



EdgeShield: Attack resistant secure and privacy-aware remote sensing image retrieval system for military and geological applications using edge computing

Ajitesh M¹ · Deekshith M¹ · Arun Amaithi Rajan¹ · Vetriselvi V¹ · Hemanth D¹

Received: 13 October 2023 / Accepted: 22 February 2024 / Published online: 21 March 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

The increasing production and processing of image data, especially in remote sensing applications, has raised concerns regarding image security, privacy, and efficient retrieval as it is most widely used in sensitive applications. In this article, to address these challenges, a novel privacy-preserving content-based image retrieval (PPCBIR) system has been proposed that leverages a trusted edge computing layer that performs image encryption and feature extraction tasks, reducing the processing overload on user devices and bolstering system efficiency. Feature extraction harnesses the MobileNetV2 deep learning model, which enables the extraction of intricate visual features, enhancing image retrieval accuracy in the presence of high inter-class similarity in the dataset. Furthermore, the system has been deployed in a distributed storage environment, ensuring image availability even during server outages. The proposed system also incorporates trusted third-party auditing (TPA) as a means to verify the integrity of images during the storage and retrieval processes. The presence of TPA plays a crucial role in maintaining the reliability and trustworthiness of the stored images. The proposed system achieves a high mean Average Precision (mAP) of 0.889, surpassing existing PPCBIR systems. Overall, the system prioritizes image retrieval performance, privacy, availability, and integrity, making it suitable for processing remote sensing image data efficiently and securely.

Keywords Remote sensing images · CBIR · Privacy preservation · Edge computing · Distributed cloud · Third party auditing

Introduction

In recent years, the exponential growth in digital media and image-based services has led to a significant increase in the production and processing of image data. Among various image types, remote sensing images have gained prominence due to their widespread use in applications such as environmental monitoring, urban planning, and agriculture (Wen et al. 2023). Remote sensing images, however, pose unique challenges due to their high inter-class similarity, requiring advanced feature extraction methods to discern complex patterns and structures. To retrieve relevant information from

vast collections of remote sensing images, Content-based Image Retrieval (CBIR) has emerged as a crucial approach (Kapoor et al. 2021). CBIR enables users to search for images based on their content, such as texture, color, and shape, rather than relying solely on textual descriptions or tags. This method is particularly useful in the context of remote sensing images, where visual features play a crucial role in determining image similarity and relevance.

Alongside the need for efficient image retrieval, the confidentiality and integrity of remote-sensing images must be ensured. With the widespread adoption of cloud computing for flexible and cost-effective storage, concerns about image security and privacy have become paramount (Qin et al. 2018; Tanwar et al. 2018). The sensitive nature of remote sensing data requires robust security measures to protect against unauthorized access, tampering, or data breaches. Secure CBIR schemes can be classified into image encryption-based CBIR schemes and feature encryption-based CBIR schemes (Ma et al. 2022). The former involves users uploading encrypted images and a searchable index

Communicated by: H. Babaie

✉ Amaithi Rajan Arun
arunamaithirajan@gmail.com

¹ Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai 600025, India

to the cloud where the features are extracted from the encrypted image for similarity estimation in Top K search (Tanwar et al. 2022). In contrast, the latter involves the user uploading the encrypted image and the feature vector of the original image. As the features are extracted from the original image, more detailed and quality features can be extracted thereby providing better retrieval performance. However, these approaches are limited to a single centralized server strategy and do not address server outages and other issues faced when using a single storage setup. To address the aforementioned challenges, this paper proposes a comprehensive privacy-preserving image retrieval system specifically designed for remote sensing images and distributed cloud storage environments. The system integrates deep learning-based feature extraction methods with secure storage, retrieval techniques, secret image sharing, and an edge computing layer.

The proposed system employs advanced deep-learning networks such as MobileNetV2 to extract complex and discriminative features from remote-sensing images, enabling accurate and meaningful image retrieval. To safeguard the confidentiality and integrity of the images, encryption techniques are applied, ensuring that only authorized users can access and manipulate the data. The system utilizes a hybrid encryption model that combines spatial and frequency domain encryption techniques, enhancing both security and computational efficiency. In addition to secure image storage, the system incorporates secret image sharing (SIS) to ensure image privacy and availability in a distributed storage environment. This method divides the original image into multiple encrypted messages or "shadows", for reduced storage utilization by avoiding data redundancy, and allocates them to different cloud servers. Even in the event of server outages, the original image can be reconstructed from an arbitrary collection of shadows, ensuring continuous image availability.

Furthermore, considering the resource constraints on user devices and the computational overload posed by deep learning feature extraction, an edge computing layer is introduced. The nearest edge node acts as a proxy for the user, performing tasks such as image encryption, feature extraction, and image retrieval processes on behalf of the user. This edge layer reduces the cost of data transfer to the cloud, minimizes data exposure, and improves overall computational efficiency, speed, and reliability. Figure 1 depicts the planned system's overall layout.

Major contributions in this article are,

- For better retrieval accuracy, a Feature encryption-based CBIR scheme involving MobilenetV2-based deep learning feature extraction techniques
- To reduce the processing overload of the user, a trusted edge computing layer is proposed which performs encryption and feature extraction tasks

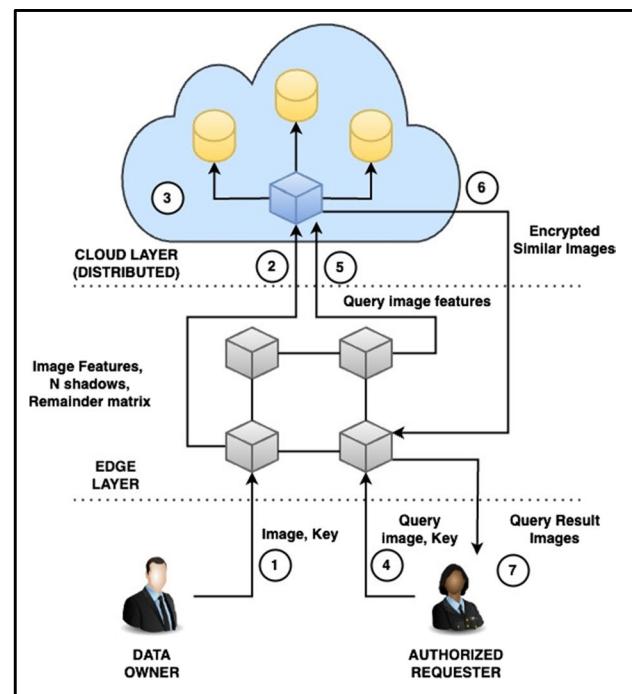


Fig. 1 General System Model

- To ensure availability, the distributed cloud environment is utilized
- To achieve data integrity in PPCBIR, TPA has been deployed

The rest of the article is structured as follows. "Related work" section discusses the related works. The background information required to understand the proposed system has been discussed in "Background" section. "Proposed system" section briefs the proposed EdgeShield for PPCBIR. "Experimental results" section shows the experimental setup and results. Security and retrieval performance analysis are detailed in "Performance analysis" section. "Conclusion and future works" section summarizes the article with the conclusion and future work.

Related work

This section has been split into 2 subsections as follows: Privacy-preserving image retrieval techniques and Trusted storage in the cloud.

Privacy-preserving image retrieval

This subsection discusses the existing PPCBIR methods proposed in recent years. A secure content-based image retrieval system based on the Local Binary Pattern (LBP) and Bag of

Words (BoW) model has been proposed by Xia et al. (2021). The image content is protected by the big-block permutation, 3×3 small block permutation within big blocks, pixel permutation within 3×3 small blocks, and polyalphabetic cipher. The use of polyalphabetic cipher improves confidentiality and causes no degradation in terms of retrieval accuracy since the substitution tables are generated by the order-preserving encryption. Without any interaction from the data owner, the cloud server computes the LBP histograms directly from the encrypted big blocks. The feature vector is then calculated using the local LBP histograms and the BOW model, yielding effective retrieval accuracy. Despite the system's robustness and highly efficient image retrieval, it has a large computational overhead and constrained availability. Qin et al. (2014) proposed SecSIFT, a high-performance privacy-preserving Scalar Invariant Feature Transform (SIFT) feature detection system. The proposed system distributes the computation procedures of SIFT to a set of independent, cooperative cloud servers, and keeps the outsourced computation procedures as simple as possible, thus enabling implementation with practical computation and communication complexity. A privacy-preserving image retrieval in the distributed environment was proposed by Zhou et al. (2022), where a gateway can encrypt the digital images that are collected from smart devices and upload the encrypted images to multiple cloud servers. The authorized researcher can use similarity retrieval to identify images that are comparable to the query by searching through these encrypted images. The suggested method resolves the issue of maintaining image availability while also guaranteeing image security and similarity search on encrypted images. They created a two-stage image encryption system based on a combination of image encryption that supports similarity searches on encrypted domains and private image sharing, which significantly increased image security and availability while concurrently preserving retrieval accuracy.

A PPCBIR scheme not only efficiently retrieves images based on the primitive visual image feature but simultaneously imposes security aspects to its transmission. In the system proposed by Sengar and Kumar (2022), The owner used a logistic map to generate an encrypted image database and an encrypted image feature database, which were then moved to a centralized database. The user then uses the same method to construct the encrypted query image feature vector and sends it to a central database. To find a few of the most similar encrypted images that will be sent to the user side, a similar image search has been conducted across the encrypted feature image database in a centralized database. The user will now decode those received images using the key they obtained from the owner side to obtain the final result. A secure CBIR in the Cloud with Key Confidentiality (Li et al. 2020) implements kNN for searching in a dataset. It makes the supposition that each entity in this

system can be partially trusted. The k-means approach is used to streamline the descriptors of massive databases with over 10,000 images in order to speed up search performance. In the searching phase, trapdoor verification is also used to make sure the trapdoor is valid. The technology is very adaptable and offers improved security and speedy recovery.

The field of CBIR for satellite images has witnessed significant advancements in recent years. However, there is a continuous need for improvement in retrieval accuracy and computational complexity. High spatial resolution remote sensing images come with distinct layers, clear textures, and rich spatial information. Mingchang et al. (2019) proposed a method to realize scene-level classification of high spatial resolution images by extracting the depth features of high-resolution remote sensing images using a residual learning network (ResNet), and low-level features, including color moment features and gray-level co-occurrence matrix features. They were then used to construct various scenes semantic features of high-resolution images and created a classification model with the training support vector machine (SVM). A similar image detection method involving multi-level features has been proposed by Zhang et al. (2019) using ground objects' multi-features, such as color, texture, and shape. Though this method has a high accuracy rate, if the non-cloud area is highly reflective, irregular, and as smooth as the cloud area, it is easily misidentified as the cloud area by the proposed detection method. Sunitha and Sivarani (2021) introduced an efficient CBIR system that leverages Weighted Brownian Motion-based Monarch Butterfly Optimizations (WBMMBO). The proposed approach incorporates various steps, including contrast enhancement using the Adjusted Intensity-based Variant of Adaptive Histograms Equalization (AIVA), feature extraction (LPDF, DCD, BoVW, SF, and BRIEF), feature selection with WBMMBO, similarity computation using MSSIM, and threshold-centered checking. Preliminary results indicate notable enhancements in precision and recall rates, along with improved computational efficiency. Traditional convolutional neural network (CNN) models have exhibited limitations in terms of training time and high-dimensional feature outputs. In the field of content-based remote sensing image retrieval. To address this challenge, a study by Hou et al. (2020) explores the use of the MobileNets model and proposes a fine-tuning approach by adjusting the dimensions of the final fully connected layer to learn low-dimensional representations. Experimental results demonstrate that the MobileNets model achieves superior retrieval performance compared to other CNN models, such as ResNet152, in terms of retrieval accuracy and training speed. This offers a simple yet effective solution to improve retrieval performance. One disadvantage is that the MobileNets-based CBIR approach has relatively high computational requirements, especially during the fine-tuning process.

TDHPIR is an entirely novel triplet deep CNN hashing-based privacy-preserving image retrieval method put forth by Zhang et al. (2020) to enhance the efficiency of image retrieval securely in the cloud. A triplet deep CNN model was proposed to simultaneously learn visual representations and hash codes for higher-quality hash code learning. In addition, a unique hierarchical bit-scalable hash code-based S-Tree called the H2S-Tree is created to speed up hashing-based image search. Though image search is efficient and image retrieval accuracy is high, it has limited robustness against adversarial attacks.

Cloud data integrity verification

This subsection is dedicated to analyzing the existing methods to secure cloud data integrity. Kumar et al. (2021) proposed a model that utilizes lightweight cryptographic systems and hashing techniques to ensure data security and integrity when auditing outsourced data from cloud service providers. The proposed system focuses on data reliability through correctness verification and error recovery analysis. It also emphasizes the time efficiency of the system compared to other models. Additionally, they claim resistance against known cryptanalytic attacks and showcase the secure and highly efficient performance of the proposed system, through extensive compression techniques. Garg et al. (2023) proposed an effective and reliable data integrity verification scheme based on Schnorr signatures. Schnorr signatures offer the advantage of linear signature verification equations and the ability to perform batch verification on multiple blocks. While existing schemes rely on Boneh Lynn and Shacham (BLS) or RSA signature schemes, the proposed scheme stands out for its high efficiency and security. Experimental results demonstrate that the proposed scheme significantly reduces verification computation costs. Similarly, an effective public auditing protocol has been proposed by Bhavyasree et al. (2021), where the system involves a third-party auditor (TPA) that evaluates the precision of cloud data without the need for complete data retrieval or adding extra online load for users and servers. The proposed system ensures data confidentiality, integrity, and proper storage. The user encrypts file blocks using AES algorithms, generates MD-5 hash values for each block, and creates an AES signature for the entire file. The cloud server stores the encrypted file blocks. When a user requests TPA auditing, the TPA directly obtains the encrypted files, generates MD-5 hash values, and compares its AES signature with the user's signature. The validation process determines if the data is intact and not compromised. The verified data integrity status is then communicated back to the user, providing assurance and trust in the security of cloud data.

In third-party auditing systems, various popular hashing algorithms are utilized to ensure data integrity and security.

The widely used MD5 generates a 128-bit hash value but is vulnerable to collision attacks, limiting its suitability for cryptographic applications. For enhanced security, SHA-512 from the SHA-2 family is preferred, providing a robust 512-bit hash value with resistance against collisions. Perceptual hashing techniques, such as visual hashing, are pivotal in multimedia content analysis, generating compact hashes that capture image similarity for tasks like search, duplicate detection, and copyright protection. Another prominent hashing algorithm is BLAKE2 (Kumar et al. 2017), known for its high performance and security. An enhanced version of the original BLAKE function, BLAKE2 offers various output sizes and finds applications in data integrity verification, password hashing, and message authentication. Additionally, the emerging BLAKE3 algorithm proves to be an excellent choice. It combines innovative ideas from different hashing algorithms, providing exceptional performance, robust security, and resistance against various cryptographic attacks. With its speed, versatility, and ability to generate hash values of variable lengths, BLAKE3 is increasingly recognized as a favorable choice for ensuring data integrity and security in third-party auditing systems.

This literature survey explores recent advancements in Privacy-Preserving Content-Based Image Retrieval (PPCBIR) for remote sensing images and diverse strategies for ensuring data integrity in cloud computing, which are required in sensitive areas such as the military. In PPCBIR, various approaches, including encryption-based security measures and deep learning techniques, prioritize image security while maintaining retrieval accuracy and efficiency. The survey highlights the importance of privacy in distributed environments and acknowledges ongoing challenges, reflecting the dynamic landscape of the field. In data integrity for cloud computing, lightweight cryptography, efficient verification schemes, and public auditing protocols are discussed. Robust hashing algorithms like BLAKE-3 are recommended, addressing the evolving security requirements in cloud-based data storage and retrieval. This survey encompasses privacy, security, efficiency, and adaptability, addressing critical challenges in secure image retrieval and data integrity in cloud environments. From the literature survey done, it is vivid that distributed and secure remote sensing image retrieval is required in sensitive areas such as the military.

Background

This section provides the background technical information to understand the proposed system.

Hybrid image encryption

Over the past few decades, several encryption schemes have been proposed to address security concerns. However, these schemes often suffer from vulnerabilities such as time

inefficiency and weak security. The work of Shafique et al. (2021), aims to provide the highest level of security for digital data by incorporating chaos to scramble the rows and columns of the plaintext image. Additionally, a noisy image is generated using a chaotic logistic map and carefully selected initial conditions based on thorough analysis.

The mathematical form of the logistic map:

$$X_{i+1} = W * X_i (1 - X_i) * (2 + X_i)$$

Initial conditions:

$$X \subseteq (0, 1)$$

$$W \subseteq [1.42, 1.60]$$

To reduce encryption computational time, a Discrete Wavelet Transform (DWT) is employed, focusing only on encrypting the low-frequency bands since they contain the majority of the plaintext information. Their proposed encryption algorithm can successfully decrypt the plaintext image with minimal information loss, although the content of the plaintext image can still be visually perceived. Decryption is the same as encryption, done in reverse order. Our work uses this algorithm due to its low time complexity, without any compromise on the encryption strength, striking a balance between security and efficiency. Figure 2 depicts the flow of encryption.

Feature extraction using MobileNetV2 model

Image feature extraction is a technique used to capture relevant information and patterns from images (Rajath et al. 2023). It involves identifying distinctive attributes or characteristics that represent the visual content of an image. Traditional methods relied on manually designing and selecting features, but deep learning-based methods have revolutionized this

field. Deep learning utilizes neural networks, particularly convolutional neural networks (CNNs), to automatically learn and extract features from images. These deep learning models can capture both low-level visual patterns and high-level semantic representations, enabling superior performance in tasks such as image classification, object detection, and image recognition. By learning from large-scale datasets, deep learning models generalize well and produce transferable features applicable to various domains. This advancement in image feature extraction has greatly improved the capabilities of computer vision systems in analyzing and understanding visual data. In this article, the MobileNetV2 model has been used for feature extraction (Fig. 3).

Secret image sharing scheme

A robust method for image secret sharing that combines two k-out-of-n secret sharing schemes: i) Shamir's secret sharing scheme, and ii) matrix projection secret sharing scheme (Bai 2006). The proposed technique enables the division of a colored secret image into n image shares (or shadows) and significantly reduces the size of shares, ensuring that:

- i. any k image shares (where $k \leq n$) are adequate to reconstruct the secret image without any loss, and
- ii. any $(k - 1)$ or fewer image shares do not contain sufficient information to unveil the secret image.

For encrypted RGB images, SIS is performed on every channel separately. The resulting remainder matrices and shadows of the three channels are merged together.

To share an image $N \times N$ across n cloud servers, the image shares are constructed in the following steps: (Secret Sharing).

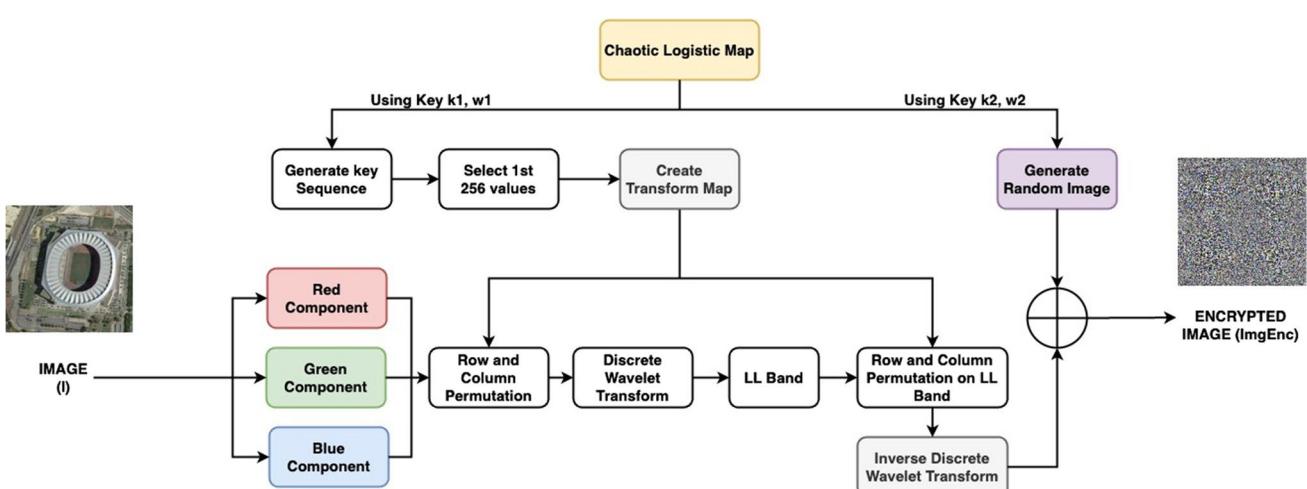


Fig. 2 Hybrid Image Encryption Model

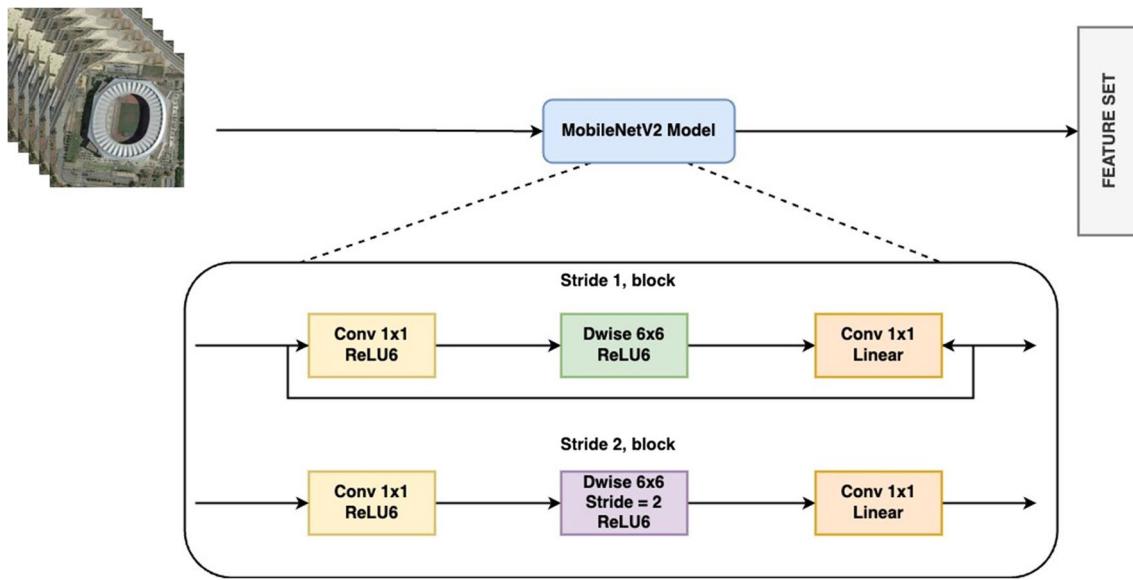


Fig. 3 MobileNetV2 Model

1. Split image I into constituent R , G , and B channels.
2. Construct n shadow vectors v_i and remainder matrix R for each channel.

A = random matrix $n \times k$

$$A_{\text{proj}} = (A(A'A)^{-1}A')(\bmod p)$$

$$R = (I - A_{\text{proj}})(\bmod p)$$

n linearly independent $k \times 1$ random vectors X_i

shadow $v_i = (A \times X_i)(\bmod p)$, for $1 \leq i \leq n$

3. Merge the three channels of each shadow and the remainder matrix
4. Distribute each shadow to a unique cloud server and make the remainder matrix publicly known.

To reconstruct the image from the k shares, stored in any k cloud servers, the following steps are performed: (Secret recovery).

1. Construct a matrix B using any k shadows

$$B = [v_1 \ v_2 \ \dots \ v_k]$$

2. Reconstruct the original image using the projection of matrix B and the remainder matrix R

$$B_{\text{proj}} = (B(BB)^{-1}B)(\bmod p)$$

$$\text{reconst_img} = (B_{\text{proj}} + R)(\bmod p)$$

This technique protects the secret image from loss, theft, or corruption and is proven to be dependable, secure, and efficient. This methodology has a number of advantages over existing image secret-sharing techniques, including the capacity to

process data in real-time, a high rate of compression for image share sizes, and strong secret image protection.

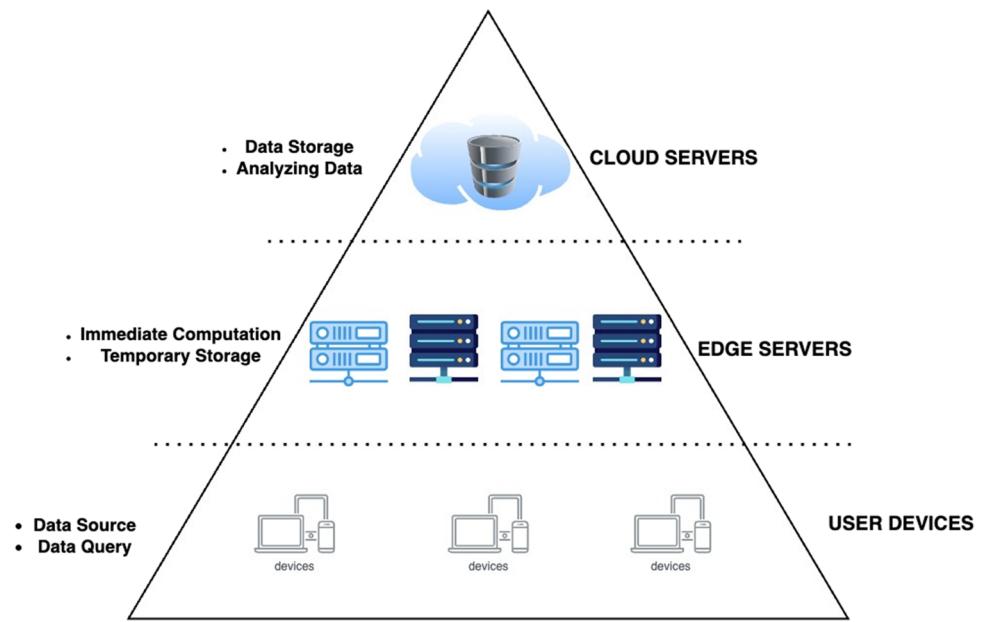
Edge computing

Edge computing is a decentralized computing paradigm that brings computing resources closer to data sources and end-users (Pérez et al. 2022). It addresses the need for real-time processing, low latency, and improved data privacy and security. By processing and analyzing data at the network edge, edge computing reduces reliance on centralized cloud infrastructure, enhances performance, and enables applications in areas such as IoT, autonomous vehicles, and smart cities. It offers localized computing capabilities, reduces data transfer and network congestion, and provides greater control over sensitive data. Edge computing is a complementary approach to cloud computing, enabling faster, privacy-enhanced, and low-latency processing at the edge of the network (Fig. 4).

Distributed cloud

Image storage in distributed cloud environments employs a master–slave architecture to provide scalable, fault-tolerant, and efficient solutions for managing image data. The master cloud acts as the central authority, managing storage metadata and maintaining records of relevant image information. The image data is distributed across multiple slave cloud servers, allowing for seamless scalability as the data volume grows. Efficient retrieval and access are facilitated through parallel processing and retrieval from multiple servers simultaneously. The master–slave architecture provides

Fig. 4 The three-tier architecture of edge computing



high availability, fault tolerance, and scalability, making it an ideal approach for managing and storing large volumes of image data in distributed cloud environments.

Third party auditor

Data corruption can be detected immediately by verifying the integrity of the data stored in the cloud. Third Party Auditor (TPA) acts as an independent and secure third-party verifier that ensures the integrity and authenticity of the data stored in the cloud, and provides an additional layer of security and trust to the system. TPA verifies the integrity of the stored data by computing its hash value and comparing it

with the hash value of the original data, stored in its secure database (Chakraborty et al. 2018).

Proposed system

The proposed system is composed of user, edge, and distributed cloud layers as shown in Fig. 5. A master-slave architecture is employed for the distributed cloud environment. The master manages and coordinates the storage in slave clouds. The system has adopted a feature-extraction-based privacy-preserving CBIR scheme, the features of the uploaded image are extracted in the edge node and sent to the

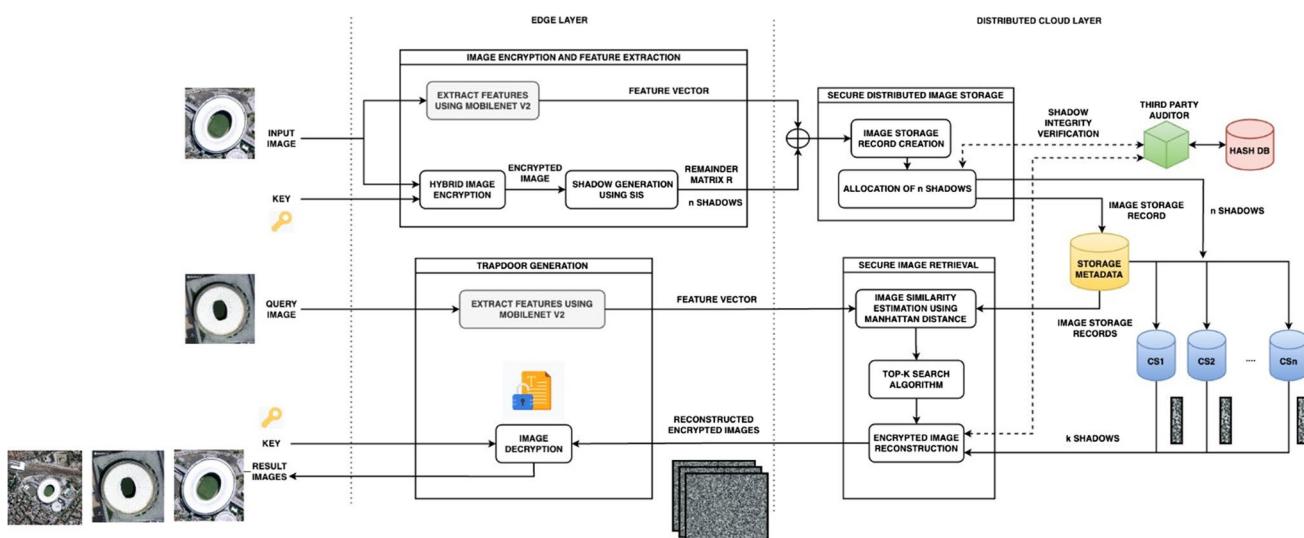


Fig. 5 Architecture Diagram

master. Image encryption, decryption, and shadow generation are performed in the edge node. The Edge node acts as a proxy between the user and the cloud storage. Introducing an edge layer between the user and the cloud environment helps in reducing the computational overhead on the user end. Deep learning-based feature extraction techniques require high amounts of computing power to complete their tasks quickly to reduce the overall retrieval and storage latencies. Trusted edge nodes with high computing and GPU resources achieve reduced request latencies, meanwhile taking the load off the users and performing their tasks. During image upload storage, the edge node performs encryption over the user-uploaded image, performs shadow generation using SIS, extracts feature from the original uploaded image, and finally sends the feature vector, remainder matrix, and shadows to the master cloud. The system's scalability is ensured through edge-based feature extraction for high-resolution images, minimizing data transmission to the cloud. Load balancing across edge nodes optimizes resource usage, preventing congestion during concurrent high-resolution image queries. Dynamic scaling, monitored at the edge, adjusts resources based on user numbers or processing demands, ensuring optimal performance.

A master-slave architecture is instituted for the distributed cloud environment. The master cloud handles the incoming storage and retrieval requests with the help of stored meta-records of the uploaded images. The slave clouds are the actual storage units, each slave storing one shadow of the uploaded image. A TPA is employed to ensure the integrity of the shadows stored and detect attacks that intend to corrupt the data immediately. When the user uploads an image to store in the cloud environment, the uploaded image is sent to the edge node which performs the following functions, i) performs encryption over the user-uploaded image. ii) performs shadow generation using SIS on encrypted user images. iii) Extract features from the user-uploaded image. Then the edge node sends the feature vector, remainder matrix, and shadows to the master cloud. The master cloud creates a meta record for the new image stored and maintains the shadow references. The remainder matrix is stored in the master metaDB. The slave cloud stores the image shadow and updates its logs.

Later during image retrieval, the user uploads a query image and it is sent to the edge node where it extracts the query image features and sends it to the master cloud. The master cloud then performs the top K search based on the query image features, fetches the shadows from appropriate slave clouds, and the stored encrypted image is reconstructed. The top K similar reconstructed encrypted images are sent back to the edge node and it decrypts them and sends the results back to the user. During image retrieval, the edge node extracts the query image features and sends it to the master cloud. Upon receiving the reconstructed encrypted images from the master cloud, the

edge node decrypts and sends them to the authorized requestor. The detailed modular information is given below.

Detailed modular design:

1. $I_{enc} \leftarrow \text{Encryption}(I, k_1, w_1, k_2, w_2)$: This module encrypts the input image I using keys k_1, w_1, k_2, w_2 and outputs encrypted image I_{enc} .
2. $F \leftarrow \text{Feature_Ext}(I)$: This module extracts features from input image I using MobileNetV2 feature extraction method, and outputs a feature vector F .
3. $R, \text{Shadows}[n] \leftarrow \text{Shadow_gen}(I_{enc})$: In this module, n shadows are generated using the Secret Image Sharing (SIS) scheme, where the encrypted image I_{enc} is provided as input. A remainder matrix R and n shadows are produced as output.
4. $\text{Rec}_{\text{DB}} \leftarrow \text{Sec_Storage}(F, R, \text{Shadows}[n])$: This module stores feature vector F , remainder matrix R , and n shadows in the distributed cloud and creates a record Rec_{DB} in the master cloud MetaDB. $\text{Shadows}[n]$ are also sent to TPA for integrity verification.
5. $I_{enc}[K] \leftarrow \text{Sec_Retrieval}(I_q)$: This module returns Top-K similar images $I_{enc}[K]$ from the cloud that is similar to query image I_q . First, the feature vector F is extracted from I_q and then F is compared with the stored feature vectors. Then shadows of top K images with the least Manhattan distance are fetched from the cloud. Encrypted images are reconstructed using those shadows as output.
6. $I[K] \leftarrow \text{Decryption}(I_{enc}[K], k_1, w_1, k_2, w_2)$: The Top-K encrypted images $I_{enc}[K]$ are decrypted using the keys k_1, w_1, k_2, w_2 provided as input. The decrypted images $I[K]$ are provided as output to the user.

The proposed system has been detailed in three subsections based on the objectives: secure storage, secure retrieval, and attack resistance.

Secure image storage

In feature-extraction-based CBIR schemes, encryption, and feature extraction steps are performed earlier and the results are sent to the cloud for storage. The edge node on behalf of the user performs encryption over the user-uploaded image, performs shadow generation using the Secret Sharing of SIS scheme, extracts features from the original uploaded image, and finally sends the feature vector, remainder matrix, and shadows to the master cloud. In the distributed cloud environment, the storage of uploaded images is a primary focus. To achieve space-efficient storage, a secret image-sharing scheme is utilized to generate a remainder matrix and shadows. The SIS scheme generates n shadows, which are stored in n cloud servers, and out of which any k shadows can be used to reconstruct the image with negligible or no data loss. Meanwhile, a

third-party auditor (TPA) is employed to ensure the integrity and authenticity of the shadows stored in slave clouds,

adding an additional layer of security and trust to the system. Algorithm 1 explains the flow of image storage.

Algorithm 1 Secure Image Storage

Input: Feature vector f , Array of n shadows, Remainder matrix R

Output: Updated records in Master Cloud MetaDB

1. Create a record entry for the uploaded image in the MetaDB of the Master Cloud
 2. Store Image id, feature vector f , Remainder matrix R , and references to each shadow
 3. For every shadow:
 - a. Allocate the shadow to one cloud server
 - b. Verify the integrity of the stored shadow
 - c. Update shadow reference in the record entry
-

When an image is uploaded, the master cloud receives the feature vector, remainder matrix, and n shadows. The feature vector and remainder matrix are used to create a new storage meta-record, which serves as the storage metadata for the image. This record includes the feature vector and references to the n shadows stored across multiple cloud servers. Simultaneously, the master cloud sends each shadow to the TPA and the corresponding slave cloud for storage. The slave clouds store the shadows and confirm their successful storage to both the master and TPA. The TPA verifies the integrity of each shadow by computing its hash value

and comparing it with the original hash value stored in its secure database. Algorithm 2 demonstrates the verification process by TPA.

- If all shadows pass the integrity verification, the master cloud stores the storage meta-record, signifying the successful storage of the image.
- If the TPA identifies any malicious activity from a slave cloud, indicating tampering or compromise of a shadow, the master invokes the recovery service.

Algorithm 2 Integrity Verification By TPA

Input: Original shadows, Shadow references[]

Output: is_malicious[]

1. Calculate the hash value of all the shadows, Hash_val_orig[], received from the master.
 2. Hash_val_orig[i] = blake3_hash(shadow[i])
 3. Store Hash_val in hash_db of TPA
 4. Fetch the stored shadows from all the slaves.
 5. Calculate the hash value of the received shadows, Hash_val_Stored[], from the slaves.
 6. For every shadow :
 - If Hash_val_orig[i] == Hash_val_Stored[i] :
 1. is_malicious[slave_i] = false
 - else :
 1. is_malicious[slave_i] = true
 7. Send the results, is_malicious, to the master
-

By combining the SIS scheme, distributed storage across slave clouds, and involvement of a TPA, this system ensures secure, efficient, and distributed storage of images while maintaining data integrity. The TPA's independent verification provides an additional layer of trust, safeguarding against malicious activities and enhancing the overall security of the system.

Secure image retrieval

Secure image search involves retrieving encrypted images similar to a query image while ensuring data integrity and security. The process includes two main steps: Top K search and Image reconstruction. When an authorized requester queries the edge node with a query image, the features of the query image are extracted and sent to the master cloud for similarity estimation. The master cloud estimates the similarity by computing the Manhattan distance between

the query feature vector and stored feature vectors to identify the top-K similar images. The K shadows and remainder matrix of these images are retrieved for encrypted image reconstruction using the Secret Recovery of SIS scheme. Simultaneously, the third-party auditor (TPA) is involved in integrity verification. The master fetches shadows from three available slave clouds and sends the slave cloud IDs to the TPA. The TPA verifies the integrity of the fetched shadows by comparing their hashes with the original shadows' hashes stored in its hash database. If a malicious slave cloud or integrity violation is detected, another shadow is fetched, and parallel recovery is initiated. The K-reconstructed encrypted images are then securely sent back to the requesting edge node. The edge node then performs image decryption and sends the decrypted results to the requester. This combined approach ensures secure image retrieval, integrity verification through TPA, and privacy preservation in a distributed cloud environment.

Algorithm 3 Top – K Image Search

Input: Query image Q

Output: Top K similar images (Encrypted)

1. Extract the feature vector q from the query image Q
 2. Calculate the Manhattan distance between q and the feature vectors of stored images f_i
- $$\text{Manhattan}(A, B) = \sum_i |A_i - B_i|$$
3. Filter the top K images with the least Manhattan distance
 4. For every similar image in top- K :
 - a. Retrieve any k shadows of the image from the total n shadows
 - b. Verify the integrity of the retrieved shadows using TPA
 - c. Reconstruct the encrypted image from the k shadows and the stored Remainder matrix R
 5. Return K reconstructed encrypted images
-

Attack mitigation and recovery system

In the event of an attack triggered by one of the slaves during the storage, the master flags the malicious slave and initiates the slave recovery mechanism as shown in Fig. 6. The recovery mechanism terminates the malicious slave. A new slave is created and the shadow is stored in the new slave. The integrity of the shadow stored in the new slave is verified, and the meta-record is updated and stored in the master meta-db. The recovery mechanism then performs data migration from the uncorrupted backup of the malicious slave to the new slave.

In the case of an attack during image retrieval, the mitigation and retrieval are similar. Here, as soon as a slave cloud is detected to be malicious, a shadow from another slave cloud is fetched and the recovery mechanism is executed in parallel. Overall query request time is unaffected. This provides the security and availability of the data to the user.

Experimental results

Experimental setup

The proposed work is implemented in Python, where Flask API is used for developing the edge node and cloud servers. ReactJS is used for building the frontend, to allow the users to interact with the system. We used Docker and

Docker-compose to deploy the distributed cloud environment. Docker allows us to build a network of master and slave clouds. To evaluate the real-time performance of the system, we deployed on to AWS EC2 cloud service, which hosts Docker. The network of master and slave clouds deployed using Docker behaves similarly to real-world cloud servers. Tensorflow is used for employing pre-trained MobileNetV2 for feature extraction.

Dataset description

We used the Northwestern Polytechnical University's (NWPU) NWPU-RESISC45 dataset, a publicly accessible benchmark for Remote Sensing Image Scene Classification (RESISC). There are 700 images in each of the 45 scene classifications in this dataset's 31,500 total images. With the use of various benchmark datasets, including PatternNet and UC Merced Land Use, the proposed study has been simulated. Datasets are listed in Table 1.

Image storage results

Secure image storage flow is depicted in Fig. 7. The user uploads the image to be stored along with the encryption keys via the user interface, as shown in Fig. 8. The user request is directed to the nearest edge node, where features of the image are extracted and the image is encrypted. Through the SIS scheme, the shadows and remainder

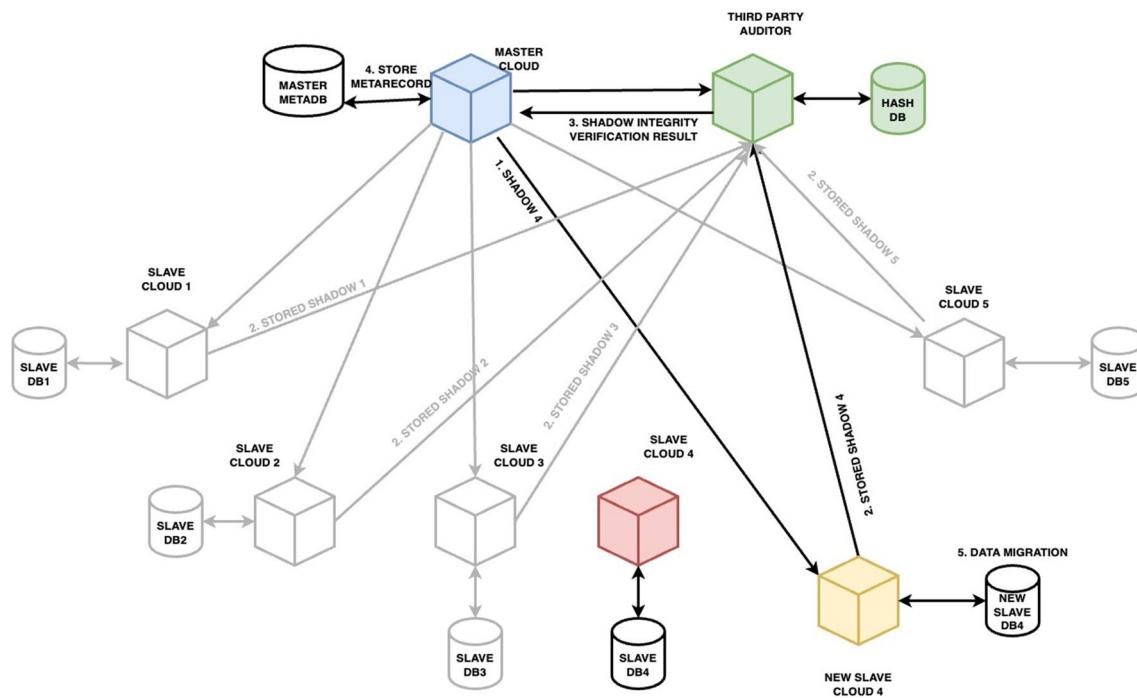
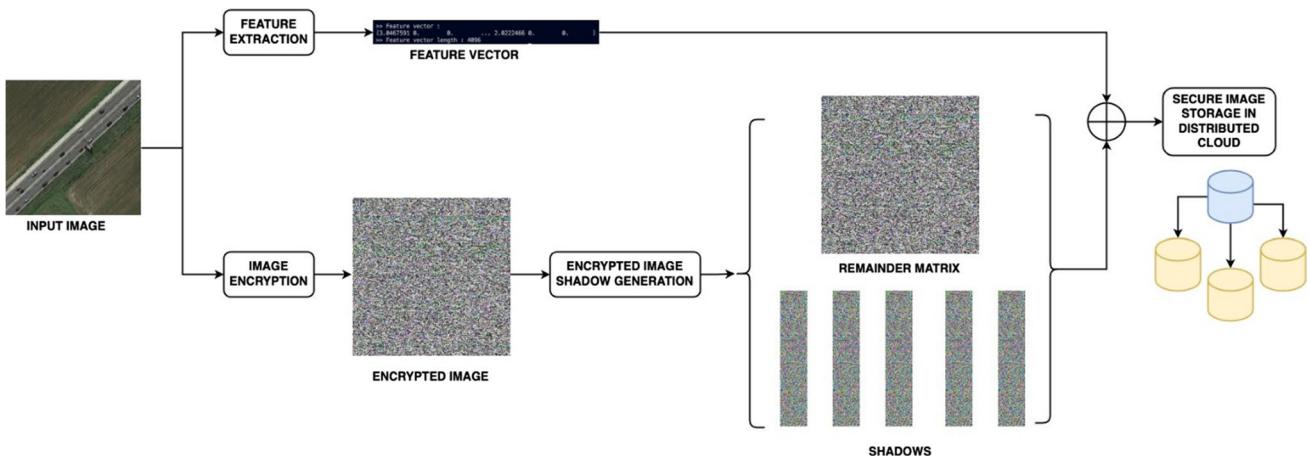
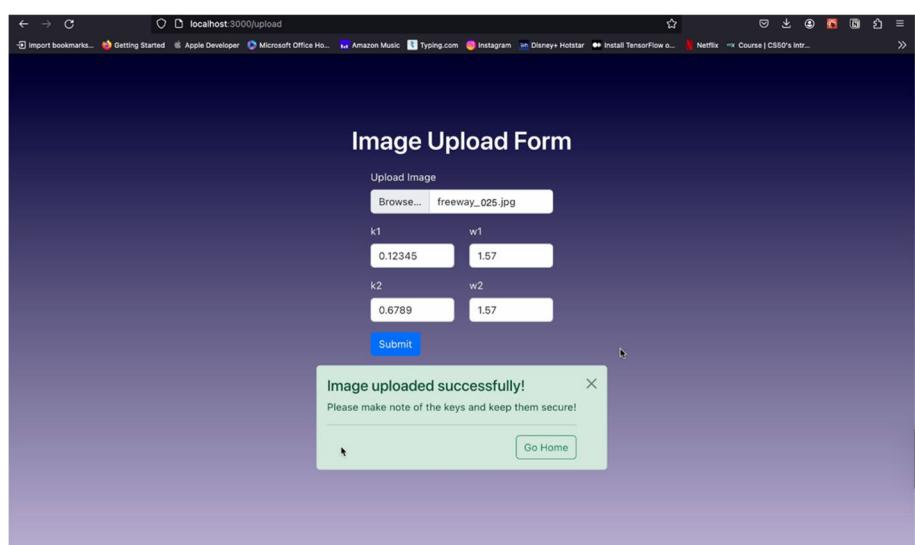


Fig. 6 Attack mitigation and recovery

Table 1 Datasets description

| Dataset | Class | Image number | Images per class | Sources |
|--------------------|-------|--------------|------------------|---|
| NWPU-RESISC45 | 45 | 31,500 | 700 | Google Earth Imagery |
| PatternNet | 38 | 30400 | 800 | Google Earth Imagery |
| UC Merced Land Use | 21 | 2100 | 100 | USGS National Map Urban Area Imagery |

**Fig. 7** Working flow of image storage**Fig. 8** Uploading an image

matrix of the encrypted image are generated. The edge node sends the feature vector, remainder matrix, and 5 shadows to the master cloud for storage. The master cloud then stores the remainder matrix in it and one shadow each, in 5 slave clouds. Master cloud then creates and stores a meta DB record with the image ID, feature vector, reference to the remainder matrix stored in the master, and shadow-slave references. Figure 9 shows the resulting master meta records and slave log records created.

Secure image retrieval results

An authorized requestor with the appropriate encryption key, uploads a query image via the user interface, as shown in Figs. 10 and 11. The user query request is directed to the nearest edge node, where the features of the query image are extracted and sent to the master cloud. In the master cloud, similarity estimation is performed by computing the Manhattan distance between the query features and stored

| image_id | feature_vector | remainder_matrix | shadows |
|-------------|--|--|---------|
| freeway_025 | 0.0 0.4759554862976074 0.9 freeway_025_remainder_img.txt | {0: 'slave-cloud-1:5000', 1: 'slave-cloud-2:5000', 2: 'slave-cloud-3:5000', 3: 'slave-cloud-4:5000'} | |
| image_id | stored_at | | |
| freeway_025 | 2023-03-16 9:51:50 | | |
| | | | |
| | | | |

Fig. 9 The stored master meta-record and slave logs

feature vectors. Then the image IDs of top K ($K=10$) similar images with the smallest Manhattan distance are identified. Then for every similar image, the remainder matrix from the master and 3 shadows from any three available slave clouds are retrieved. The stored encrypted image is reconstructed using the SIS scheme. The top 10 similar images in encrypted form are sent back to the edge node. In the edge node, the images are decrypted and sent back to the user. This overall flow of image retrieval is depicted in Fig. 11.

Performance analysis

In this section, the performance of the system has been evaluated using various metrics. Image security analysis, image retrieval performance analysis based on different feature extraction mechanisms, Attack impact analysis,

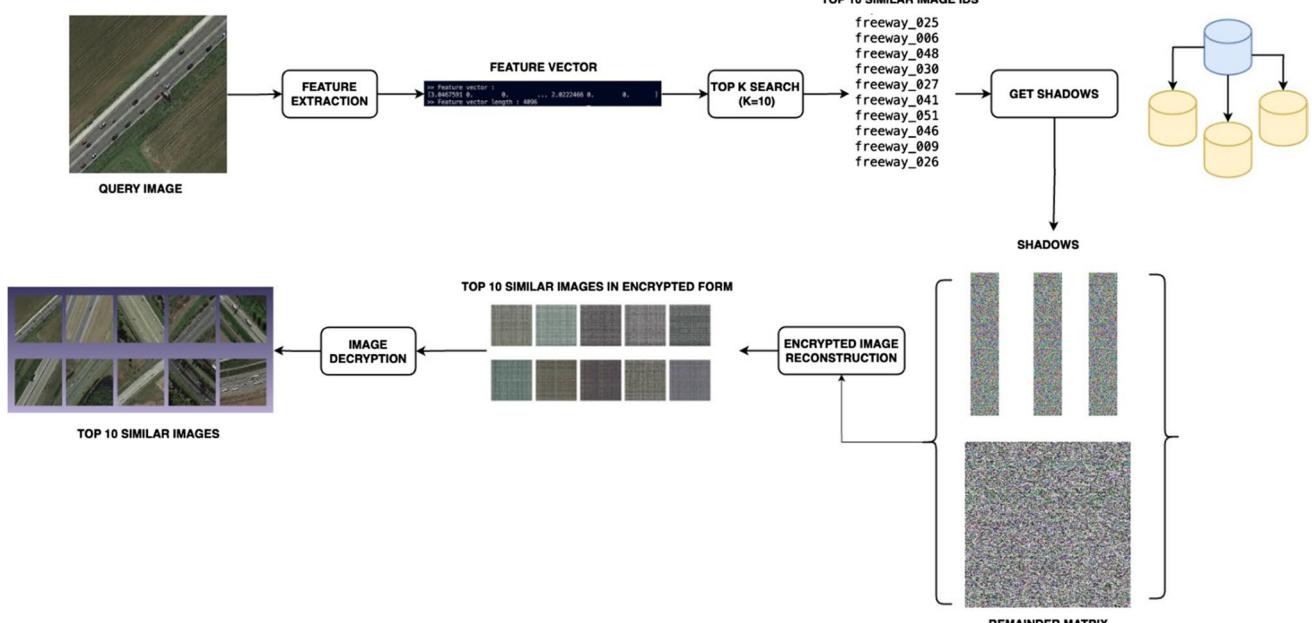
and retrieval time analysis are briefly investigated and presented.

Image encryption security analysis

In this subsection, we have shown that the applied image encryption algorithm is attack-resistant for remote-sensing images. Both histogram and correlation analysis show the secureness of the encrypted image.

Histogram analysis

The pixel distribution of any image can be observed by the histogram analysis (Shafique et al. 2021). Figure 12a shows the pixel distribution in different channels of the original and the encrypted image. For a strong encryption technique, the histogram of the ciphertext image should be uniform

**Fig. 10** Working flow of image retrieval

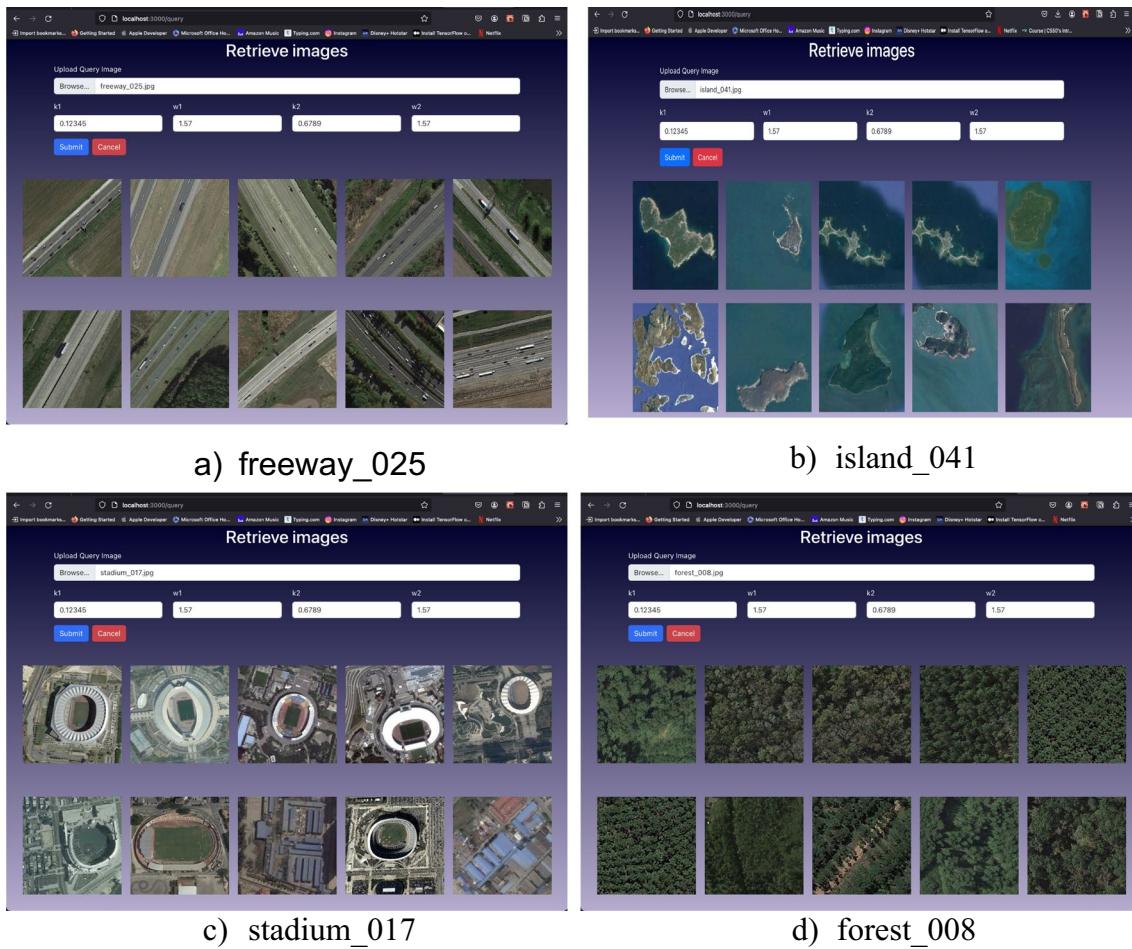


Fig. 11 Sample Query Results

and completely different from the histogram of the original image.

Correlation analysis

Based on the correlation plot analysis of both the plaintext and encrypted images shown in Fig. 12b, it was observed that the encryption algorithm has effectively randomized the pixel values of the encrypted image and eliminated any correlation between adjacent pixels in the image, thus ensuring the confidentiality of the original plaintext image. Therefore, the correlation plot analysis provides strong evidence of the encryption algorithm's effectiveness in ensuring the security of the encrypted image.

Image retrieval performance analysis

The retrieval performance of the PPCBIR scheme can be evaluated by analyzing the search results in terms of precision, recall, and F1 score. Precision is the ratio of the number of similar images recovered to the total number of images recovered, Recall is the ratio of the number of similar images

recovered to the total number of similar images and the F1 Score is the harmonic mean of precision and recall.

$$\text{Precision} = \frac{\text{No. of similar images retrieved}}{\text{Total no. of images retrieved}}$$

$$\text{Recall} = \frac{\text{No. of similar images retrieved}}{\text{Total no. of similar images}}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Mean Average Precision (mAP) measures the effectiveness of a retrieval system by computing the average precision over a set of queries. A high mAP value indicates that the system is successful in retrieving relevant images for the given queries.

$$\text{mAP} = \frac{1}{|Q|} \sum_i AP(q_i)$$

$|Q|$ is the total number of queries in the set.

$AP(q_i)$ is the Average Precision score for the i^{th} query.

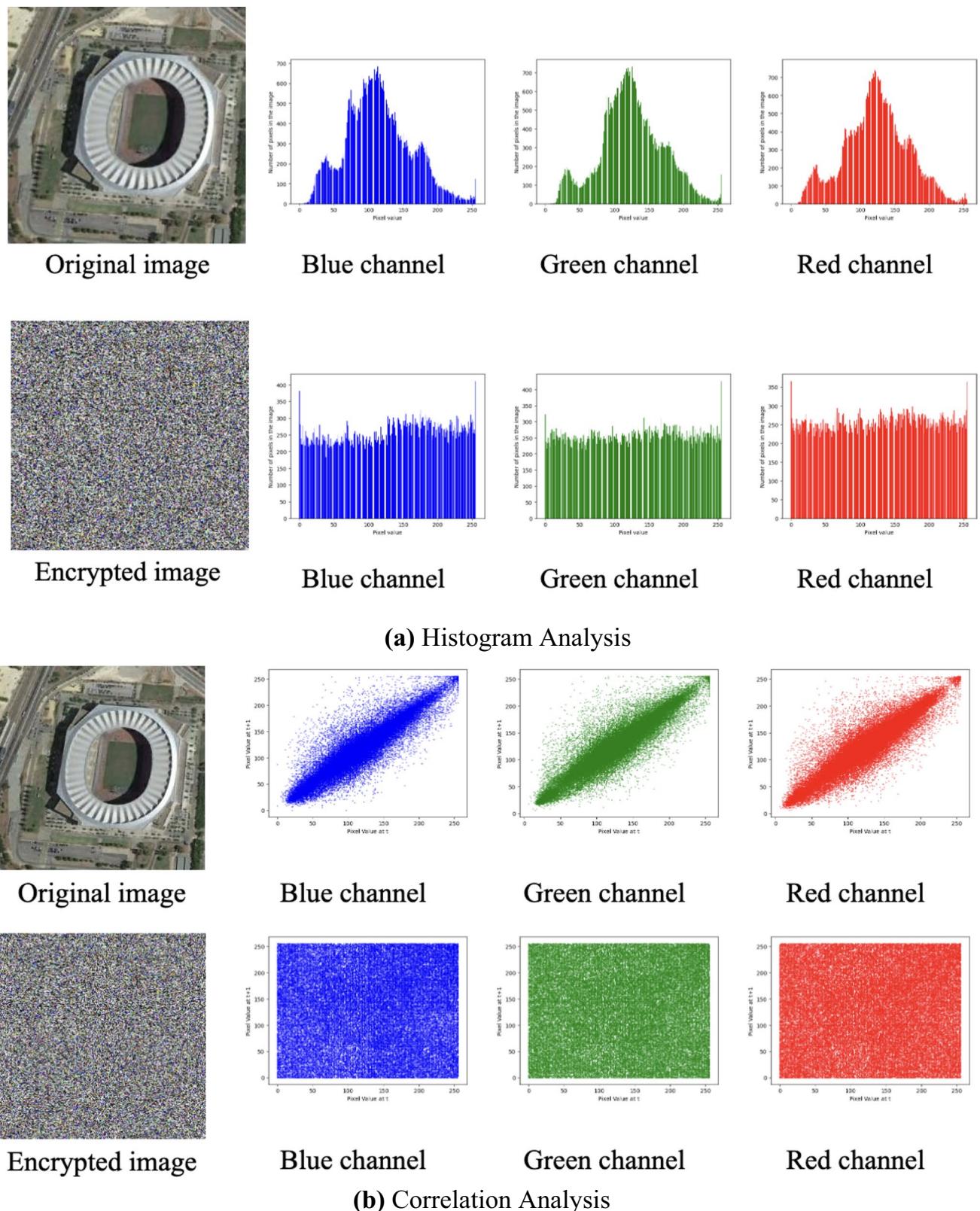


Fig. 12 (a) Histogram Analysis. **b** Correlation Analysis

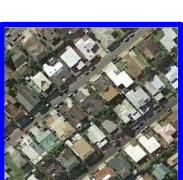
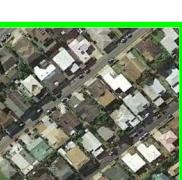
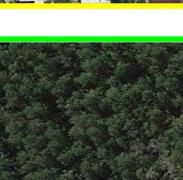
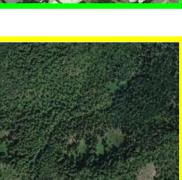
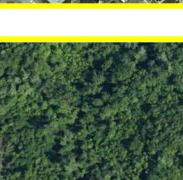
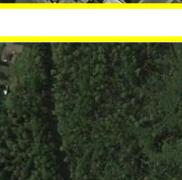
| Query Image | Results | | | | | |
|---|---|--|---|---|--|---|
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Fig. 13 Sample results of image retrieval

Performance under recent existing DL models

The retrieval performance of feature extraction-based CBIR schemes depends upon the features that best identify and describe the complex features of the image. CNN-based

deep learning techniques can be used to extract complex, more meaningful features. Here we present a comparative study on how retrieval performance varies under different deep-learning models for feature extraction. We have studied 9 different pre-trained models from Tensorflow. We have

Table 2 Retrieval Accuracy of NWPU (airplane, diamond, residential)

| NWPU | airplane | | | baseball_diamond | | | dense_residential | | |
|--------|----------|-------|-------|------------------|-------|-------|-------------------|-------|-------|
| | Prec | Rcal | F-S | Prec | Rcal | F-S | Prec | Rcal | F-S |
| VGG19 | 0.804 | 0.161 | 0.268 | 0.620 | 0.124 | 0.207 | 0.832 | 0.166 | 0.277 |
| VGG16 | 0.808 | 0.162 | 0.269 | 0.660 | 0.132 | 0.220 | 0.816 | 0.163 | 0.272 |
| MobN | 0.814 | 0.162 | 0.271 | 0.668 | 0.134 | 0.223 | 0.826 | 0.165 | 0.275 |
| MobV2 | 0.756 | 0.151 | 0.252 | 0.714 | 0.143 | 0.238 | 0.930 | 0.186 | 0.310 |
| IncV3 | 0.758 | 0.152 | 0.253 | 0.732 | 0.146 | 0.244 | 0.912 | 0.182 | 0.304 |
| Xcep | 0.788 | 0.158 | 0.263 | 0.698 | 0.139 | 0.233 | 0.746 | 0.149 | 0.249 |
| Resnet | 0.752 | 0.150 | 0.251 | 0.792 | 0.158 | 0.264 | 0.856 | 0.171 | 0.285 |
| D121 | 0.811 | 0.160 | 0.267 | 0.756 | 0.151 | 0.252 | 0.856 | 0.171 | 0.285 |
| D169 | 0.804 | 0.161 | 0.268 | 0.794 | 0.159 | 0.265 | 0.790 | 0.158 | 0.263 |

Table 3 Retrieval Accuracy of NWPU (forest, freeway, storage_tank)

| NWPU | forest | | | freeway | | | storage_tank | | |
|--------|--------------|--------------|--------------|--------------|---------------|--------------|--------------|--------------|--------------|
| | Prec | Rcal | F-S | Prec | Rcal | F-S | Prec | Rcal | F-S |
| VGG19 | 1 | 0.2 | 0.333 | 0.638 | 0.128 | 0.213 | 0.882 | 0.176 | 0.294 |
| VGG16 | 1 | 0.2 | 0.333 | 0.628 | 0.126 | 0.209 | 0.888 | 0.178 | 0.296 |
| MobN | 1 | 0.2 | 0.333 | 0.724 | 0.145 | 0.241 | 0.804 | 0.161 | 0.268 |
| MobV2 | 0.988 | 0.198 | 0.329 | 0.864 | 0.1073 | 0.288 | 0.802 | 0.160 | 0.267 |
| IncV3 | 0.990 | 0.198 | 0.330 | 0.718 | 0.143 | 0.239 | 0.790 | 0.158 | 0.263 |
| Xcep | 0.984 | 0.197 | 0.328 | 0.642 | 0.128 | 0.214 | 0.782 | 0.156 | 0.261 |
| Resnet | 1 | 0.2 | 0.333 | 0.702 | 0.140 | 0.234 | 0.784 | 0.157 | 0.261 |
| D121 | 1 | 0.2 | 0.333 | 0.770 | 0.154 | 0.257 | 0.862 | 0.172 | 0.287 |
| D169 | 1 | 0.2 | 0.333 | 0.750 | 0.150 | 0.250 | 0.822 | 0.164 | 0.274 |

Table 4 Retrieval Accuracy of PatternNet (airplane, diamond, residential)

| PNet | airplane | | | baseball_diamond | | | dense_residential | | |
|--------|----------|------------|--------------|------------------|--------------|--------------|-------------------|------------|--------------|
| | Prec | Rcal | F-S | Prec | Rcal | F-S | Prec | Rcal | F-S |
| VGG19 | 0.998 | 0.199 | 0.333 | 0.710 | 0.142 | 0.237 | 1 | 0.2 | 0.333 |
| VGG16 | 0.996 | 0.199 | 0.332 | 0.818 | 0.164 | 0.273 | 1 | 0.2 | 0.333 |
| MobN | 0.998 | 0.199 | 0.333 | 0.772 | 0.154 | 0.257 | 1 | 0.2 | 0.333 |
| MobV2 | 1 | 0.2 | 0.333 | 0.948 | 0.189 | 0.316 | 1 | 0.2 | 0.333 |
| IncV3 | 1 | 0.2 | 0.333 | 0.862 | 0.172 | 0.287 | 1 | 0.2 | 0.333 |
| Xcep | 1 | 0.2 | 0.333 | 0.782 | 0.156 | 0.261 | 1 | 0.2 | 0.333 |
| Resnet | 1 | 0.2 | 0.333 | 0.920 | 0.184 | 0.307 | 1 | 0.2 | 0.333 |
| D121 | 1 | 0.2 | 0.333 | 0.922 | 0.184 | 0.307 | 1 | 0.2 | 0.333 |
| D169 | 1 | 0.2 | 0.333 | 0.910 | 0.180 | 0.311 | 1 | 0.2 | 0.333 |

shown 6 sample classes of images to demonstrate the results from each dataset. Figure 13 shows the sample results from each class.

Tables 2, 3, 4, 5, 6 and 7 presents the retrieval accuracy of each class with respect to 9 different feature extraction mechanisms and $k = 10$. The six common classes are taken from 3 different datasets NWPU, PatternNet, and UC Merced Land Use.

We have run the experiments on the entire dataset and come up with the following Table 8. From the results, it is vivid that MobileNetV2 features perform well for remote sensing image retrieval.

mAP is another important evaluation metric of CBIR. Experiments are conducted in the same 6 classes and tabulated from Tables 9, 10 and 11 and compared the overall results in Table 12.

Table 5 Retrieval Accuracy of PatternNet (forest, freeway, storage_tank)

| PNet | forest | | | freeway | | | storage_tank | | |
|--------|----------|------------|--------------|----------|------------|--------------|--------------|--------------|--------------|
| | Prec | Rcal | F-S | Prec | Rcal | F-S | Prec | Rcal | F-S |
| VGG19 | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.866 | 0.173 | 0.289 |
| VGG16 | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.930 | 0.186 | 0.310 |
| MobN | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.856 | 0.171 | 0.285 |
| MobV2 | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.968 | 0.194 | 0.323 |
| IncV3 | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.840 | 0.168 | 0.280 |
| Xcep | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.838 | 0.168 | 0.279 |
| Resnet | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.932 | 0.186 | 0.310 |
| D121 | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.926 | 0.185 | 0.309 |
| D169 | 1 | 0.2 | 0.333 | 1 | 0.2 | 0.333 | 0.918 | 0.184 | 0.306 |

Table 6 Retrieval Accuracy of UC Merced Land Use (airplane, diamond, residential)

| UCM | airplane | | | baseball_diamond | | | dense_residential | | |
|--------|--------------|--------------|--------------|------------------|--------------|--------------|-------------------|--------------|--------------|
| | Prec | Rcal | F-S | Prec | Rcal | F-S | Prec | Rcal | F-S |
| VGG19 | 0.868 | 0.174 | 0.289 | 0.698 | 0.139 | 0.233 | 0.990 | 0.198 | 0.330 |
| VGG16 | 0.862 | 0.172 | 0.287 | 0.790 | 0.158 | 0.263 | 0.984 | 0.197 | 0.328 |
| MobN | 0.990 | 0.198 | 0.330 | 0.848 | 0.169 | 0.283 | 0.946 | 0.189 | 0.315 |
| MobV2 | 0.972 | 0.194 | 0.324 | 0.928 | 0.186 | 0.309 | 0.996 | 0.199 | 0.332 |
| IncV3 | 0.956 | 0.191 | 0.319 | 0.832 | 0.166 | 0.277 | 0.898 | 0.179 | 0.299 |
| Xcep | 0.986 | 0.197 | 0.329 | 0.716 | 0.143 | 0.239 | 0.978 | 0.196 | 0.326 |
| Resnet | 0.922 | 0.184 | 0.307 | 0.888 | 0.178 | 0.296 | 0.982 | 0.196 | 0.327 |
| D121 | 0.958 | 0.192 | 0.319 | 0.906 | 0.181 | 0.302 | 0.986 | 0.197 | 0.329 |
| D169 | 0.954 | 0.190 | 0.318 | 0.914 | 0.183 | 0.305 | 0.982 | 0.196 | 0.327 |

Table 7 Retrieval Accuracy of UC Merced Land Use (forest, freeway, storage_tank)

| UCM | forest | | | freeway | | | storage_tank | | |
|--------|----------|------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Prec | Rcal | F-S | Prec | Rcal | F-S | Prec | Rcal | F-S |
| VGG19 | 1 | 0.2 | 0.333 | 0.970 | 0.194 | 0.323 | 0.774 | 0.155 | 0.258 |
| VGG16 | 1 | 0.2 | 0.333 | 0.956 | 0.191 | 0.319 | 0.814 | 0.163 | 0.271 |
| MobN | 1 | 0.2 | 0.333 | 1 | 0.210 | 0.333 | 0.754 | 0.151 | 0.251 |
| MobV2 | 1 | 0.2 | 0.333 | 0.994 | 0.199 | 0.331 | 0.834 | 0.167 | 0.278 |
| IncV3 | 1 | 0.2 | 0.333 | 0.952 | 0.190 | 0.317 | 0.868 | 0.174 | 0.289 |
| Xcep | 1 | 0.2 | 0.333 | 0.980 | 0.196 | 0.327 | 0.852 | 0.170 | 0.284 |
| Resnet | 1 | 0.2 | 0.333 | 0.942 | 0.188 | 0.314 | 0.850 | 0.170 | 0.283 |
| D121 | 1 | 0.2 | 0.333 | 0.988 | 0.198 | 0.329 | 0.888 | 0.178 | 0.296 |
| D169 | 1 | 0.2 | 0.333 | 0.980 | 0.196 | 0.327 | 0.757 | 0.151 | 0.253 |

From Table 12 and Fig. 14, we can observe that the MobileNetV2 Deep learning model has achieved a maximum mAP value compared to the other models in all the datasets. The high mAP value can be attributed to the effectiveness of the retrieval scheme in ranking the images based on their relevance to the query. This indicates that the retrieval system using this model is more effective in retrieving relevant images. Additionally, with a focus on energy efficiency at edge devices, the MobileNetV2 model was selected to achieve a balance between energy efficiency and high retrieval accuracy.

Retrieval performance for different k values

Based on the analysis of the plot of K versus precision in Fig. 15 for values of K=5, 10, 15, and 20, it was found that the highest precision was achieved for K=5. However, this resulted in a relatively small result set. On the other hand, K=10 provided a reasonably high precision and a larger result set, making it the preferred choice. Additionally, the recall and f1 score were computed and plotted against the K values. The plots suggest that the recall and f1 score increase with an increase in K. These results indicate that

Table 8 Retrieval Accuracy Comparison

| | NWPU | | | PatternNet | | | UCM | | |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Prec | Rcal | F-S | Prec | Rcal | F-S | Prec | Rcal | F-S |
| VGG19 | 0.796 | 0.159 | 0.265 | 0.929 | 0.185 | 0.309 | 0.883 | 0.176 | 0.294 |
| VGG16 | 0.811 | 0.16 | 0.266 | 0.957 | 0.191 | 0.319 | 0.901 | 0.180 | 0.301 |
| MobN | 0.806 | 0.161 | 0.268 | 0.937 | 0.187 | 0.312 | 0.923 | 0.184 | 0.308 |
| MobV2 | 0.842 | 0.168 | 0.280 | 0.986 | 0.197 | 0.328 | 0.954 | 0.190 | 0.318 |
| IncV3 | 0.816 | 0.163 | 0.272 | 0.950 | 0.190 | 0.317 | 0.918 | 0.183 | 0.306 |
| Xcep | 0.773 | 0.155 | 0.258 | 0.936 | 0.187 | 0.312 | 0.919 | 0.184 | 0.306 |
| Resnet | 0.814 | 0.163 | 0.271 | 0.975 | 0.195 | 0.325 | 0.930 | 0.186 | 0.310 |
| D121 | 0.840 | 0.168 | 0.280 | 0.974 | 0.195 | 0.325 | 0.954 | 0.191 | 0.318 |
| D169 | 0.827 | 0.165 | 0.275 | 0.969 | 0.194 | 0.323 | 0.931 | 0.186 | 0.310 |

Table 9 MAP of 6 classes in NWPU

| NWPU | airplane | baseball_diamond | dense_residential | forest | freeway | storage_tank |
|--------|--------------|------------------|-------------------|--------------|--------------|--------------|
| VGG19 | 0.767 | 0.567 | 0.796 | 1 | 0.582 | 0.847 |
| VGG16 | 0.776 | 0.579 | 0.778 | 1 | 0.572 | 0.859 |
| MobN | 0.777 | 0.609 | 0.798 | 1 | 0.678 | 0.759 |
| MobV2 | 0.723 | 0.648 | 0.912 | 0.986 | 0.843 | 0.770 |
| IncV3 | 0.727 | 0.674 | 0.886 | 0.987 | 0.667 | 0.751 |
| Xcep | 0.745 | 0.634 | 0.708 | 0.977 | 0.569 | 0.746 |
| Resnet | 0.704 | 0.764 | 0.838 | 1 | 0.657 | 0.748 |
| D121 | 0.766 | 0.709 | 0.834 | 1 | 0.725 | 0.835 |
| D169 | 0.771 | 0.758 | 0.757 | 1 | 0.712 | 0.784 |

Table 10 MAP of 6 classes in PNet

| PNet | airplane | baseball_diamond | dense_residential | forest | freeway | storage_tank |
|--------|----------|------------------|-------------------|----------|----------|--------------|
| VGG19 | 0.996 | 0.674 | 1 | 1 | 1 | 0.857 |
| VGG16 | 0.995 | 0.789 | 1 | 1 | 1 | 0.922 |
| MobN | 0.998 | 0.750 | 1 | 1 | 1 | 0.851 |
| MobV2 | 1 | 0.935 | 1 | 1 | 1 | 0.963 |
| IncV3 | 1 | 0.841 | 1 | 1 | 1 | 0.811 |
| Xcep | 1 | 0.746 | 1 | 1 | 1 | 0.821 |
| Resnet | 1 | 0.910 | 1 | 1 | 1 | 0.921 |
| D121 | 1 | 0.911 | 1 | 1 | 1 | 0.912 |
| D169 | 1 | 0.886 | 1 | 1 | 1 | 0.903 |

the system's performance improves with an increase in the number of retrieved images, and selecting $K=10$ strikes a balance between precision and recall in the image retrieval process, making it a suitable choice for the system.

The impact of K on accuracy has been also experimented with. Accuracy in query results is linked to the number of relevant images in the database. Therefore, K doesn't have to remain a fixed constant; it can be customized based on the desired result set size. For instance, with 30 relevant images in the database, setting $K=10$ would yield an accuracy close to 1, retrieving nearly 10 out of 30 relevant images.

On the other hand, if $K=50$, the accuracy would be around 0.6, retrieving all 30 relevant and 20 irrelevant images. In summary, accuracy is largely influenced by the quantity of relevant images in the database. The parameter K can be adjusted to suit the required result set size or fine-tuned as a constant based on the dataset.

Precision vs recall

Figure 16 shows the Precision-Recall curve which highlights the trade-off between precision and recall for

Table 11 MAP of 6 classes in UCM

| | UCM | airplane | baseball_diamond | dense_residential | forest | freeway | storage_tank |
|--------|--------------|--------------|------------------|-------------------|--------------|--------------|--------------|
| VGG19 | 0.854 | 0.647 | 0.988 | 1 | 0.968 | 0.746 | |
| VGG16 | 0.844 | 0.764 | 0.982 | 1 | 0.954 | 0.774 | |
| MobN | 0.986 | 0.825 | 0.939 | 1 | 1 | 0.715 | |
| MobV2 | 0.962 | 0.913 | 0.995 | 1 | 0.994 | 0.802 | |
| IncV3 | 0.950 | 0.803 | 0.881 | 1 | 0.949 | 0.836 | |
| Xcep | 0.981 | 0.674 | 0.974 | 1 | 0.978 | 0.822 | |
| Resnet | 0.908 | 0.879 | 0.981 | 1 | 0.939 | 0.827 | |
| D121 | 0.951 | 0.895 | 0.984 | 1 | 0.988 | 0.870 | |
| D169 | 0.944 | 0.907 | 0.980 | 1 | 0.977 | 0.729 | |

Table 12 Comparative Analysis of mAP

| mAP | NWPU | PNet | UCM |
|-------------|--------------|--------------|--------------|
| VGG16 | 0.761 | 0.951 | 0.886 |
| VGG19 | 0.759 | 0.921 | 0.867 |
| MobileNet | 0.770 | 0.933 | 0.911 |
| MobileNetV2 | 0.813 | 0.983 | 0.944 |
| InceptionV3 | 0.782 | 0.941 | 0.903 |
| Xception | 0.732 | 0.927 | 0.904 |
| ResNet50 | 0.785 | 0.971 | 0.922 |
| DenseNet121 | 0.811 | 0.971 | 0.947 |
| DenseNet169 | 0.797 | 0.964 | 0.922 |

different retrieval scenarios. The results indicate that the precision of the system decreases and recall increases as the number of images retrieved increases. This is expected as retrieving a larger number of images results in a wider pool of candidates, which can lead to a lower precision,

while increasing the likelihood of retrieving relevant images, leading to higher recall. These results suggest that the system can effectively retrieve a larger number of images while maintaining a reasonable level of precision and recall.

Retrieval time analysis

Retrieval time is the turnaround time from the query image that has been given to receiving the encrypted images set. It is linearly dependent on the dataset size that we have in the repository. Figure 17 clearly shows the dependence.

Similarity measure metric analysis

Manhattan and Euclidean distance measures are commonly employed as similarity metrics in CBIR (Kapoor et al. 2021). The selection of a specific similarity measure

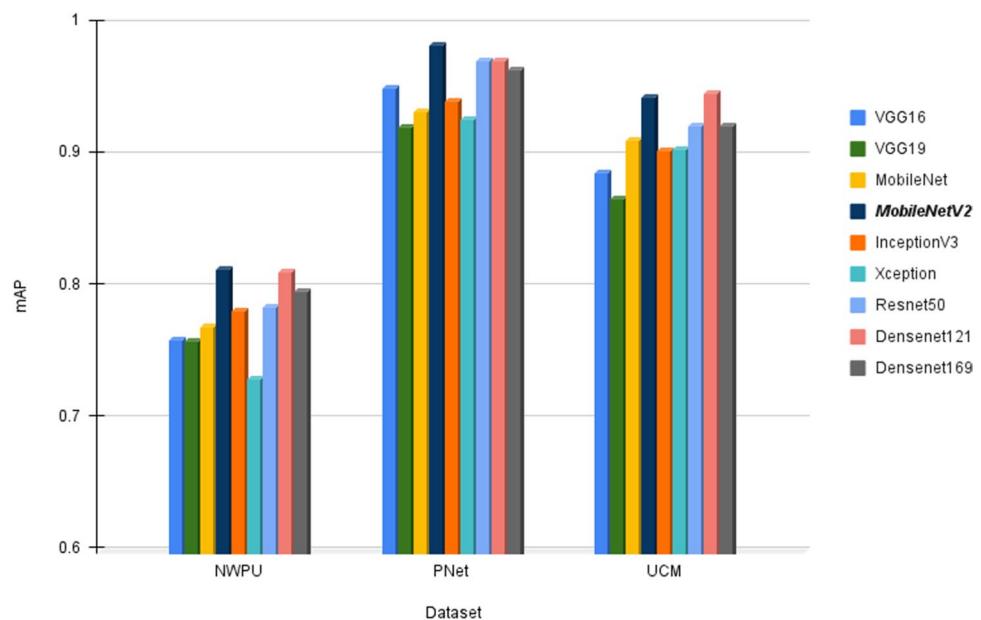
Fig. 14 Comparative Analysis of mAP

Fig. 15 K vs Precision, Recall, and F1 score (Fixed Mobilenet V2, NWPU)

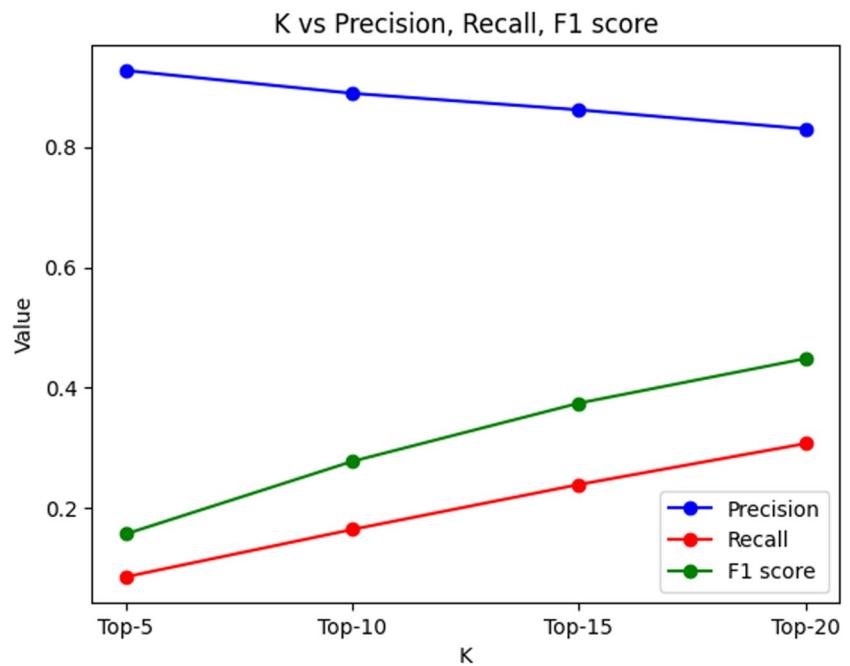
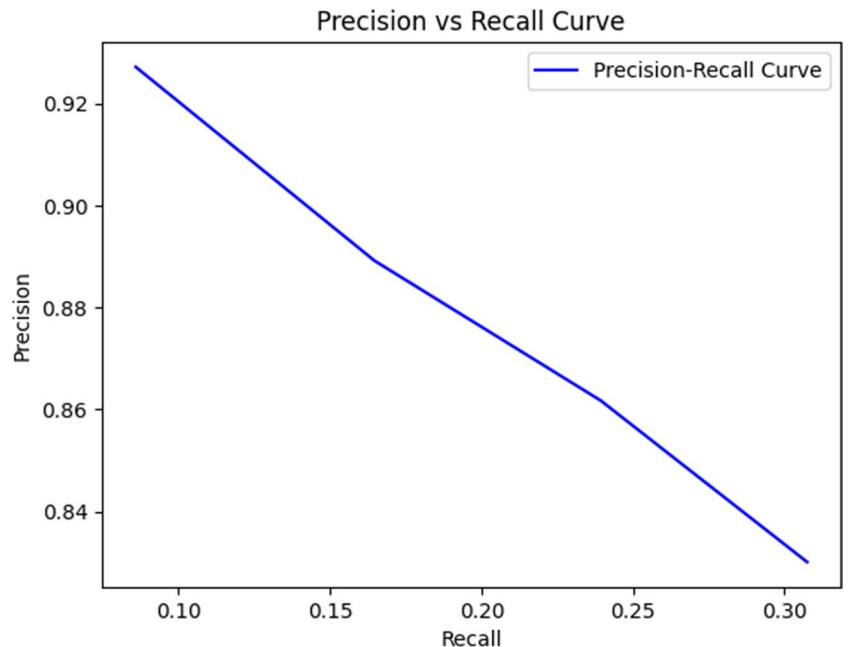


Fig. 16 Precision vs Recall (Fixed K, Mobilenet V2)

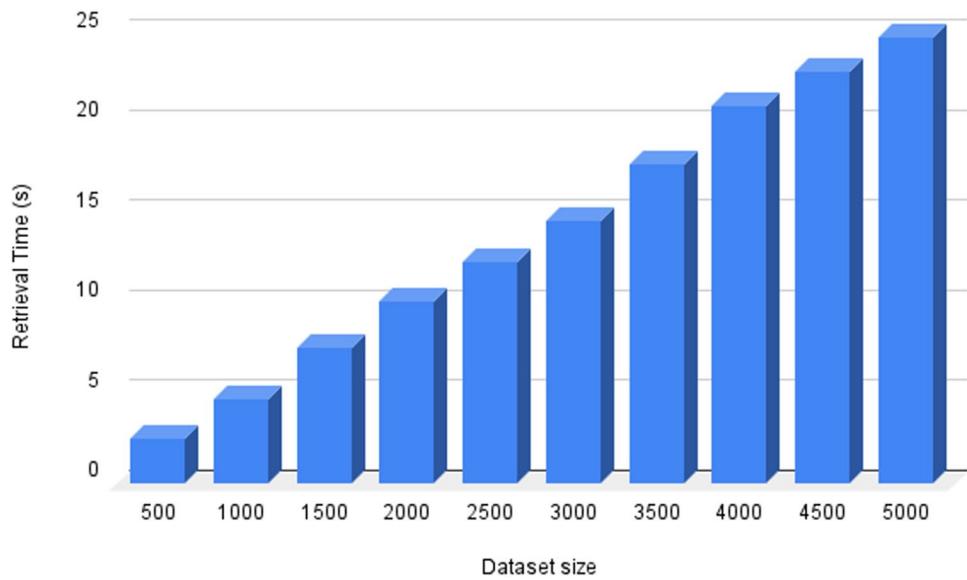


is determined through an experimental analysis, where the accuracy of the retrieved results is compared using both measures. The results of this comparative analysis are detailed in Table 13.

From Table 13, it is evident that the Manhattan distance measure provides better results compared to Euclidean distance. Also, Manhattan distance is comparatively more lightweight in terms of computation.

System availability analysis

The plot shown in Fig. 18 shows the time taken to retrieve the image when different numbers of servers are active, ranging from all five servers to only one server. For cases where at least three servers are active, the retrieval time is constant, and hence normalized to 1. However, if less than three servers are active, the retrieval time is set to zero,

Fig. 17 Retrieval time Analysis**Table 13** Similarity Measure Analysis

| mAP | NWPU | | PNet | | UCM | |
|-------------------|-------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| | Query Class | Manhattan Distance | Euclidean distance | Manhattan Distance | Euclidean distance | Manhattan Distance |
| airplane | 0.724 | 0.711 | 0.998 | 0.932 | 0.962 | 0.897 |
| baseball_diamond | 0.652 | 0.630 | 0.935 | 0.878 | 0.913 | 0.893 |
| dense_residential | 0.912 | 0.892 | 0.989 | 0.932 | 0.995 | 0.991 |
| forest | 0.985 | 0.943 | 0.990 | 0.901 | 0.996 | 0.891 |
| freeway | 0.843 | 0.810 | 0.988 | 0.899 | 0.994 | 0.943 |
| storage_tank | 0.771 | 0.741 | 0.963 | 0.912 | 0.811 | 0.765 |

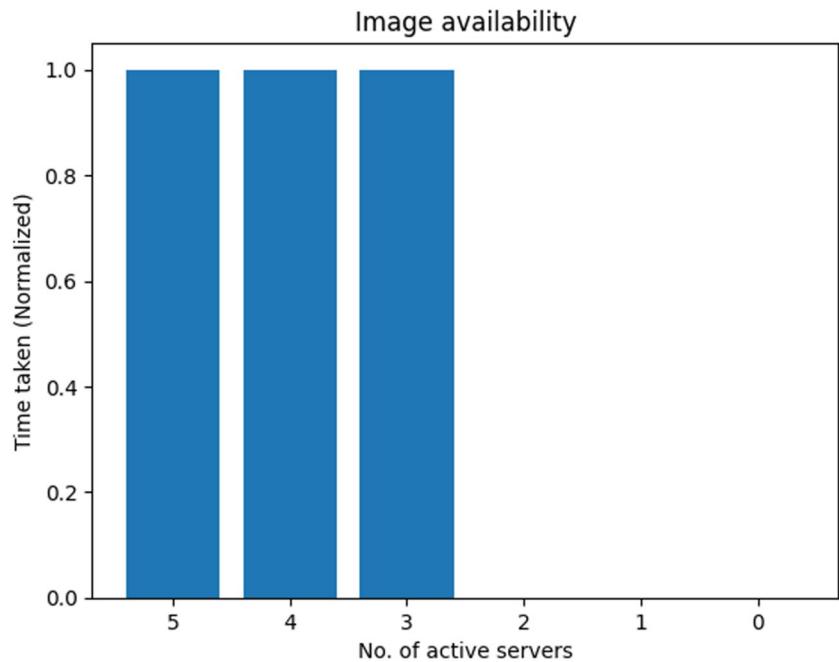
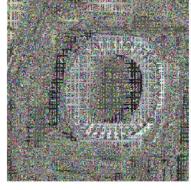
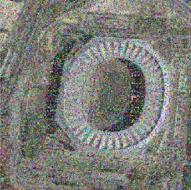
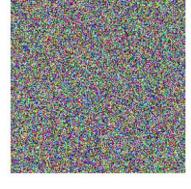
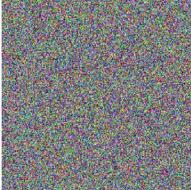
Fig. 18 System availability analysis

Table 14 Various Image Attacks

| Original Image | Attack Type | Reconstructed Image |
|---|---------------------------|--|
| | 15% Salt and Pepper Noise |  NC = 0.39 |
|  | 75% Crop |  NC = 0.33 |
| | 10% Compression |  NC = 0.47 |
| | 1% Gaussian Noise |  NC = 0.001 |
| | 5% Blur |  NC = 0.0003 |

as the system cannot retrieve the image without at least three active servers. The results of the analysis show that the retrieval time is not affected as long as a minimum of three active servers are available. This demonstrates the robustness of the system to handle server failures, as long as the minimum number of active servers is maintained, and ensures the availability of images and reliability of the image retrieval process.

Security threat analysis

In our distributed cloud setup, only the master cloud is exposed to the internet. All the slave clouds and the databases are connected internally and not accessible from the internet. Hence, an intruder can only perform insider attacks on the system. When the intruder gains access to the stored remainder matrix and shadows, even when he reconstructs

the stored image, without the encryption keys he cannot decrypt the reconstructed encrypted image. Therefore the privacy of the data is not compromised.

Attacks focussing on data corruption can be performed at both master meta-db and slave databases. Storing data in a secure database service with regular backup (with integrity verification) can eliminate the corruption of already stored data. But still, the attacker can perform various image-based attacks (shown in Table 14) on the retrieved shadows to corrupt the data and make the data meaningless when reconstructed and decrypted. The encryption algorithm (Shafique et al. 2021) is resilient to image attacks such as cropping and noise attacks, up to a certain limit. However, beyond this limit, decryption of the encrypted image results in a highly distorted image. The Table 14 illustrates the various attacks along with their limits, beyond which the encryption algorithm is no longer resistant. To ensure that the decrypted image has negligible distortion, it is essential to detect and prevent image data corruption. The results of the analysis emphasize the need for a robust data corruption detection mechanism to maintain the integrity of the encrypted images, and to ensure that the decrypted images are of high quality.

Attack detection and recovery model analysis

This subsection is dedicated to discussing how the proposed system reduces the impact of security attacks. Figure 19 shows the distributed cloud setup, where one slave cloud is malicious. A malicious slave cloud can corrupt the data under the following scenarios:

- During storage, it can corrupt the shadow before storing it. In this case, data corruption is irreversible and is a very serious threat

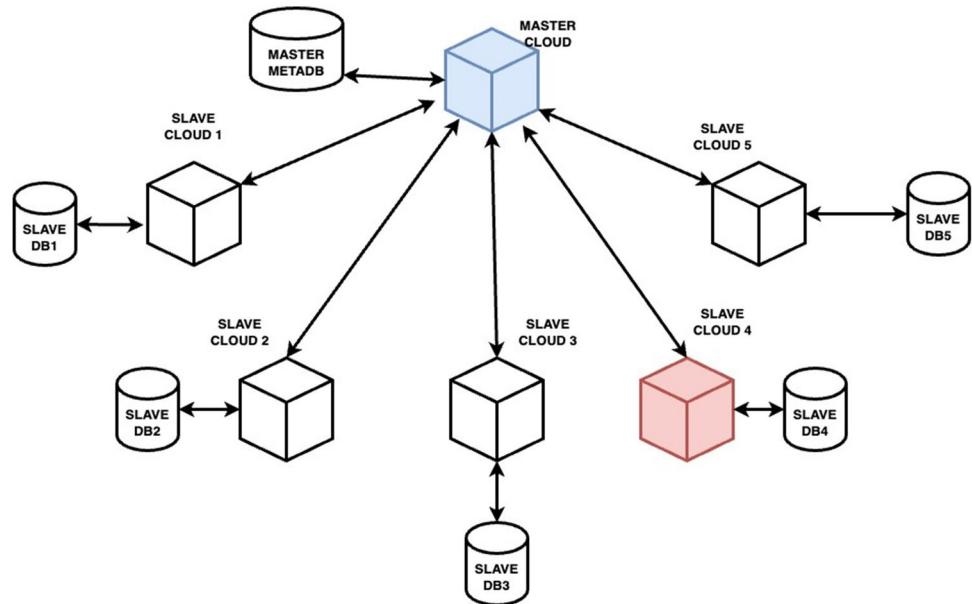
Figure 20 depicts the attack scenario during storage. The master sends each shadow to TPA and the corresponding slave for storage. Each slave stores the received shadow and sends a confirmation to the master and TPA. TPA verifies the integrity of the shadow immediately after receiving the confirmation from the corresponding slave.

- During retrieval, it can corrupt the retrieved copy of the shadow before sending it to the master. In this case, the original data stored is not modified, only the retrieved data is modified.

Figure 21 shows the attack scenario during retrieval. The master fetches shadows from any 3 available slave clouds and sends the list of IDs of those slave clouds to TPA. TPA fetches the shadow from those 3 slave clouds and verifies their integrity by comparing the fetched shadow's hash with the original shadow's hash stored in the TPA's hash db.

Figure 22 shows the plot for NC value between various original images and their corresponding decrypted and retrieved copy that was generated for two cases of the system under attack (50% crop attack): one where the TPA is present and the other where the TPA is not present. In the first case, all images had an NC value close to 1, implying that the retrieved image is nearly identical to the original image. Hence it is evident that the attack was unsuccessful and our system is resistant to these image-based attacks. In the second case, the NC value fluctuated significantly around 0.4, implying large distortions in the

Fig. 19 Distributed Cloud Setup



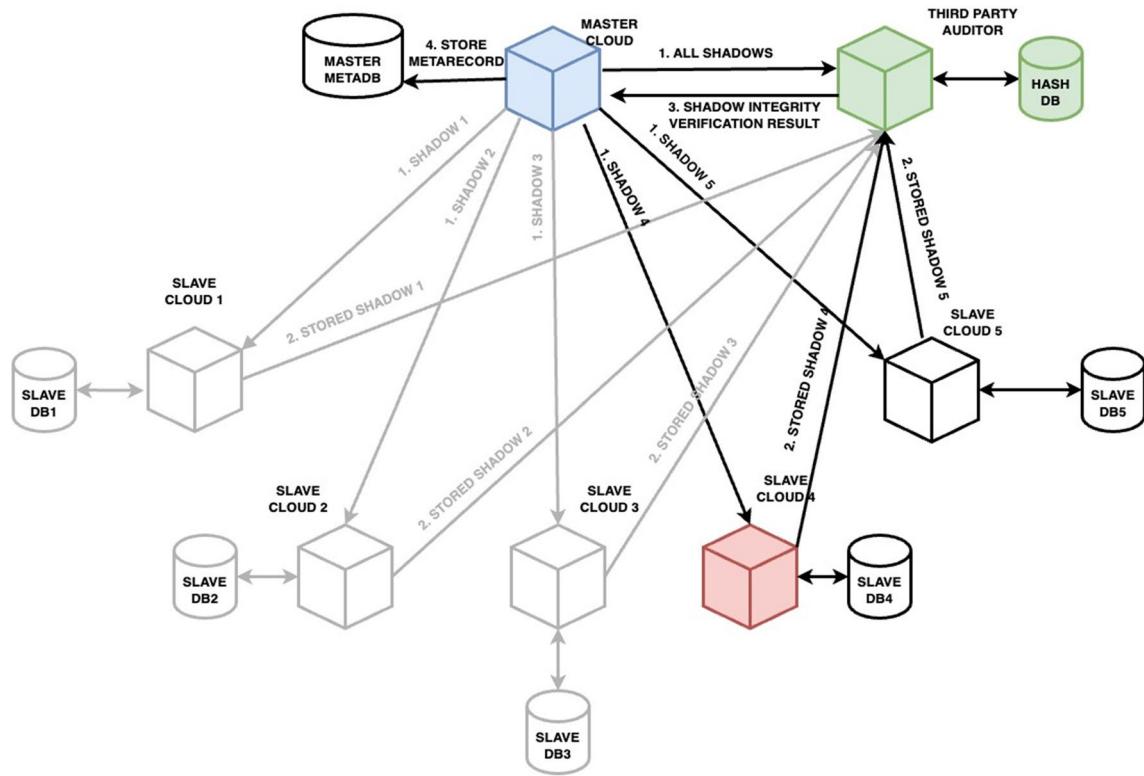


Fig. 20 Attack during storage

retrieved image and variations from the original image. Here the attack was successful in the absence of TPA, which is the case in the existing retrieval systems. To prevent such image attacks which result in a lower NC value, it is important to detect them beforehand, thereby ensuring that the images are retrieved with negligible distortion. This analysis highlights the importance of the detection and prevention of image data corruption and how it affects the accuracy and integrity of the retrieved images in the system. This analysis also supports the fact that our system is resistant to these data integrity-compromising, image-based attacks.

Theoretical analysis of the security provided by Hybrid Image Encryption (HE), Secret Image Sharing (SIS), and Third-Party Auditing (TPA) on the cloud side in EdgeShield involves establishing mathematical foundations for each component. Here's a conceptual breakdown:

Let I be the original image, and E denote the encryption function. Chaotic logistic map and inverse inverse discrete wavelet transform (IDWT) together form the hybrid encryption process:

$$HE(I) = E(\text{Rand_Img(Chaotic_Sequence})$$

$$\oplus \text{ IDWT}(I) \forall_{\text{channels}}$$

n shadows S_1, S_2, \dots, S_n and a remainder matrix R are generated from the encrypted image. The sharing process can be expressed as

$$SIS(HE(I)) = \{S_1, S_2, \dots, S_n, R\}$$

TPA involves the verification V of the integrity of the stored shadows and the reconstruction R' of the original image during retrieval with k shadows. The verification process can be expressed as

$$V(\{S_1, S_2, \dots, S_k, R\}) = \{0, 1\} \left\{ \begin{array}{l} \text{where } 0 \text{ indicates integrity} \\ \text{verification failure, and } 1 \text{ indicates success.} \end{array} \right.$$

The reconstruction function can be denoted as

$$R'(\{S_1, S_2, \dots, S_k, R\}) = HE^{-1}(\{S_1, S_2, \dots, S_k, R\})$$

where HE^{-1} is the hybrid image decryption function.

The collective security of EdgeShield using HE, SIS, and TPA can be represented as

$$\begin{aligned} TPA(SIS(HE(I))) &= V(\{S_1, S_2, \dots, S_k, R\}) \\ &\ast R'(\{S_1, S_2, \dots, S_k, R\}) = I \end{aligned}$$

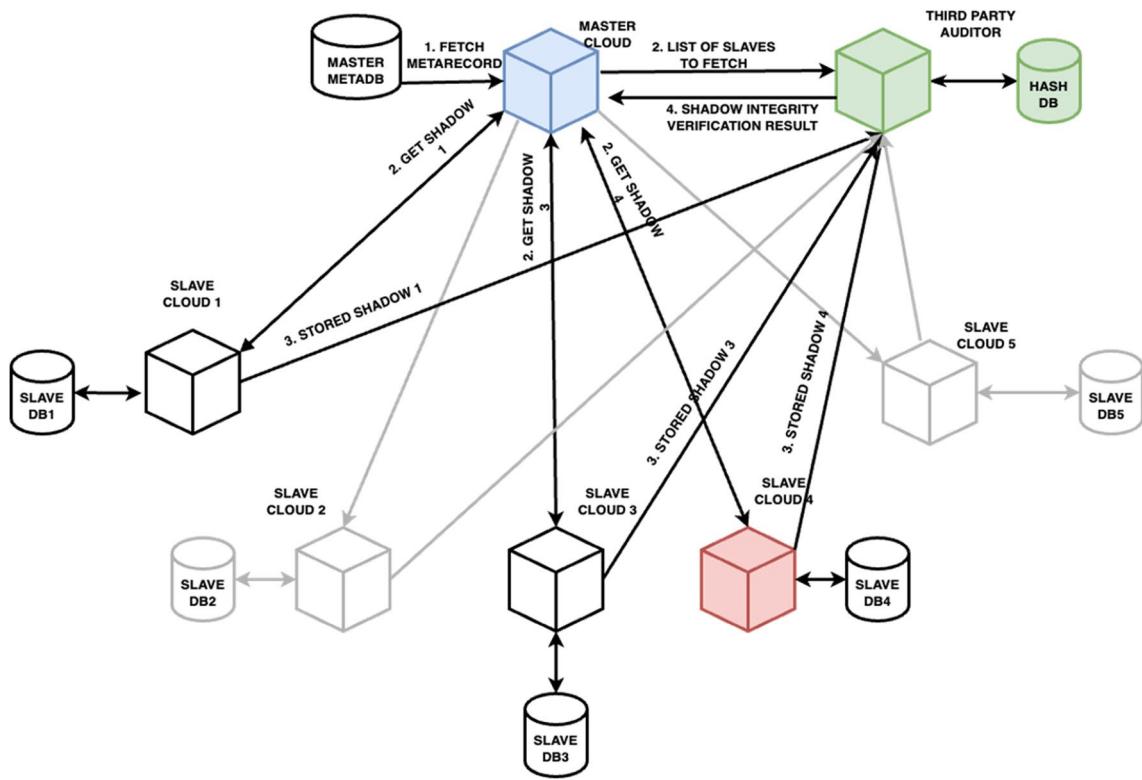
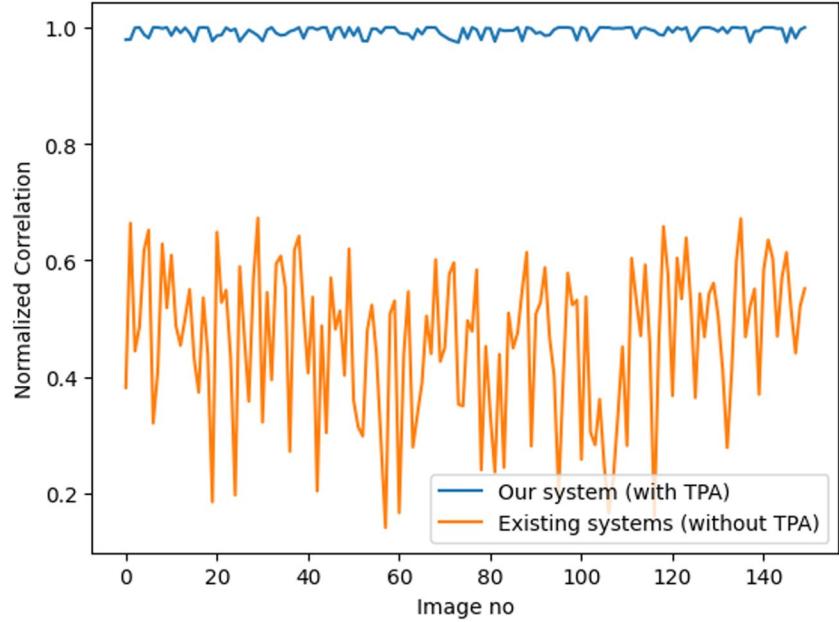


Fig. 21 Attack during retrieval

Fig. 22 Attack Impact Analysis



This equation symbolizes the theoretical framework wherein the combined contributions of third-party auditing, secret image sharing, and hybrid image encryption converge to ensure security within EdgeShield.

Conclusion and future works

This paper presents an advanced image retrieval system that emphasizes the integration of the PPCBIR scheme,

edge computing, distributed cloud environment, and third-party auditing, with a focus on ensuring secure storage and retrieval of images while preserving privacy. The retrieval accuracy is improved by using a CBIR scheme involving MobileNetV2-based feature extraction techniques. To reduce processing overload on users, an edge computing layer is introduced that performs encryption and feature extraction tasks. Finally, to ensure the availability of images, a distributed cloud environment is implemented, enabling users to access images even during a server blackout. In military kinds of sensitive domains, to detect and mitigate image data corruption, third-party auditing was implemented. The system has been designed with the principles of data security as a primary goal. Confidentiality, integrity, and availability have all been achieved through the use of encryption, third-party auditing, and distributed cloud storage respectively. The system performed well in various performance evaluations such as mean average precision, normalized correlation, and impact under attack. The results demonstrate that the system is effective in ensuring privacy, security, and accuracy in image retrieval while maintaining the availability of images for authorized users. In the future system's retrieval performance can be improved by extracting more meaningful, complex features. To improve security, various attacks at the edge layer can be comprehensively studied and mitigated.

Author contributions Conceptualization: Vetriselvi V, Ajitesh M, Deekshith M, Arun Amaithi Rajan, Hemanth D;

Methodology and Development: Ajitesh M, Deekshith M, Hemanth D;

Formal analysis and investigation: Ajitesh M, Deekshith M, Hemanth D;

Writing - original draft preparation: Ajitesh M, Deekshith M, Arun Amaithi Rajan, Hemanth D;

Writing - review and editing: Deekshith M, Ajitesh M, Arun Amaithi Rajan, Vetriselvi V;

Supervision: Vetriselvi V.

Funding No funding was received for this work.

Data availability All publicly available datasets have been utilized for this work.

Declarations

Competing interest The authors declare no competing interests.

References

- Bai L (2006) A Reliable (k, n) Image secret sharing scheme. 2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, Indianapolis, IN, USA. pp 31–36. 10.1109/DASC.2006
- Bhavyasree V, Yalla P (2021) Public auditing to provide privacy preservation of cloud data using ring signatures. 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 1154–1160. <https://doi.org/10.1109/I-SMAC52330.2021.9640805>
- Chakraborty S, Singh S, Thokchom S (2018) Integrity checking using third party auditor in cloud storage. 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, pp 1–6. <https://doi.org/10.1109/IC3.2018.8530649>
- Garg N, Nehra A, Baza M, Kumar N (2023) Secure and efficient data integrity verification scheme for cloud data storage. 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 1–6, <https://doi.org/10.1109/CCNC51644.2023.10059690>
- Hou D, Miao Z, Xing H et al (2020) Exploiting low dimensional features from the MobileNets for remote sensing image retrieval. Earth Sci Inform 13:1437–1443. <https://doi.org/10.1007/s12145-020-00484-3>
- Kapoor R, Sharma D, Gulati T (2021) State of the art content based image retrieval techniques using deep learning: a survey. Multimed Tools Appl 80:29561–29583. <https://doi.org/10.1007/s11042-021-11045-1>
- Kumar S, Kumar D, Lamkuche HS (2021) TPA auditing to enhance the privacy and security in cloud systems. JCSANDM 10(3):537–568
- Kumar A, Jones R, Joshi P (2017) Survey of cryptographic hashing algorithms for message signing. In: IJCST, 8(2)
- Li J-S, Liu I-H, Tsai C-J, Su Z-Y, Li C-F, Liu C-G (2020) Secure content-based image retrieval in the cloud with key confidentiality. IEEE Access 8:114940–114952. <https://doi.org/10.1109/ACCESS.2020.3003928>
- Mingchang W, Zhang X, Niu X, Wang F, Zhang X (2019) Scene classification of high-resolution remotely sensed image based on ResNet. J Geovis Spat Anal. 3:16. <https://doi.org/10.1007/s41651-019-0039-9>
- Pérez J, Díaz J, Berrocal J et al (2022) Edge computing. Computing 104:2711–2747. <https://doi.org/10.1007/s00607-022-01104-2>
- Qin Z, Weng J, Cui Y, Ren K (2018) Privacy-Preserving image processing in the cloud. IEEE Cloud Comput 5(2):48–57. <https://doi.org/10.1109/MCC.2018.022171667>
- Qin Z, Yan J, Ren K, Chen CW, Wang C (2014) Towards efficient privacy-preserving image feature extraction in cloud computing. In: Proceedings of the 22nd ACM International Conference on Multimedia pp 497–506
- Rajath AN, Vidyalakshmi K, Keshava Murthy GN (2023) A comprehensive analysis on deep learning based image retrieval. 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), Dharwad, India, pp 1–4. <https://doi.org/10.1109/ICAISC58445.2023.10200622>
- Sengar SS, Kumar S (2022) Content-based secure image retrieval in an untrusted third party environment. EasyChair Preprint 9037
- Shafique A, Hazzazi MM, Alharbi AR, Hussain I (2021) Integration of spatial and frequency domain encryption for digital images. IEEE Access 9:149943–149954. <https://doi.org/10.1109/ACCESS.2021.3125961>
- Sunitha T, Sivarani TS (2021) An efficient content-based satellite image retrieval system for big data utilizing threshold based checking method. Earth Sci Inform 14:1847–1859. <https://doi.org/10.1007/s12145-021-00629-y>
- Tanwar VK, Rajput AS, Raman B, Bhargava R (2018) Privacy preserving image scaling using 2D bicubic interpolation over the cloud. 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, pp 2073–2078. <https://doi.org/10.1109/SMC.2018.00357>

- Tanwar VK, Raman B, Rajput AS, Bhargava R (2022) SecureDL: A privacy preserving deep learning model for image recognition over cloud. *J Vis Commun Image Represent* 86:103503
- Wen L, Cheng Y, Fang Y, Li X (2023) A comprehensive survey of oriented object detection in remote sensing images. *Expert Syst Appl* 224:119960. <https://doi.org/10.1016/j.eswa.2023.119960>
- Wentao W, Zhou T, Qin J, Xiang X, Tan Y, Cai Z (2022) A privacy-preserving content-based image retrieval method based on deep learning in cloud computing. *Expert Syst Appl* 203:117508. <https://doi.org/10.1016/j.eswa.2022.117508>
- Xia Z, Wang L, Tang J, Xiong NN, Weng J (2021) A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing. In: IEEE Transactions on Network Science and Engineering, 8, (1):318–330. <https://doi.org/10.1109/TNSE.2020.3038218>
- Zhang J, Zhou Q, Shen X et al (2019) Cloud detection in high-resolution remote sensing images using multi-features of ground objects. *J Geovis Spat Anal* 3:14. <https://doi.org/10.1007/s41651-019-0037-y>
- Zhang C, Zhu L, Zhang S, Yu W (2020) TDHPIR: An efficient deep hashing based privacy-preserving image retrieval method. *Neurocomputing* 406:386–398. <https://doi.org/10.1016/j.neucom.2019.11.119>
- Zhou F, Qin S, Hou R, Zhang Z (2022) Privacy-preserving image retrieval in a distributed environment. *Int J Intell Syst* 37:7478–7501. <https://doi.org/10.1002/int.22890>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.