other retrieval systems. The conclusions and discussions are made in Section 5.

2. Related Works

This section delves deeply into the intricate techniques behind fine-grained cryptographic image retrieval schemes while reviewing pertinent research, including feature extraction methods, YOLO method, encryption methods, similarity measurement, and access control methods.

2.1. Feature Extraction Methods

In CBIR, a prevalent technique for extracting essential information involves capturing an image's color, texture, and shape. This extracted feature is crucial as it is pivotal in the search process. By comparing these feature values, the system can more effectively identify query results that align with the user's search request. The quality of these feature descriptors directly impacts the accuracy of the image search results. With this in mind, we will introduce three widely used methods for feature extraction: the edge histogram descriptor, the color layout descriptor, and the VGG-16 module.

VGG-16, as elucidated by Tao et al. [19], stands as a modular convolutional neural network architecture. It accomplishes feature extraction by employing multiple convolution and pooling layers, followed by fully connected layers and a softmax layer for classification. The initial two modules comprise two convolutional layers each, complemented by a maximum pooling layer. Modules three to five integrate three convolutional layers alongside a maximum pooling layer. The sixth module encompasses three fully connected layers.

Notably, all convolutional kernels maintain a 3×3 size with a stride of 1, while the pooling layers adopt a 2×2 size with a stride of 2. An illustrative example of a typical input VGG-16 image is $224 \times 224 \times 3$. This signifies a colored image spanning 224 pixels in width and height, featuring three channels (RGB). As the image progresses through the first module's convolutional layer, the number of channels ascends from 3 to 64, aligning with a corresponding 64-channel maximum pooling layer.

The primary role of the maximum pooling layers is to downsize the feature dimensions. Consequently, a 224-pixel-wide image is halved into 112 pixels in the first module, resulting in an image size of $112 \times 112 \times 64$. Similarly, the fifth module leads to a $7 \times 7 \times 512$ dimension as the image advances through the network.

Figure 1 delineates the architecture of the VGG-16 [19]. In typical use cases, the later stages of the network, encompassing fully connected and softmax layers, are excluded, and only the five initial modules, involving convolutional and pooling layers, are retained for feature vector extraction.

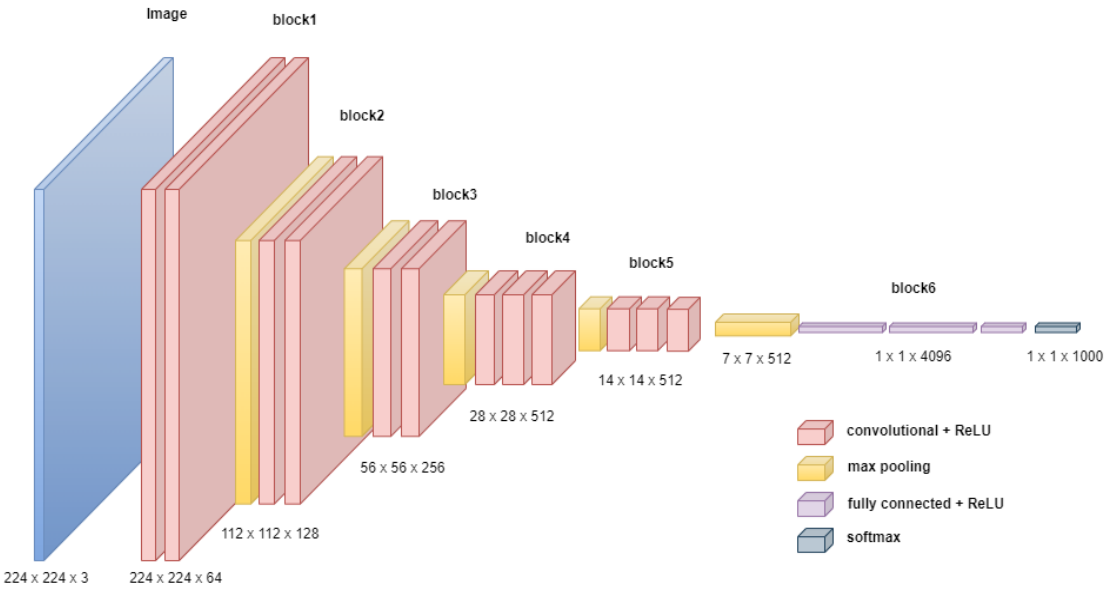

Figure 1. VGG-16 network architecture.

2.2. Yolov5

The inaugural iteration of YOLO, known as YOLOv1, was introduced by Joseph Redmon et al. in 2015 [20]. This version's network architecture takes inspiration from GoogleNet, employing convolutional layers for feature extraction and connectivity layers for object class and location determination. YOLOv1's approach involves dividing images into fixed-size grids, predicting bounding boxes within each grid. However, a limitation arises when the objects under evaluation cannot effectively be enclosed within these fixed-size grids, leading to inaccurate predictions for object location and class. This deficiency becomes particularly pronounced for partially occluded objects, where features may not be fully captured within the predicted grid, resulting in imprecise classification and localization.

Subsequently, YOLOv5 [21] offers flexibility by providing four model sizes (s, m, l, and x) catering to different requirements. YOLOv5's network structure can be delineated into four primary components: input, backbone, neck, and prediction.

(1) Input: The input layer of the YOLOv5 network accepts images with dimensions of $608 \times 608 \times 3$ as input data. This layer undertakes preprocessing tasks, such as MLLE data enhancement (manifold learning-based image enhancement), to enhance the visual quality of the images. This augmentation improves contrast and brightness, leading to superior outcomes in terms of visual presentation[22]. Moreover, YOLOv5 incorporates adaptive image scaling to handle images of varying sizes. Dealing with a broad spectrum of image sizes in the training set poses a challenge. The conventional resizing of all images to a fixed dimension can result in distortion. Adaptive image scaling adeptly adjusts the scaling ratio based on individual image sizes. This approach ensures that the image's shape and aspect ratio are preserved post-scaling. This preservation of proportions enhances the object detection accuracy in YOLOv5's framework.

(2) Backbone: Within YOLOv5, the CSPDarknet53 architecture serves as the backbone. This architectural design, conceptualized by Joseph Redmon, leverages the cross-stage partial network module to enhance the performance of Darknet53. Darknet53, originally proposed by Joseph Redmon, constitutes a deep convolutional neural network architecture. Moreover, the backbone of YOLOv5 incorporates the focus architecture. This innovative design partitions the input feature map into multiple sub-maps. Subsequently, the convolution operations are executed on each sub-feature map. Eventually, the outputs from these convolutions are amalgamated. This design rationale aims to facilitate the model in more effectively capturing nuanced features within images.

(3) Neck: This segment predominantly serves the purpose of feature fusion and multi-scale processing. Conventional object detection techniques often rely on a singular-scale detector for object location and categorization, which can result in inaccuracies when detecting small-scale objects. In response, YOLOv5 incorporates a multi-scale feature pyramid into its feature extraction process by integrating a feature pyramid network (FPN)[23]. This strategic approach extracts features starting from the network's bottom layer and then employs up and down-sampling operations on the lower-layer features. The up-sampling operation enlarges the scale of the feature map, while the down-sampling process reduces it. These operations collectively generate a feature pyramid. Connections are established between each layer of features and adjacent upper layers, and these linked features are subsequently integrated to form the ultimate feature pyramid. The features within this pyramid are ultimately harnessed for object detection. By leveraging this feature pyramid, YOLOv5 effectively addresses the challenge of detecting objects at diverse scales. This approach enhances the detection accuracy by furnishing comprehensive object information across various scales, producing more robust results.

(4) Prediction: The intersection over union (IOU) loss constitutes a pivotal loss function in object detection, quantifying the intersection ratio to the union between the predicted

and actual bounding boxes. However, IOU does not account for size discrepancies or positional deviations between predicted and actual boxes, rendering it less precise. To overcome this limitation, YOLOv5 adopts the CIOU loss (complete intersection over union loss) as its loss function. CIOU loss is an enhanced iteration of IOU. It computes the standard IOU and integrates a corrective factor derived from the distance between the centers of the predicted and actual boxes and the disparities in width and height. This correction factor refines the IOU computation. In scenarios where the gap in the center distance or dimension differences is more significant, the compensation factor becomes more substantial, leading to lower loss values. This mechanism considers both the overlap between predicted and actual boxes and the disparities in the size and position. The result is a more precise loss value, enhancing the object detection accuracy.

2.3. Secure k-Nearest Neighbor (skNN)

K-nearest neighbor (KNN) represents a prevalent supervised machine learning technique that measures distances between query points and training set points. Identifying the k-closest neighbors subsequently predicts or analyzes these neighbors to infer characteristics about the query point.

In contrast, secure k-nearest neighbor (skNN) [3] emerges as a privacy-focused data retrieval algorithm designed to address the privacy concerns associated with kNN queries. Situations often arise where datasets contain sensitive information, posing a substantial privacy risk when conventional kNN distance computations are applied to the data.

The skNN method has been developed to bolster retrieval security. This technique is based on symmetric encryption algorithms. By adopting this approach, the goal is to safeguard data privacy during distance computations, thereby mitigating the risk of privacy breaches that could arise from utilizing traditional kNN methods on sensitive data.

Furthermore, to enable the detection of encrypted images, skNN employs the concept of homomorphic encryption. This principle ensures privacy protection and facilitates computations on ciphertexts without decryption. Remarkably, the decrypted outcome aligns with the original kNN computation result. The forthcoming section provides an overview of the algorithm's introduction:

$$I_i = [I_{i,1}, I_{i,2}, \ldots, I_{i,n}]$$

signifies that the index vector within the ith photo in the image set possesses n dimensions.

$$Q_q = [Q_{q,1}, Q_{q,2}, \ldots, Q_{q,n}]$$

indicates that the query vector within the qth photo of the query image comprises n dimensions.

(1) Secret key generation: As shown in Equation (1), two random numbers r_1 and r_2 are generated to define the range and $r_1 > r_2$. η is a public parameter and a random number chosen by $0 \sim r_1$. M is a random and inverse matrix size of $(2n \times 2n)$. M^{-1} is the inverse matrix of M.

$$key = [\eta, M, M^{-1}] \tag{1}$$

(2) Encryption index vector: Extending by I_i, as shown in Equation (2), the $n-1$ random vector is generated by random number in $[0, r_2]$. $2n$-dimensional vector $\vec{I}_i$ is shown in Equation (2).

$$\vec{I}_i = [I_i, -\frac{1}{2}\sum_{j=1}^{n} I_{i,j}^2, S] \tag{2}$$

As shown in Equation (3), the vector $\vec{I}_i$ is encrypted as $\tilde{I}_i$ with Equation (3). $\vec{e}_i$ is a random and noise matrix chosen by $[0, r_1]$, where $2|max(\vec{e}_i)|$ must be less than η, and $|max(\vec{e}_i)|$ is the maximum absolute value of $\vec{e}_i$.

$$\tilde{I}_i = (\eta \cdot \vec{I}_i + \vec{e}_i) \cdot M \tag{3}$$

(3) Encryption query vector:
Extending Q_q with Equation (4), where δ_q is a random number and $\delta_q \in [0, r_2]$, θ_q is a $(n-1)$ random vector and $\theta_q \in [0, r_2]$ to obtain a $2n$-vector $\vec{Q}_q$.

$$\vec{Q}_q = [\delta_q Q_q, \delta_q, \theta_q] \tag{4}$$

With Equation (5), the vector $\vec{Q}_q$ is encrypted as $\tilde{Q}_q$ and $\vec{e_q}$ is a noise random $(2n)$-vector by $0 \sim r_1$. Moreover, $2\left|max(\vec{e_q})\right|$ must be less than η. $\left|max(\vec{e_q})\right|$ denotes the maximum absolute value of $\vec{e_q}$. ${\vec{Q_q}}^T$ and ${\vec{e_q}}^T$ represent the inverse matrices of $\vec{Q}_q$ and $\vec{e_q}$, respectively.

$$\tilde{I_q} = M^{-1} \cdot (\eta \cdot {\vec{Q_q}}^T + {\vec{e_q}}^T) \tag{5}$$

(4) Interpolation value: The interpolation values I_i and I_q can be computed with Equation (6):

$$\begin{aligned} Comp_i &= \frac{\tilde{I}_i \cdot \tilde{Q}_q}{\eta^2} \\ &= -\frac{\delta_q}{2}(\left\|Q_q - I_i\right\|^2 - \left\|Q_q\right\|^2) + Q_q^T \end{aligned} \tag{6}$$

(5) Similarity computation: The similarity of two different images *ia* and *ib* can be computed with Equation (7). If the value is positive, the two images are similar; otherwise, they are much different.

$$\begin{aligned} Comp_{ia} - Comp_{ib} &= \frac{\tilde{I_{ia}} \cdot \tilde{Q}_q}{\eta^2} - \frac{\tilde{I_{ib}} \cdot \tilde{Q}_q}{\eta^2} \\ &= \frac{\delta_q}{2}(\left\|Q_q - I_{ia}\right\|^2) - (\left\|Q_q - I_{ib}\right\|^2) \end{aligned} \tag{7}$$

3. Proposed Method

The section presents an intricate account of the threat model, system architecture, and the sequential processes encompassed by that architecture. While delving into the threat model, it analyzes potential security threats and risk factors to fortify the system's defensive capabilities. The system architecture delineates the comprehensive design framework of the system, elucidating the interplay between modules and the methodologies governing data flow. This approach ensures both the logical coherence and functional soundness of the system. Furthermore, each step within the system architecture undergoes comprehensive examination, fostering a profound comprehension of operational principles and the significance of each sequential action.

3.1. The Flowchart of System Architecture

Within the cloud environment, the scheme for retrieving fine-grained encryption patterns is primarily structured around four distinct entities: the image owner, the user, the trusted organization (the certificate authority (CA)), and the CSP. Figure 1 depicts the architectural layout among these four entities.

(1) CA: The unit in charge of key management assists image owners and subscribers in generating and distributing keys.
(2) Owner: Its responsibilities encompass training the image set, extracting index vectors from image slices, encrypting both the index vectors and the image set, and ultimately entrusting the resulting ciphertext to the CSP.

(3) User: Conduct object detection on the query image and extract the localized images of the identified objects. Apply the identical feature extraction and encryption techniques to these localized images, mirroring the methods employed by the image owner. Subsequently, upload the encrypted query features of the localized images to the CSP as a query request. Upon receiving the search results, utilize the private key furnished by the CA to decrypt.
(4) CSP: Upon receiving the query request submitted by the user, the system retrieves the top k most similar images from the encrypted index vectors, forming the retrieval results. Subsequently, after verifying the user's identity, this returns the retrieval results to the user.

As shown in Figure 2, the CA initially assumes the key generation and distribution role, disseminating the generated key across image owners, users, and CSPs. Following this, image owners employ the RoleEnc algorithm to encrypt the role code directed towards users using the distributed key. Furthermore, the image owners utilize the IndexEnc and ImageEnc algorithms to encrypt the index vector and image set, ensuring their security and confidentiality. Subsequently, these encrypted data units are uploaded to the CSPs. A dual-layer encryption approach is adopted to reinforce the encrypted data's security during transmission. The encrypted index vector, encrypted image set, and role polynomial are collectively re-encrypted utilizing the IndexReEnc algorithm. These doubly encrypted data units are then uploaded to the CSPs to ensure their safeguarding throughout delivery. Upon receiving the encrypted role code from the image owner, the user initiates the process by decrypting it using the RoleDec algorithm. Subsequently, the user employs YOLOv5 with a seed file for querying object detection within the image. The system then extracts partial images based on object coordinates and generates query vectors. These query vectors are subsequently encrypted using the QueryEnc algorithm. The user implements an additional layer of encryption to enhance the security of the encrypted query vectors during transmission. The user utilizes the QueryReEnc algorithm to re-encrypt the encrypted query vectors alongside their role code, creating a query request uploaded to the CSP. This double-layer encryption strategy safeguards the encrypted query vectors throughout the delivery process. Upon the reception of encrypted data from both the image owner and the user, the CSP executes decryption using the IndexDec and QueryDec algorithms. During retrieval, it calculates the inner product between the encrypted index vector and the encrypted query vector, subsequently performing a comparison based on the relative Euclidean distance. An adverse comparison value indicates that the former index vector's length is more minor, implying closer proximity to the queried vector of the user's interest. The CSP then cross-references the user's role code with the role polynomial associated with the index vector to verify the user's authorization. Following this verification, the CSP identifies the first k-most similar encrypted images, which it re-encrypts using the CSP_ReEnc algorithm. These re-encrypted images constitute the retrieval results transmitted back to the user. Upon reception of the retrieval results, the user can utilize the ImageDec algorithm to decrypt the results, revealing the plaintext images corresponding to the retrieved k images.

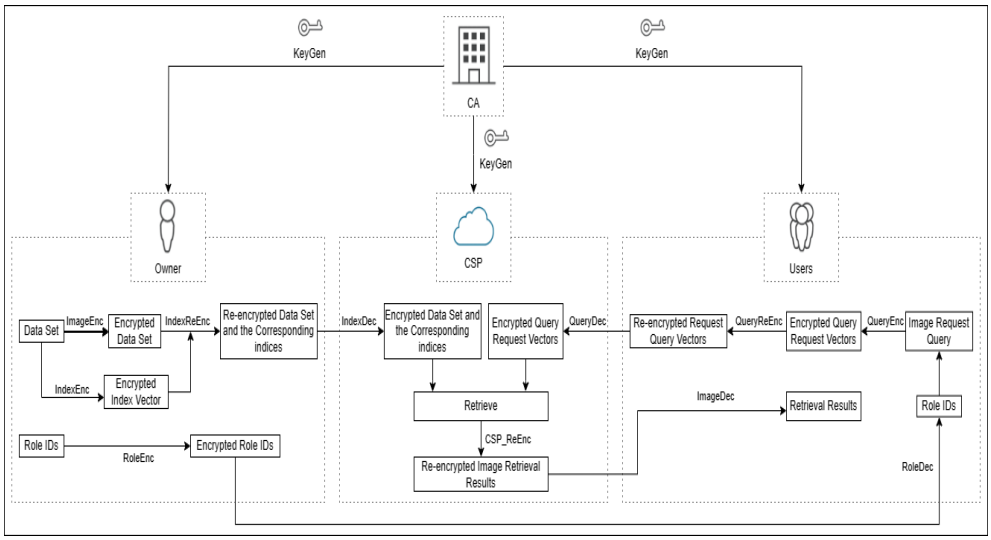

Figure 2. System architecture.

3.2. The Threat Model

In the threat model, the authorized users are trusted, and the CSP is honest but curious. That is, CSP can provide correct services for users, but it might be interesting to have private data stored in the cloud. Two threat models are considered and discussed.

(1) Known Ciphertext attack: CSP might try to infer the related information from to the encrypted images, encrypted indices, and the returned images.
(2) Known background attack: Through statistical analysis, CSP might infer the relationship between plaintexts and ciphertext.

3.3. Feature Extraction

The YOLOv5 is used to recognize the objects in images, where the positions (x_{max}, y_{max}) and (x_{min}, y_{min}) in the image are used to slice the images. Next, the features are extracted by sliced images using the VGG16 [24] model, where the feature size is 25,088 ($7 \times 7 \times 512$). To make the system efficient, different dimension reductions are used. The image dataset, index vectors, and query vectors are defined as follows.

(1) Image dataset: The set M includes m images, denoted by

$$M = [1, 2, \ldots, m]$$

(2) Index vector: The given image has i objects, and each object has an index vector with n-dimension, defined as

$$I_i = [I_{i,1}, I_{i,2}, \ldots, I_{i,n}]$$

(3) Query vector: q images are used as primitive data for a query, and each one is the size of n-dimension as

$$Q_q = [Q_{q,1}, Q_{q,2}, \ldots, Q_{q,n}]$$

3.4. Fine-Grained Access Control

The authors are authorized to access the specific objects in images. An authorized user is assigned to a role and assigned the corresponding authorization to the role. The steps are shown as follows.

(1) Role: A role set, denoted by R, describes the positions or purposes that people have in an organization. The u-th user in R is represented by R_u.
(2) Role-based symmetric polynomial: The identifications of the classified objects are encrypted with Equation (8) by $f(x)$. For instance, four roles are 1, 2, 3, and 4. The region that is just for role IDs is to generate the role polynomial with Equation (8). For

example, as the regions authorized for IDs 3 and 4, the polynomial can be generated as $f(x) = N(x-3)(x-4)$.

(3) The index vectors are mapped to the generated role polynomials, denoted by $(I_i, f(x))$.

$$f(x) = \prod_{ids}(x - id) \tag{8}$$

3.5. Keygen

According to the ElGamal algorithm, a pair secret key is created and denoted by $(pk_o,\ sk_o)$ and $(pk_u,\ sk_u)$, where pk_o and sk_o are the owner's public key and private key. With Equation (1), the secret key is generated and represented by k_{skNN}. The owner and users randomly generate a 128-bit secret key, denoted by k_{AES}.

Thus, the owner and the users own the key k_{skNN} and k_{AES}. The keys are generated as listed below.

(1) Owner:$(pk_o,\ sk_o,\ k_{skNN},\ k_{AES},\ pk_u)$
(2) Users:$(pk_u,\ sk_u,\ k_{skNN},\ k_{AES})$
(3) CSP:$(pk_o,\ pk_u)$

In addition, the owner and users own the secret keys k_{reo} and k_{reu} with Equations (9) and (10), respectively.

$$k_{reo} = g^{pk_o} \bmod p \tag{9}$$

$$k_{reu} = g^{pk_u} \bmod p \tag{10}$$

3.6. Encrypt

Due to the image containing sensitive data or personal information, two encryption methods, AES and skNN, are applied to the proposed scheme. AES firstly encrypts the image, and then skNN is used to encrypt the images implied to contain the features of the image. On the user's side, users could use skNN to retrieve the encrypted images. The Euclid distance is adopted to calculate the differences between the encrypted images and query vectors to find the nearest ones to return to the users. During the encryption procedure, the role ID, weights, encrypted images, encrypted vectors, and query vectors are listed below.

(1) IndexEnc: Through skNN encryption, the owner uses the secret key k_{skNN} to encrypt the index vectors I_i with Equation (2). The I_i can be extended to $\tilde{I}_i$ by Equation (3).
(2) ImageEnc: According to the AES encryption method, the owner can encrypt the images M to $\tilde{M}$ with key k_{AES}.
(3) QueryEnc: Through the skNN encryption method, the user used key k_{skNN} to encrypt the query and extend the query to be Q_q with Equation (4). Finally, Q_q can be encrypted as $\tilde{Q}_q$ with Equation (5).

3.7. ReEnc

The scheme uses the ElGamal encryption method to encrypt the role ID to ensure that the role ID can be securely transferred to the user side. To prevent the key from being stolen by unauthorized users, the ElGamal encryption method encrypts the encrypted image. The details are listed below.

(1) RoleEnc: With the Elgamal encryption method as shown in Equation (11), the image owner uses the encryption key pk_u to encrypt the roid ID R_u to obtain $C_R = (c_{r1}, c_{r2})$.

$$\begin{aligned} c_{r1} &= g^r \ mod\ p \\ c_{r2} &= R_u \cdot pk_u^r \ mod\ p \end{aligned} \tag{11}$$

(2) IndexReEnc: With Equation (12), the owner encrypts the encrypted contents $\tilde{M}$ and $\tilde{I}_i$ with the key k_{reo}. Also, to obtain $C_I = (c_{i1}, c_{i2})$ from $f(x)$ for access control.

$$\begin{aligned} c_{i1} &= g^r \ mod \ p \\ c_{i2} &= (\tilde{I}_i, f(x), \tilde{M}) \cdot k_{reo}^r \ mod \ p \end{aligned} \tag{12}$$

(3) QueryReEnc: Through the Elgamal encryption method, the users re-encrypt the query vector with the secret key k_{reu} to generate the encrypted query $\tilde{Q}_q$ by role ID R_u to obtain $C_Q = (c_{q1}, c_{q2})$.

$$\begin{aligned} c_{q1} &= g^r \ mod \ p \\ c_{q2} &= (\tilde{Q}_q, R_u) \cdot k_{reu}^r \ mod \ p \end{aligned} \tag{13}$$

(4) CSP_ReEnc: When CSP retrieves the most k similar encrypted images and checks the users' identities. The most similar images are encrypted by the Elgamal encryption method. CSP uses users' public key pk_u to encrypt the retrieved results M_k to obtain $C_K = (c_{k1}, c_{k2})$ with Equation (14).

$$\begin{aligned} c_{k1} &= g^r \ mod \ p \\ c_{k2} &= M_k \cdot pk_u^r \ mod \ p \end{aligned} \tag{14}$$

(5) As shown in Figure 3, the owners transmit C_R to the users and C_I up to CSP. After that, the users can request a query C_Q to CSP. CSP obtains the most similar images C_K back to the users.

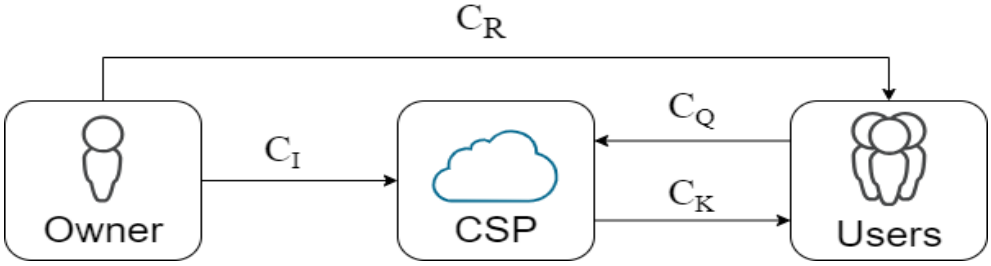

Figure 3. Query of the encryption images diagram.

3.8. Retrieval Process

The CSP decrypts C_I and C_Q as $(\tilde{I}_i, f(x), \tilde{M})$ and $(\tilde{Q}_q, R_u)$. With Equation (6), the CSP calculates the Euclid distance between $\tilde{I}_i$ and $\tilde{Q}_q$ to find the k most similar images. Then, CSP uses R_u to $f(x)$ to check whether the users have rights to access the images. If the $f(x)$ is 0, the user can access the image; otherwise, it is unauthorized.

3.9. Decrypt

As described below, the decryption procedure contains RoleDec, IndexDec, QueryDec, and ImageDec stages.

(1) RoleDec: User can use their own private key sk_u to decrypt the secrets $C_R = (c_{r1}, c_{r2})$ obtaining the role ID R_u with Equation (15).

$$c_{r2} \cdot c_{r1}^{-sk_u} = R_u \cdot g^{rx} \cdot g^{-rx} = R_u \tag{15}$$

(2) IndexDec: CSP uses the public key pk_o from the owner to decrypt $C_I = (c_{i1}, c_{i2})$ and obtain $\tilde{I}_i, f(x)$ and $\tilde{M}$, as shown in Equation (16).

$$c_{i2} \cdot c_{i1}^{-pk_o} = ((\tilde{I}_i, f(x), \tilde{M}) \cdot g^{rx}) \cdot g^{-rx} = (\tilde{I}_i, f(x), \tilde{M}) \tag{16}$$

(3) QueryDec: CSP uses the public key pk_u to decrypt $C_Q = (c_{q1}, c_{q2})$ to obtain the query vectors $\tilde{Q}_q$ and role ID R_u as shown in Equation (17).

$$c_{q2} \cdot c_{q1}^{-pk_u} = ((\tilde{Q}_q, R_u) \cdot g^{rx}) \cdot g^{-rx} = (\tilde{Q}_q, R_u) \tag{17}$$

(4) ImageDec: After the users obtain the retrieval results, the users can decrypt the secret image with their own private key pk_u to decrypt $C_K = (c_{k1}, c_{k2})$ and obtain the retrieval results M_k as shown in Equation (18).

$$c_{k2} \cdot c_{k1}^{-sk_u} = M_k \cdot g^{rx} \cdot g^{-rx} = M_k \tag{18}$$

Finally, the users use private key k_{AES} to decrypt M_k to obtain k secret images.

4. Experimental Results

This paper uses the Python programming language in conjunction with the TensorFlow 2.13.0 package, and it is necessary to establish a development environment. The computer host configurations are as follows:

(1) Processor: AMD Ryzen 7 3700X 8-Core Processor 3.60 GHz
(2) Memory: 32.0 GB
(3) Graphics Card: NVIDIA GeForce RTX 2080 SUPER
(4) Operating System: Windows 11

The dataset is the Caltech101 dataset [5], which includes 101 different categories, including faces, animals, plants, etc. The total number of the dataset is 9144, and the number in each category is in the range of 40–800. This section introduces the preprocessing, dimension reduction, measure metrics, and threat model discussion.

4.1. Preprocessing

Object detection and dimension reduction are processed. In object detection, the label Img is used to label the images to mark the category of the object. The YOLOv5 is used to detect the category message and its position. The features are extracted from the bounding boxes as shown in Figure 4.

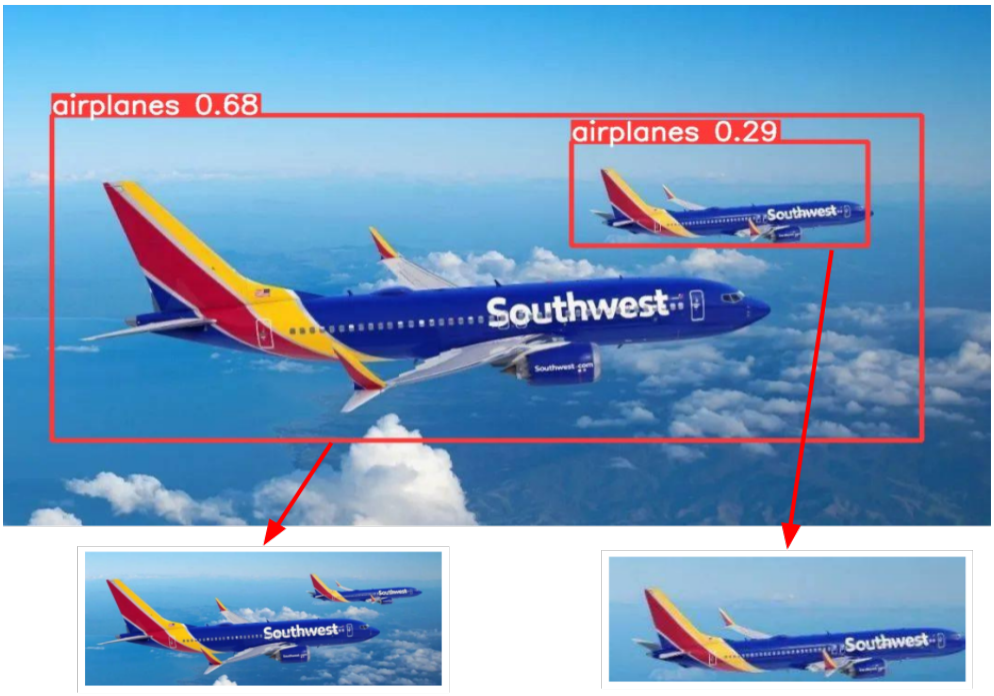

Figure 4. The bounding box examples.

Dimension Reduction

The VGG16 is used to extract the features. To reduce the time cost of image retrieval, principal component analysis (PCA) is used to reduce the dimension of the features. Using a linear transformation, PCA is a typical dimension reduction method for transforming a high-dimension into a low-dimension space. After transformation, the low-dimension space can preserve the features the original dimension space [25].

4.2. Measurement Metric

The performance evaluates whether the most k-th similar images meet the users' requests as shown in Figure 5. The average precision (AP) at top-k($P@k$) [4,5] and mean average precision (MAP) [7,26] as shown in Equation (19) and Equation (20), respectively. In Equation (19), there are $(I_n)_{n=1}^{p}$ returned images and $(R_m)_{m=1}^{q}$ are correct images. MAP obtains the average value of multiple query requests. For example, five returned images $(I_1,\ I_2,\ I_3,\ I_4,\ I_5)$, and I_1 and I_3 are correct. AP is calculated as $(1/1 + 2/3)/2 = 5/6$. MAP is the average value of AP, ranging from 0 to 1. The higher the MAP is, the more accurate it is. The $P@k$ value (i.e., Equation (21)) is also in the range of 0 and 1, where k is the returned image and $num\ proper$ indicates the correct number of returned images. The higher $P@k$ value presents higher accuracy.

$$AP = \sum_{m=1}^{q} (m/n)/q \tag{19}$$

$$MAP = \sum_{t=1}^{d} \frac{AP_t}{d} \tag{20}$$

$$P@k = \frac{num\ proper}{k} \tag{21}$$

Figure 5. Top-k searching results.

Different dimension reductions and feature extractions are to evaluate the corresponding accuracy and compare them to the others.

Three feature extraction methods, EHD, CLD, and VGG16, extract the 64-vector features. In Figures 6 and 7, the VGG16 feature extraction is the best one. Thus, VGG16 feature extraction methods are used in the proposed scheme for image retrieval.

In Tables 2 and 3, the proposed scheme uses VGG16 to extract the features. The results show the highest accuracy when the feature vector size is 128.

In Figure 8, the proposed work compares to MU-TEIR[6], which are under the calteh101 dataset, and the feature extraction is also adopted by VGG16, the vector size is 128. The values of $P@k$, while k is 5, 10, 15, 20, 25, and 30, the proposed work is superior to MU-TEIR [6]. When k is equal to 5, the accuracy is 82%. While the accuracy of the method [4], as shown in Table 1, is higher than ours, it lacks support for fine-grained encrypted image retrieval.

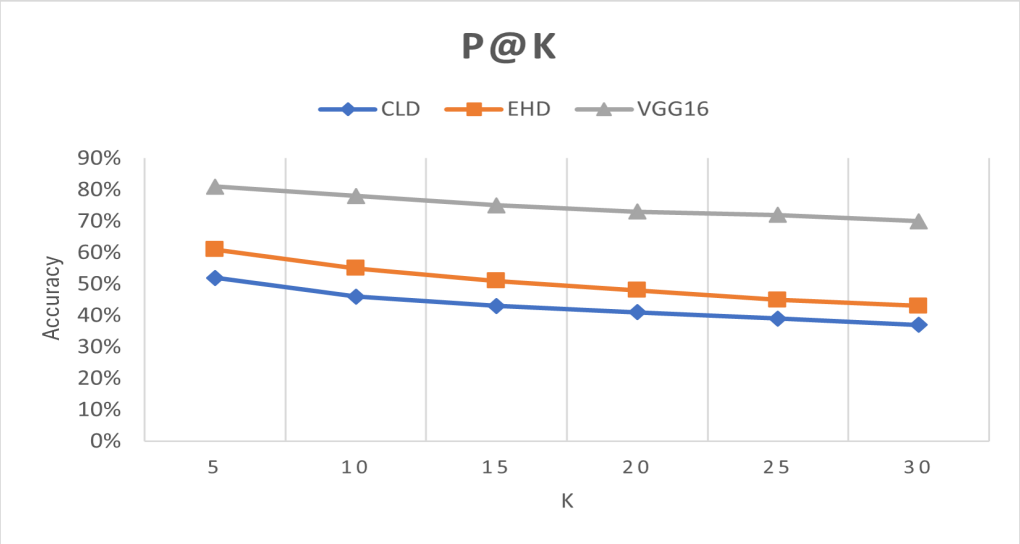

Figure 6. The accuracy of different feature extractions (*P@k*).

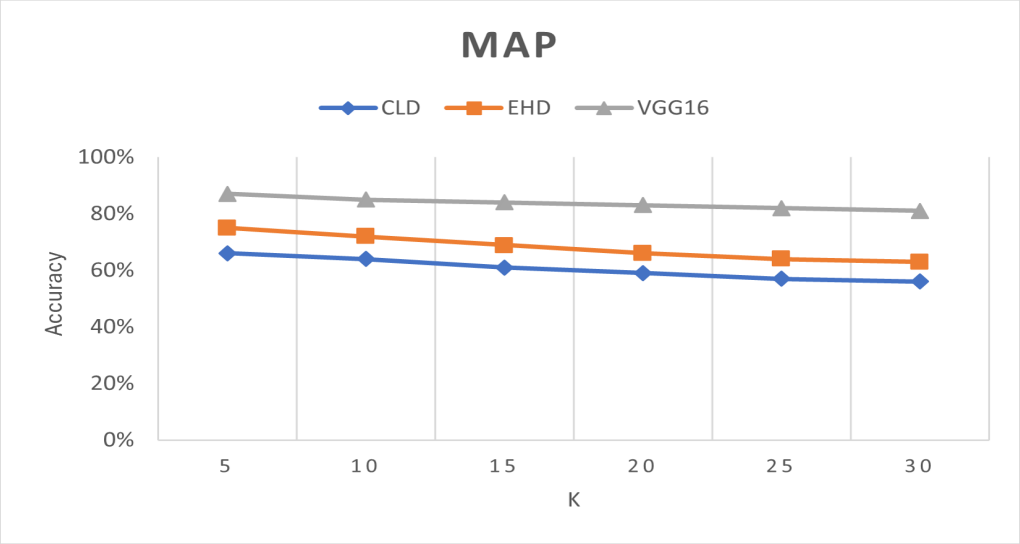

Figure 7. The accuracy of different feature extractions (MAP).

Table 2. Different accuracies on different dimension reduction (*P@k*).

	k = 5	k = 10	k = 15	k = 20	k = 25	k = 30
PCA32	0.76	0.73	0.70	0.68	0.67	0.65
PCA64	0.81	0.78	0.75	0.73	0.72	0.70
PCA128	0.82	0.79	0.76	0.74	0.72	0.69
PCA256	0.81	0.77	0.73	0.71	0.68	0.66
PCA512	0.78	0.72	0.69	0.66	0.63	0.60

Table 3. Different accuracies on different dimension reduction (MAP).

	k = 5	k = 10	k = 15	k = 20	k = 25	k = 30
PCA32	0.84	0.82	0.79	0.78	0.77	0.76
PCA64	0.87	0.85	0.84	0.83	0.82	0.81
PCA128	0.88	0.87	0.85	0.84	0.83	0.82
PCA256	0.87	0.85	0.84	0.83	0.82	0.81
PCA512	0.85	0.84	0.82	0.81	0.79	0.78

4.3. Security Analysis

(1) Known-ciphertext attacks: CSP has encrypted images but lacks the corresponding plaintext. However, CSP can collect the query request to statistically analyze the minor differences to induce the encrypted image mapped to the features to trace the source.

(2) Known-background attacks: CSP has rich background knowledge, including keywords related to statistically different datasets. The previous query requests and the returned queried results are linked together to find the specific plaintext. The attackers expose the implied message of the query requests. The attacker may extract more insights from the known background knowledge through statistical analysis and data correlation disclosure, thereby cracking or inferring hidden content in query requests. This study employs the concept of unverifiable queries, which means that, even if the CSP obtains partial plaintext information about the retrieval results, it is still unable to deduce the hidden content of the query request based on this plaintext information.

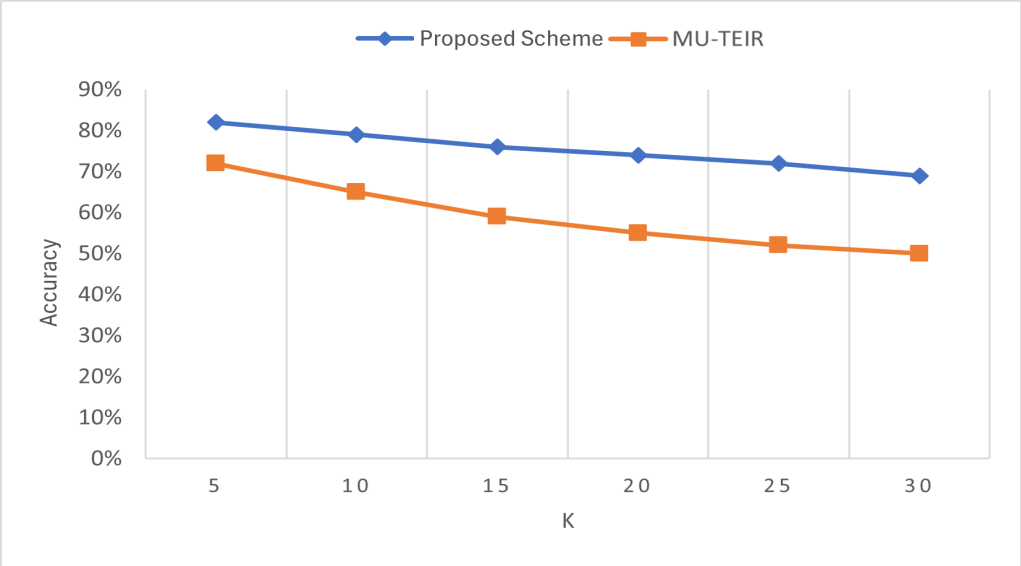

Figure 8. Comparison performances with the other works.

4.3.1. Encrypted Image Dissimilarity

When encrypting images, it is essential to ensure that the encrypted images obtained after multiple encryption processes of the same image exhibit dissimilarity. Such dissimilarity enhances the security of encryption algorithms, preventing attackers from deducing the encryption algorithm or decrypting the key by analyzing the similarities between multiple encrypted images. Ensuring that the outcomes of each encryption process differ heightens the challenge for attackers attempting to speculate and analyze the encryption algorithm. This dissimilarity also ensures that, even if attackers have information about multiple encrypted images, they cannot deduce the content of the original image. Therefore, to ensure the security of encryption algorithms, every encryption of the same image should produce different encrypted images to enhance the algorithm's resistance to analysis and decryption, ensuring the security and confidentiality of encrypted images.

The peak signal-to-noise ratio (PSNR) serves as a quantitative indicator for measuring the dissimilarity between images, as expressed in Equation (22), where MAX represents the maximum possible pixel value, and the mean squared error (MSE) denotes the average squared difference between the original image and the target image. MSE is commonly employed to assess the performance of regression models and compare the accuracy of various models. Furthermore, PSNR is measured in decibels (dB), with higher PSNR values signifying a better image quality and more remarkable similarity. Generally, a PSNR value

above 30 dB indicates minor image differences, whereas a PSNR value below 20 dB suggests more pronounced disparities between images.

$$PSNR = 10 \cdot log_{10} \frac{MAX^2}{MSE} \tag{22}$$

This paper compares two encrypted images to demonstrate the scenario where the original images are the same but the encrypted images are different. It calculates their differences, as shown in Figure 9.

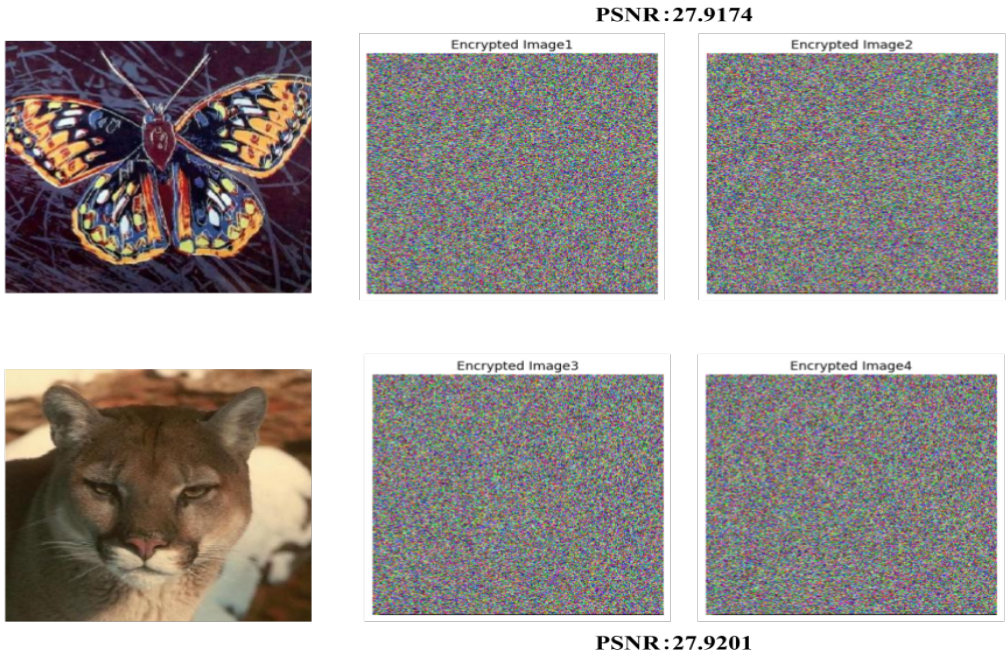

Figure 9. The encrypted images.

In this study, the AES encryption method was used, and the same key was applied during the encryption process to encrypt the first image of a butterfly and the second image of a cougar. These two encryption processes represent the encryption of two uploaded images. These four encrypted images are called Encrypted Image1, Encrypted Image2, Encrypted Image3, and Encrypted Image4, respectively.

By calculating the PSNR between Encrypted Image1 and Encrypted Image2, it was approximately 27.91 dB, while the PSNR between Encrypted Image3 and Encrypted Image4 was approximately 27.92 dB. This indicates that there are differences between the two sets of encrypted images. Furthermore, the study also compared encrypted images of different types. Taking Encrypted Image1 and Encrypted Image3 as examples, the PSNR between these two encrypted images was approximately 27.91 dB, confirming that the PSNR of encrypted images, whether of the same or different types, falls within the range of 27.

Due to the randomness of the encryption process and the role of the encryption key, the two encryption processes will still generate different encrypted results even when uploading the same original image. This dissimilarity is an essential characteristic of encryption algorithms, as it prevents third parties from deducing the rules of the encryption algorithm or decrypting the key based on the encrypted results. Therefore, by demonstrating the scenario where the original images are the same but the encrypted images are different, this study established the existence of dissimilarity during the encryption process. This dissimilarity enhances the security of encryption algorithms, preventing attackers from deducing the encryption algorithm or decrypting the key based on the similarities between multiple encrypted images and further safeguarding against unauthorized access attempts.

4.3.2. Query Unlinkability

"Query unlinkability" is achieved by introducing randomness during the encryption process, ensuring that the same query image will produce different encrypted results in different query contexts. For the encryption method applied to indexing, this study chose the skNN method. The following will provide proof of the query vector unlinkability generated by this method.

According to skNN encrypting the query vector with Equation (2), the query vector is extended to assign the random numbers in the range of $0 \sim r_2$, denoted by δ_q, and in the range of $0 \sim r_2$, denoted by θ_q. With Equation (3), the extended vector is added to a noise vector $\vec{e_q}$, which is in the range of $0 \sim r_1$. Thus, the encrypted query request differs from and is independent of previous or subsequent requests.

This design ensures that, under the observation of a CSP, it is impossible to determine whether two query feature vectors originate from the same image. In other words, the CSP cannot deduce or guess hidden content within query requests by analyzing the encrypted results of query vectors. This proof demonstrates that the method adopted in this research effectively resists known background attacks, thereby preserving the unlinkability between queries and enhancing the system's overall security.

5. Conclusions

This study aimed to address the limitations of existing solutions in encrypted image retrieval, explicitly focusing on image encryption and retrieval in cloud environments, and has designed a fine-grained encrypted image retrieval scheme. Firstly, the VGG-16 module of CNN is employed to extract features from the local regions of images, which is more accurate in describing image content than traditional feature extraction methods, effectively improving retrieval precision. In experiments, this study compared the proposed scheme with conventional local and global feature extraction methods (such as CLD and EHD), validating its effectiveness, with the best retrieval precision achieved under 128-dimensional feature vectors and various dimensionality reduction techniques.

Secondly, this scheme supports the search for small images within larger images. It achieves this by utilizing object detection techniques, specifically YOLOv5, to detect objects within the original image and extract them as local images. These local images' feature vectors are encrypted, and similarity matching is performed in the cloud, enabling encrypted small image retrieval within larger images.

Finally, fine-grained access control is implemented using role-based polynomial access control technology. The access structure for each local image is determined through the construction of role polynomials, with different access roles assigned to different users. If a user is an authorized party for a specific local image, the computation result equals the root of the role polynomial. Otherwise, the querying user is considered unauthorized, effectively ensuring whether a user is an authorized party for a particular local image and thereby guaranteeing access security.

This research contributes to retrieval precision, small image searching within larger images, and fine-grained access control. This research outcome is believed to promote the development of encrypted image retrieval technology and provide valuable references and guidance for research and applications in related fields.

Author Contributions: Methodology, Y.-H.C. and M.-C.H.; formal analysis and validation, M.-C.H.; funding acquisition, Y.-H.C.; writing—original draft preparation, Y.-H.C. and M.-C.H.; writing—review and editing, Y.-H.C. and M.-C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology of the Republic of China, Taiwan, under Grant MOST 110-2221-E-182-026-MY3, and in part by the Kaohsiung Chang Gung Memorial Hospital, with grant numbers CMRPD3N0011 and CMRPD3P0011.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://data.caltech.edu/records/mzrjq-6wc02.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Research, T.M. Defending The Expanding Attack Surface, 2022. Available online: https://www.trendmicro.com/vinfo/us/security/research-and-analysis/threat-reports/roundup/defending-the-expanding-attack-surface-trend-micro-2022-midyear-cybersecurity-report (accessed on 31 August 2022).
2. Reidenberg, J.R. Resolving conflicting international data privacy rules in cyberspace. *Stan. Law Rev.* **1999**, *52*, 1315. [CrossRef]
3. Li, Y.; Ma, J.; Miao, Y.; Li, H.; Yan, Q.; Wang, Y.; Liu, X.; Choo, K.K.R. DVREI: Dynamic Verifiable Retrieval over Encrypted Images. *IEEE Trans. Comput.* **2021**, *71*, 1755–1769. [CrossRef]
4. Li, Y.; Ma, J.; Miao, Y.; Wang, Y.; Yang, T.; Liu, X.; Choo, K.K.R. Traceable and controllable encrypted cloud image search in multi-user settings. *IEEE Trans. Cloud Comput.* **2020**, *10*, 2936–2948. [CrossRef]
5. Tong, Q.; Miao, Y.; Chen, L.; Weng, J.; Liu, X.; Choo, K.K.R.; Deng, R. VFIRM: Verifiable Fine-Grained Encrypted Image Retrieval in Multi-owner Multi-user Settings. *IEEE Trans. Serv. Comput.* **2021**, *15*, 3606–3619. [CrossRef]
6. Yang, T.; Ma, J.; Miao, Y.; Wang, Y.; Liu, X.; Choo, K.K.R.; Xiao, B. MU-TEIR: Traceable Encrypted Image Retrieval in the Multi-user Setting. *IEEE Trans. Serv. Comput.* **2022**, *16*, 1282–1295. [CrossRef]
7. Yuan, J.; Yu, S.; Guo, L. SEISA: Secure and efficient encrypted image search with access control. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 2083–2091.
8. Shen, M.; Cheng, G.; Zhu, L.; Du, X.; Hu, J. Content-based multi-source encrypted image retrieval in clouds with privacy preservation. *Future Gener. Comput. Syst.* **2020**, *109*, 621–632. [CrossRef]
9. Vikhar, P.; Karde, P. Improved CBIR system using edge histogram descriptor (EHD) and support vector machine (SVM). In Proceedings of the 2016 International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India, 18–19 November 2016; pp. 1–5.
10. Rejeb, I.B.; Ouni, S.; Zagrouba, E. Intra and inter spatial color descriptor for content based image retrieval. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
11. Hsu, C.Y.; Lu, C.S.; Pei, S.C. Image feature extraction in encrypted domain with privacy-preserving SIFT. *IEEE Trans. Image Process.* **2012**, *21*, 4593–4607. [PubMed]
12. Li, M.; Zhang, M.; Wang, Q.; Chow, S.S.; Du, M.; Chen, Y.; Lit, C. InstantCryptoGram: Secure image retrieval service. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 2222–2230.
13. Zhang, L.; Jung, T.; Liu, K.; Li, X.Y.; Ding, X.; Gu, J.; Liu, Y. Pic: Enable large-scale privacy preserving content-based image search on cloud. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *28*, 3258–3271. [CrossRef]
14. Perronnin, F.; Liu, Y.; Sánchez, J.; Poirier, H. Large-scale image retrieval with compressed fisher vectors. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3384–3391.
15. Li, X.; Xue, Q.; Chuah, M.C. Casheirs: Cloud assisted scalable hierarchical encrypted based image retrieval system. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
16. Li, Y.; Ma, J.; Miao, Y.; Liu, L.; Liu, X.; Choo, K.K.R. Secure and verifiable multikey image search in cloud-assisted edge computing. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5348–5359. [CrossRef]
17. Li, Y.; Ma, J.; Miao, Y.; Wang, Y.; Liu, X.; Choo, K.K.R. Similarity search for encrypted images in secure cloud computing. *IEEE Trans. Cloud Comput.* **2020**, *10*, 1142–1155. [CrossRef]
18. Amitha, I.; Narayanan, N. Object retrieval in images using SIFT and R-CNN. In Proceedings of the 2020 International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 13–14 February 2020; pp. 1–5.
19. Tao, J.; Gu, Y.; Sun, J.; Bie, Y.; Wang, H. Research on vgg16 convolutional neural network feature classification algorithm based on Transfer Learning. In Proceedings of the 2021 2nd China International SAR Symposium (CISS), Shanghai, China, 3–5 November 2021; pp. 1–3.
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
21. Sun, L.; Hu, G.; Chen, C.; Cai, H.; Li, C.; Zhang, S.; Chen, J. Lightweight Apple Detection in Complex Orchards Using YOLOV5-PRE. *Horticulturae* **2022**, *8*, 1169. [CrossRef]
22. Zhang, J.; Zhang, J.; Zhou, K.; Zhang, Y.; Chen, H.; Yan, X. An Improved YOLOv5-Based Underwater Object-Detection Framework. *Sensors* **2023**, *23*, 3693. [CrossRef] [PubMed]
23. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
24. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.

25. Sehgal, S.; Singh, H.; Agarwal, M.; Bhasker, V. Data analysis using principal component analysis. In Proceedings of the 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), Greater Noida, India, 7–8 November 2014; pp. 45–48.
26. Xia, Z.; Wang, L.; Tang, J.; Xiong, N.N.; Weng, J. A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 318–330. [CrossRef]