Progressive privacy-preserving batch retrieval of lung CT image sequences based on edge-cloud collaborative computation

Yi Zhuang☯, Nan Jiang☯

1 College of Computer & Information Engineering, Zhejiang Gongshang University, Hangzhou, P.R. China, 2 Affiliated Hangzhou First People’s Hospital, Zhejiang University School of Medicine, Hangzhou, P.R. China

☯ These authors contributed equally to this work.
* zhuang@zjgsu.edu.cn

Abstract

Background

A computer tomography image (CI) sequence can be regarded as a time-series data that is composed of a great deal of nearby and similar CIs. Since the computational and I/O costs of similarity measure, encryption, and decryption calculation during a similarity retrieval of the large CI sequences (CISs) are extremely high, deploying all retrieval tasks in the cloud, however, will lead to excessive computing load on the cloud, which will greatly and negatively affect the retrieval performance.

Methodologies

To tackle the above challenges, the paper proposes a progressive privacy-preserving Batch Retrieval scheme for the lung CISs based on edge-cloud collaborative computation called the BRS method. There are four supporting techniques to enable the BRS method, such as: 1) batch similarity measure for CISs, 2) CIB-based privacy preserving scheme, 3) uniform edge-cloud index framework, and 4) edge buffering.

Results

The experimental results reveal that our method outperforms the state-of-the-art approaches in terms of efficiency and scalability, drastically reducing response time by lowering network communication costs while enhancing retrieval safety and accuracy.

Introduction

With the rapid growth of the number of medical images and the increasing demand for remote diagnosis, content-based mobile retrieval for Computed tomography Image Sequences (CIS) in telemedicine systems (TSs) [1] plays an increasingly important role in disease diagnosis in recent years. Fig 1 illustrates an example of a lung CIS consisting of a large number of nearby and visually similar Computed tomography Image(CI)s. As one of the main tasks of the TS,
mobile-terminal-based high-resolution CIS retrieval enables medical professionals to identify lesion tissues in aberrant CIs and carry out the computer-assisted diagnosis and treatment.

The motivations of the CIS retrieval in edge-cloud collaborative computing mode are based on the following key observations:

- Instead of using the whole CIS, traditional CI retrieval takes a single CI as the retrieval one to perform similarity comparison which is ineffectual and inadequate in modeling the whole retrieval CIS leading to poor retrieval precision ratio;
- As the CIS data belongs to patients’ personal privacy information [2], it should be encrypted during the retrieval processing; otherwise, the personal information leakage will take place;
- To better understand the condition of their patients during remote consultations, doctors will frequently retrieve and examine their CISs in real time, which involves high computational costs, as well as the intensive transmission of the CISs. Therefore, deploying and executing all expensive retrieval and computing tasks in the cloud will result in significant computational overhead and have a negative impact on the retrieval’s performance improvement. To efficiently reduce the load of cloud computing, edge computing [3] came into being. As a new distributed computing mode, edge computing makes up for some shortcomings of cloud computing and diverts most computing tasks to edge device nodes (i.e., edge server (ES)) around the mobile terminal. This can not only significantly reduce the computing load on the cloud, but also reduce the transmission cost [4, 5] to support the retrieval in real time [6];
- For these mobile terminals whose computing resources are constraint such as the battery reserves, screen resolutions and computational powers, etc. The data transmission is negatively affected by the unstable network bandwidth which causes delays in the data retrieval and transmission, especially in rural areas with inadequate mobile communication infrastructure [1].

Based on the above analysis, the paper presents a privacy-preserving Batch Retrieval method for large lung CISs in the edge-cloud computing network, called the BRS, by analyzing the similarity of the nearby CIs in the sequence. There are few studies on how to speedup the batch similarity retrieval of the large CISs using the edge-cloud collaborative computing environment. Specifically, when a user submits a retrieval CIS(XR), firstly, an index mechanism at the edge layer called eIndex is used to quickly judge whether there are some answer CISs similar to XR in the edge buffer. If exists, then a high-dimensional similarity retrieval of the partial answer ones supported by the cIndex is carried out by accessing the cloud; otherwise, the similarity retrieval of all CISs supported by the cIndex is performed directly through the cloud; finally the answer CISs are returned to the user node. The extensive experiments demonstrate the effectiveness, efficiency, and scalability of the BRS method.
Background

Over the past fifty years, content-based image retrieval (CBIR) has been a persistent and difficult research problem [6–10]. Due to the 'semantic gap', however, the retrieval accuracies are still not satisfactory.

As one of the key subfields of the CBIR, content-based medical image retrieval (CBMIR) research has become increasingly popular in recent years. The first CBMIR system built for high-resolution lung CTs is ASSERT [11]. After that, many prototype systems were developed, including IRMA [12], FIRE [13], and others. A noisy image bag-based technique for retrieving medical images was presented by Huang et al. [14]. To further reduce the 'semantic gap', Huang et al. [15] developed a relevance feedback technique for the CBMIR based on a noisy-smoothing model. Kitanovski et al. [16] designed a multi-modality-based CBMIR system. Lan et al. [17] proposed a simple texture feature extraction algorithm for the CBMIR. A multi-panel medical image segmentation framework for the CBMIR system was supplied by Ali et al. [18]. Based on the fusion of the wavelet optimization and adaptive block truncation coding, Kasban et al. [19] built a reliable CBMIR system. Tuyet et al. [20] used the deep learning techniques to support the salient region-based CBMIR.

Since the aforementioned CBMIR systems are based on single-PC mode, their retrieval performances are not satisfactory when dealing with a great deal of medical images [21]. Anbarasi et al. [22] developed a distributed CBMIR system using distributed database techniques. Charisi et al. [23] designed a parallel CBMIR scheme in a peer-to-peer (P2P) network. Based on the hybrid features, Depeursinge et al. [24] proposed a mobile access approach to peer-reviewed medical information. Although Zhuang et al. [25] put forward an efficient and robust CBMIR technique in a mobile wireless network, the retrieval efficiency is poor since the effectiveness of the load balance strategy needs to be further improved. Based on the previous work [25], Zhuang et al. [26] introduced a high-performance batch retrieval technique for medical images in wireless network from a standpoint of multi-retrieval optimization to further improve the retrieval efficiency. A mobile teleradiology system [27] is appropriate for streamlining the CBMIR procedure. For telemedicine applications, Chitra et al. [28] suggested an enhanced retrieval approach for brain images utilizing carrier frequency offset adjusted OFDM technique. To solve the 'semantic gap', Jiang et al. [29] introduced a novel framework of mobile similarity retrieval of medical images based on a crowdsourcing model.

On the basis of the CI analysis, Lei et al. [30] developed a sparse CNN model-based high-resolution CI retrieval technique. Yu et al. [31] presented a liver CI retrieval algorithm based on a non-tensor product wavelet. Based on an adder combining two local bit plane-based dissimilarities, Hatibaruah et al. [32] introduced a novel CI retrieval approach. Hwang et al. [33] applied a CBIR and CNN techniques to enable diffuse interstitial lung disease retrieval. To facilitate the effective diagnosis of the lung cancer, Alzubi et al. [34] designed a boosted neural network ensemble classification approach.

Despite extensive study of the CI retrieval, the majority of approaches still rely solely on this retrieval without taking the CIS retrieval into account. Meanwhile, very little research has addressed CIS retrieval in the collaborative edge-cloud environment.

Preliminaries and preprocessing

Firstly, the main symbol notations are listed in Table 1.

Fig 2 depicts the three-layer network architecture of the BRS system, which is formally stated in Definition 1.
Table 1. Main notations used throughout the paper.

| Notation | Meaning |
|----------|---------|
| $N_U$    | user node |
| $N_E$    | edge node |
| $N_C$    | cloud node |
| $\Omega$ | a set of $n$ CISs in $N_C$ |
| $\Omega'$| the CISs buffered at $N_E$ |
| $X_i$    | the $i$-th CIS and $X_i \in \Omega$ |
| $X_R$    | the retrieval CIS |
| $r_R$    | the retrieval radius |
| $\Psi$   | the final answer CISs |
| $\Psi'$  | partial answer CISs based on the $e$Index at $N_E$ |
| $\Psi''$ | partial answer CISs based on the $c$Index at $N_C$ |
| $X_{i2}$ | the $i$-th historical retrieval CIS |
| $r_{i2}$ | the $i$-th historical retrieval radius |
| $CI_{i}$ | the $j$-th CI in $X_i$ and $CI \in X_i$ |
| POA_{i}  | the $j$-th pathological object area in a CI and $j \in [1, |POA|]$ |
| NPOA     | the non-POA part of the CI |
| CIB_{i,j} | the $j$-th correlated image block of the POAs in $CI_{i}$ |
| NIB_{i,j}| the $j$-th image block of the NPOA part in $CI_{i}$ |
| sim($X_i, X_j$) | similarity between two CISs (i.e., $X_i$ and $X_j$) (ref. Eq (4)) |
| $d(CI_i, CI_j)$ | Euclidean distance between two CISs (i.e., $CI_i$ and $CI_j$) |
| MaxN     | maximal number of the CISs buffered at $N_E$ |
| $\delta$ | granularity value for the size of image blocking |

https://doi.org/10.1371/journal.pone.0274507.t001

Fig 2. The three layer architecture of the BRS system.

https://doi.org/10.1371/journal.pone.0274507.g002
Definition 1 (MECN). A mobile edge-cloud network (MECN) is represented by a graph \( G \) which can be modeled by a triplet:

\[
G = \langle N, E, T \rangle
\]

where

- \( N \) means a set of nodes, formally represented as \( N = N_U \cup N_E \cup N_C \), in which
  - \( N_U \) represents a user node that is used for: 1) submitting the retrieval; 2) decryption of the RIBs; 3) acquisition, reconstruction and display of the CISs;
  - \( N_E \) represents the edge nodes that are used for: 1) temporally storing the CISs buffered at \( N_E \); and 2) sending back the partial answer CISs to \( N_U \);
  - \( N_C \) represents the cloud nodes which are used for: 1) partition processing of the IBs; 2) encryption and storage of the CIBs; storing the NIB replicas, and 3) sending back the answer CISs to \( N_U \);
- \( E \) denotes a collection of edges representing the different network bandwidths for data communication at time \( T \), formally denoted as: \( E = \langle e_1, e_2, \ldots, e_{|E|} \rangle \), where \( e_k = (N_i, N_j) \) refers to the \( k \)-th edge in \( G \) in which \( N_i \) and \( N_j \) are connected.

Definition 2 (POA). A pathological object area (POA) in a CI can be modeled by a two-tuple:

\[
POA_i = \langle i, PO \rangle
\]

where \( i \) is the ID number of the POA, PO is the coordinate of the POA in the CI.

According to Definition 2, a non-POA part of a CI is denoted as \( NPOA \).

Definition 3 (IB). An image block (IB) can be modeled as a triplet:

\[
IB = \langle bid, PO, TP \rangle
\]

where \( bid \) refers to the block ID, PO is the coordinate of the IB in the CI, and TP is the transmission priority of the block.

Definition 4 (CIB). Given a POA (i.e., \( POA_k \)) in a CI, a correlated image block (CIB) of \( POA_k \) is an IB which is contained in or intersects with it, formally denoted as: \( CIB = \{IB_{bid} | IB_{bid} \cap POA_k \neq \emptyset \} \), where \( k \in [1, |POA|] \) and \( |POA| \) means the number of the POAs in the CI.

Definition 5 (NIB). A NIB is an IB that is contained by a NPOA in a CI, formally represented by:

\[
NIB = \{IB_{bid} | IB_{bid} \cap NPOA = IB_{bid} \}.
\]

As indicated in introduction section, there are usually some lesion tissues that the doctors may focus on in the CISs. The region of such lesion organ in the CIS is called the pathological object area (POA). In the preprocessing step, the POAs need to be preliminarily marked by medical specialists; then each CI in the sequence is equally divided into some IBs (i.e., NIB and CIB) replicas, with the CIBs being encrypted and saved at their original pixel resolutions while the NIBs are stored at a lesser resolution. As illustrated in Fig 3, there are two POAs (A and B) and one NPOA (i.e., C) in an example CI which can be segmented into \( 6 \times 8 \) IBs marked by red dash lines.

**Methodologies**

In this section, we first introduce four supporting techniques based on which a BRS algorithm is proposed next.

**Supporting techniques**

To better facilitate the batch retrieval of the lung CISs in the MECN, in this subsection, we introduce four supporting techniques: 1) batch similarity measure for CISs, 2) CIB-based privacy preserving scheme, 3) uniform edge-cloud index framework, and 4) edge buffering.
Batch similarity measure for CISs. As mentioned before, a CIS $X_i$ is a time-series data which can be modeled by a vector: $X_i = \{CI_1, CI_2, \ldots, CI_{|X_i|}\}$, where $|X_i|$ means the number of CIs in $X_i$. Due to the large amount of the CIs in a CIS, to effectively reduce the high computation cost in the CIS similarity matching, we propose a representative CI(RCI)-based batch similarity measurement of the CISs.

Before introducing the batch similarity measure, how to extract the RCI s is a challenging issue. As summarized in Algorithm 1, given a CIS $X_i$, a RCI extraction processing of $X_i$ is first performed to obtain $||X_i||$ RCIs from a CIS, where $||X_i||$ means the number of RCIs in $X_i$, $d(x, y)$ is stated in Table 1, and $\epsilon$ is a small positive threshold.

Algorithm 1 RCI extraction ($X_i$)
\begin{enumerate}
  \item \textbf{input:} $X_i$
  \item \textbf{output:} $||X_i||$ RCIs
  \item $j \leftarrow 1$, $||X_i|| \leftarrow 1$
  \item \textbf{while} ($i < ||X_i||$) \textbf{do}
  \item \hspace{1em} \textbf{if} $d(CI_j, CI_{j+1}) > \epsilon$ \textbf{then}
  \item \hspace{2em} add $CI_{j+1}$ as the $||X_i||$-th RCI;
  \item \hspace{2em} $||X_i|| \leftarrow ||X_i|| + 1$
  \item \hspace{1em} \textbf{else}
  \item \hspace{2em} $j \leftarrow j + 1$
  \item \hspace{1em} \textbf{end if}
  \item \textbf{end while}
  \item \textbf{return} $||X_i||$ RCIs
\end{enumerate}

Once the $||X_i||$ RCIs are extraction from $X_i$, $X_i$ can be re-represented as:

$X_i = \{RCl_1, RCl_2, \ldots, RCl_{||X_i||}\}$. So given two CISs ($X_m$ and $X_n$), their batch similarity can be defined as follows:

$$
sim(X_m, X_n) = \frac{\sum_{RCl_i \in X_m} 1_{(RCl_i \preceq X_m \land d(RCl_i, RCl_j) \leq \epsilon)} + \sum_{RCl_i \in X_n} 1_{(RCl_i \preceq X_n \land d(RCl_i, RCl_j) \leq \epsilon)}}{|X_m| + |X_n|} \quad (4)$$

As can be seen from Eq (4), the similarity of two CISs can be measured by the percentage of similar RCIs in the two corresponding CISs.
CIB-based privacy-preserving scheme. Before introducing the CIB-based privacy-preserving scheme, let’s first give a definition.

Definition 6 (POAR). Given a POA (i.e., POA_j), its corresponding POA-related region (POAR) consists of all CIBs in POA_j, subjecting to the following criteria:

\[
\begin{align*}
& \text{POAR}(\text{POA}_j) \cap \text{POA}_j = \text{POA}_j \\
& |\text{POAR}(\text{POA}_j)| = |\text{POA}_j|
\end{align*}
\]

where POAR(POA_j) means the corresponding POAR of POA_j, |•| denotes the number of CIBs in •.

In Fig 3, there are two POAs (i.e., A and B) in the CI that is equally segmented into 6 x 8 IBs. Based on Definition 6, the corresponding POARs of the two POAs are represented by the green shadow areas which consist of 20 CIBs. Since the nearby CIB numbers have the characteristics of continuous distribution, it is easier to use these CIBs to reconstruct the original CI. As a result, the objective of the encryption strategy is to disrupt the continuity of the ID numbers of the nearby CIBs in the CI by encoding the ID numbers of the CIBs such that the CI reconstruction is hard to perform.

So for each CIB in a CI, we first introduce an encoding scheme (IBID) of the ID numbers of the above CIBs, which is represented in Eq (6):

\[\text{IBID} = \text{SID} \cdot c_1 + \text{IID} \cdot c_2 + r\text{ID} \cdot c_3 + c\text{ID}\]

where SID mean the ID of the CIS that the CIB belongs to, IID refers to the ID of the CI in which the CIB is contained, rID is row ID, cID is column ID, c1, c2, c3 are stretch constants and c1 >> c2 >> c3.

Based on the ID numbers of the CIBs in Eq (6), their encryption and decryption strategies are described as follows:

1) **Encryption strategy:**

Algorithm 2 details the steps of the CIB-based encryption processing in which the ID numbers of the CIBs are encrypted, where δ and ω are two key values and \(\delta < \text{SID}, \delta < \text{IID}\) and \(\omega < \text{rID}\).

**Algorithm 2 Encryption()**

**input:** SID, IID, rID, cID of a CIB  
**output:** IBID: the encrypted ID number of the CIB

1: if rID is an odd number and cID is an odd number then  
2: \(\text{IBID} = (\text{SID} + \delta) \cdot c_1 + (\text{IID} - \delta) \cdot c_2 + (\text{rID} + \omega) \cdot c_3 + c\text{ID}\)  
3: else if rID is an odd number and cID is an even number then  
4: \(\text{IBID} = (\text{SID} + \delta) \cdot c_1 + (\text{IID} - \delta) \cdot c_2 + (\text{rID} - \omega) \cdot c_3 + c\text{ID}\)  
5: else if rID is an even number and cID is an odd number then  
6: \(\text{IBID} = (\text{SID} - \delta) \cdot c_1 + (\text{IID} + \delta) \cdot c_2 + (\text{rID} + \omega) \cdot c_3 + c\text{ID}\)  
7: else  
8: \(\text{IBID} = (\text{SID} - \delta) \cdot c_1 + (\text{IID} + \delta) \cdot c_2 + (\text{rID} - \omega) \cdot c_3 + c\text{ID}\)  
9: end if  
10: return the encrypted IBID

2) **Decryption strategy:**

Similarly, for the encrypted CIBs, their corresponding decryption processing is discussed in Algorithm 3.

**Algorithm 3 Decryption()**

**input:** SID, IID, rID, cID of a CIB  
**output:** IBID: the encrypted ID number of the CIB

1: if rID is an odd number and cID is an odd number then  
2: \(\text{IBID} = (\text{SID} - \delta) \cdot c_1 + (\text{IID} + \delta) \cdot c_2 + (\text{rID} - \omega) \cdot c_3 + c\text{ID}\)
3: else if rID is an odd number and cID is an even number then
4:   \[\text{IBID} = (\text{SID} - \delta) \cdot c_1 + (\text{IID} + \delta) \cdot c_2 + (rID + \omega) \cdot c_3 + cID\]
5: else if rID is an even number and cID is an odd number then
6:   \[\text{IBID} = (\text{SID} + \delta) \cdot c_1 + (\text{IID} - \delta) \cdot c_2 + (rID - \omega) \cdot c_3 + cID\]
7 else
8:   \[\text{IBID} = (\text{SID} + \delta) \cdot c_1 + (\text{IID} - \delta) \cdot c_2 + (rID + \omega) \cdot c_3 + cID\]
9: end if
10: return the encrypted IBID

For instance, assume that SID is 7, IID is 4, \(c_1, c_2, c_3\) are 1000, 100 and 10, respectively, then the original ID numbers of the CIBs before encryption are depicted in Fig 4(a). Fig 4(b) shows the encrypted ID numbers of the CIBs after encryption when \(\delta = 3\) and \(\omega = 0.6\).

Fig 4(a) shows the continuous distribution of the ID numbers of the nearby original CIBs before encryption. After encryption, as illustrated in Fig 4(b), the ID number distribution of the nearby CIBs is discrete. Therefore, the encryption of the CIBs makes it more and more difficult to find the corresponding nearby CIBs in the CI reconstruction.

Next, we proceed to analyze the probability of the successful decryption (i.e., the probability of accurate image reconstruction). Given a CI with \(m\) POAs, for each POA (i.e., POA\(i\)), the rows and columns of its corresponding POAR can be denoted as \(RS_i\) and \(CS_i\), respectively. Then, the probability that the decryption processing is successful can be derived in Eq (7):

\[
\Pr(b\text{DecryptSuccess} = \text{TRUE}) = \frac{1}{\prod_{i=1}^{m} \prod_{j=0}^{RS_i - 1} (RS_i \cdot CS_i - j)}
\]  

Based on Eq (7), with increasing number of the CIBs in a CI, the probability of the successful decryption becomes smaller and smaller which guarantees the hardness of the decryption from a theoretical level. The encrypted CIBs are stored in \(N_C\) or \(N_E\) which ensures the corresponding CIBs’ IDs in a CI presents a discrete distribution rather than continuity to a certain extent. The reconstruction and display of the CIs are conducted at \(N_U\) by reversely decrypting the ID numbers of the CIBs based on the key values (i.e., \(\delta\) and \(\omega\)).
Uniform edge-cloud indexing framework. To support faster CIS filtering processing, we propose a uniform edge-cloud index framework (UECIF) based on iDistance [35], in which the UECIF is composed of two types of indexes: the elIndex in \( N_E \) and the clIndex in \( N_C \).

- Index Construction

For the elIndex, initially, suppose that the CISs in \( \Omega \) are virtually stored in \( N_E \), which means the CISs in \( \Omega \) are physically stored in \( N_E \), they are logically, however, not buffered in \( N_E \). Then, the CISs are first grouped into the \( K \) clusters by the AP-cluster [36] based on visual similarity (i.e., Eq (4)). Given a CIS \( X_i \), its index key can be represented below:

\[
key(X_i) = j \cdot c_i + sim(X_i, X'_i)
\]

where \( X'_i \) is the cluster centre of the \( j \)-th cluster that \( X_i \) belongs to, \( sim(\cdot, \cdot) \) represents the visual similarity distance function (i.e., Eq (4)), \( j \in [1, K] \), and the constant \( c_1 \) is used to stretch the value range.

The index key is inserted into an improved \( B'\)-Tree in which a leaf node (LNode) can be modeled by a triplet: \( LNode(X_i) = < key, value, EType >, \) where \( EType = 'F' \) means \( X_i \) is not buffered in \( N_E \) otherwise, \( X_i \) has been buffered in \( N_E \). Algorithm 4 summarizes the initial construction process of the elIndex in which \( LNode(X_i).EType \leftarrow 'F' \) (line 6) means all of the CISs are virtually stored in \( N_E \).

**Algorithm 4 elIndex construction(\( \Omega \))**

**input:** \( \Omega \): the CIS set

**output:** eIdx

1: eIdx\( \leftarrow \)NULL; \( \triangleright \) initialize
2: the CISs in \( \Omega \) are grouped into \( K \) clusters; \( \triangleright \) at edge node
3: for each CIS \( X_i \) in \( \Omega \) do
4: \( key(X_i) = j \cdot c_i + sim(X_i, X'_i) \);
5: insert \( key(X_i) \) into an improved \( B'\)-tree (i.e., eIdx);
6: \( LNode(X_i).EType \leftarrow 'F' \);
7: end for
8: return the eIdx;

Similar to the above, for the clIndex, first of all, the clustering processing of the CISs(\( \Omega \)) in \( N_C \) is performed to obtain \( T \) clusters based on the above visual similarity. For a CIS \( X_i \), its index key can be derived as:

\[
KEY(X_i) = j \cdot c_i + sim(X_i, X'_i)
\]

where \( j \in [1, T] \), and other parameters and symbols are the same to that of in Eq (8). In Algorithm 5, the index key is inserted into an improved \( B'\)-Tree in which a leaf node(LNode) can be represented by a triplet: \( LNode(X_i) = < key, value, CType >, \) where \( LNode(X_i).CType \leftarrow 'T' \) means \( X_i \) is stored in \( N_C \).

**Algorithm 5 clIndex construction(\( \Omega \))**

**input:** \( \Omega \): the CIS set

**output:** cIdx

1: cIdx\( \leftarrow \)NULL; \( \triangleright \) initialize
2: the CISs in \( \Omega \) are grouped into \( T \) clusters; \( \triangleright \) at cloud node
3: for each CIS \( X_i \) in \( \Omega \) do
4: \( KEY(X_i) = j \cdot c_i + sim(X_i, X'_i) \);
5: insert \( KEY(X_i) \) into an improved \( B'\)-tree (i.e., cIdx);
6: \( LNode(X_i).CType \leftarrow 'T' \);
7: end for
8: return the cIdx;

- Index-Support Retrieval Processing
Based on Eqs (8) and (9), suppose that there are \( n \) CISs in \( \Omega \), the index keys are inserted by an improved \( B^+ \)-Tree respectively. So for a range retrieval \( \Theta(X_r, r_R) \) and each cluster \( C_j \), as illustrated in Fig 5, there are five cases in terms of the positions of the two spheres.

Case 1: in Fig 5(a), the inequalities \( \text{sim}(X'_i, X_R) - r_R < r'_i \) and \( \text{sim}(X'_i, X_R) < r_R \) are met, which means \( \Theta(X_r, r_R) \) intersects with \( \Theta(X_i, r'_i) \) by which \( X_R \) is contained. So the search range is represented as: \( [j \cdot c_i, j \cdot c_i + d(X'_i, X_R) + r'_i] \).

Case 2: in Fig 5(b), the inequalities \( \text{sim}(X'_i, X_R) - r_R \leq r'_i \) and \( \text{sim}(X'_i, X_R) \geq r_R \) are met, which means \( \Theta(X_r, r_R) \) intersects with \( \Theta(X'_i, r'_i) \) and \( \Theta(X_i, r'_i) \) does not contain \( X_R \). So the search range is represented as: \( [j \cdot c_i + \text{sim}(X'_i, X_R) - r_R, j \cdot c_i + r'_i] \).

Case 3: in Fig 5(c), the inequality \( \text{sim}(X'_i, X_R) + r'_i \leq r_R \) is met, which means \( \Theta(X_r, r_R) \) contains \( \Theta(X_i, r'_i) \). So the search range is represented as: \( [j \cdot c_i, j \cdot c_i + r'_i] \).

Case 4: in Fig 5(d), the inequality \( \text{sim}(X'_i, X_R) + r_R \leq r'_i \) is met, which means \( \Theta(X_i, r'_i) \) contains \( \Theta(X_r, r_R) \). So the search range is represented as: \( [j \cdot c_i, j \cdot c_i + r'_i] \).

Case 5: in Fig 5(e), the inequality \( \text{sim}(X'_i, X_R) > r'_i + r_R \) is met, which means \( \Theta(X_r, r_R) \) does not intersect with \( \Theta(X_i, r'_i) \). No candidate sequences are retrieved.

For the similarity retrieval in the MECN, there are two cases in terms of whether there exists a partial answer in \( N_E \): 1) the complete answer sequences (\( \Psi \)) are directly obtained from \( N_C \) based on cIndex (see Algorithm 8); 2) the complete answer sequences (\( \Psi \)) are composed of the partial answer ones (\( \Psi' \)) obtained from \( N_E \) (see Algorithm 6) and the partial answer ones (\( \Psi'' \)) from \( N_C \) (see Algorithm 7).

Algorithm 6 details the similarity range retrieval of the CISs based on the elIndex in \( N_E \). The routing \( \text{Search}() \) is the implementation of the range similarity search in the improved \( B^+ \)-Tree which is described in Algorithm 9.

**Algorithm 6** \( \text{Search}(X_R, r_R, \Omega') \)

- **input:** \( \Theta(X_r, r_R) \): the retrieval CIS,
- **Q':** the CIS whose ETTypes are ‘\( V \)’ in \( N_E \)
- **output:** \( \Psi' \): the partial answer CISs from \( N_E \)

1. \( \Psi' \leftarrow \Phi \); \> initialization
2. \( \Psi' \leftarrow \text{Search}(X_R, r_R, \Omega') \)
3. **for each candidate** CIS\((X_i) \in \Psi' \) **do**
   4. **if** \( \text{sim}(X_r, X_i) > r_R \) **then**
   5. \( \Psi' \leftarrow \Psi' - X_i \)
Algorithm 7 CSearch($X_R, r_R, \Omega'$)
\textbf{input}: $\Theta(X_R, r_R)$: the retrieval CIS,$\Omega'$: the CIS whose ETypes are 'T' in $N_c$
\textbf{output}: $\Psi'$: the partial answer CISs from cloud node
1: $\Psi' \leftarrow \Phi$; \hspace{1em} \textit{▷} initialization
2: $\Psi' \leftarrow \text{Search}(X_R, r_R, \Omega')$
3: for each candidate CIS($X_i$) $\in \Psi'$ do
4: \hspace{1em} if $\text{sim}(X_R, X_i) > r_R$ then
5: \hspace{2em} $\Psi' \leftarrow \Psi' - X_i$
6: \hspace{1em} else if LNode($X_i$).EType $= 'F'$ then
7: \hspace{2em} LNode($X_i$).EType $\leftarrow 'T'$; \hspace{1em} \textit{▷} for eIndex
8: \hspace{2em} LNode($X_i$).CType $\leftarrow 'F'$; \hspace{1em} \textit{▷} for cIndex
9: \hspace{1em} end if
10: \hspace{1em} end if
11: \hspace{1em} end if
12: \hspace{1em} end for
13: return $\Psi'$

Similarly, Algorithm 7 summarizes the index support partial similarity range retrieval of the CISs at the cloud node level. It is worth mentioning that LNode($X_i$).CType $= 'T'$ means $X_i$ is not buffered in $N_F$.

Algorithm 8 DSearch($X_R, r_R, \Omega$)
\textbf{input}: $\Theta(X_R, r_R)$: the retrieval CIS,$\Omega$: the CIS in $N_c$
\textbf{output}: $\Psi$: the complete answer CISs from $N_c$
1: $\Psi \leftarrow \Phi$ \hspace{1em} \textit{▷} initialization
2: $\Psi \leftarrow \text{Search}(X_R, r_R, \Omega)$
3: for each ($X_i$) $\in \Psi$ do
4: \hspace{1em} if $\text{sim}(X_R, X_i) > r_R$ then
5: \hspace{2em} $\Psi \leftarrow \Psi - X_i$
6: \hspace{1em} end if
7: \hspace{1em} end for
8: return $\Psi$

Algorithm 9 Search($X_R, r_R, \Omega$)
\textbf{input}: $\Theta(X_R, r_R)$: the retrieval CIS,$\Omega$: the CISs
\textbf{output}: $\Phi'$: the candidate CISs
1: $\Phi' \leftarrow \text{NULL}$; \hspace{1em} \textit{null}$\Phi'$
2: for the CISs in each cluster $C_j$ do
3: \hspace{1em} if $\text{sim}(X_R, X_j) - r_R < r'_j$ and $\text{sim}(X_R, X_j) < r_R$ then
4: \hspace{2em} $left = j \cdot c_j$, $right = j \cdot c_j + \text{sim}(X_R, X_j) + r'_j$
5: \hspace{2em} else if $\text{sim}(X_R, X_j) - r_R \leq r'_j$ and $\text{sim}(X_R, X_j) \geq r_R$ then
6: \hspace{2em} $left = j \cdot c_j + \text{sim}(X_R, X_j) - r_R$, $right = j \cdot c_j + r'_j$
7: \hspace{2em} else if $\text{sim}(X_R, X_j) + r'_j \leq r_R$ then
8: \hspace{2em} $left = j \cdot c_j$, $right = j \cdot c_j + r'_j$
9: \hspace{2em} else if $\text{sim}(X_R, X_j) + r_R \geq r'_j$ then

Finally, obtaining the complete answer CISs from the cloud node is detailed in Algorithm 8.
Index Update

When user submits a retrieval request, the eIndex needs to be updated by adding the CISs that have been accessed in this retrieval. Since the number of CISs buffered in $N_E$ is limited, how to optimally choose the buffered CISs is challenging.

For example, assume that there are six CISs in $N_E$. Tables 2 and 3 illustrate the ranking of access time (AT) and access frequencies (AF) for the six CISs, respectively. In Table 2, the ATs of the six CISs are sorted in an ascending order, which are quantitatively represented by the AT_IDs. Then a weighted AT (WAT) can be derived as follows:

$$WAT(CIS_j) = \frac{AD_{ID_j}}{\arg\max_{i \in [1,n]}(AT_{ID_i})}$$

Similarly, for the ranking of the access frequencies (AF), a weighted AF (WAF) is represented in Eq (11):

$$WAF(CIS_j) = 1 - \frac{AF_j}{\arg\max_{i \in [1,n]}(AF_i)}$$

Based on Eqs (10) and (11), given a CIS, its uniform ranking score (URS) is shown below:

$$URS(CIS_j) = WRA(CIS_j) + WAF(CIS_j)$$

Table 4 illustrates the uniform ranking scores of the six CISs. Based on Eq (12), the smaller the URS, the more important the CIS. If $MaxN$ is 4, then CIS3 and CIS4 can be removed from the edge buffer.

Algorithm 10 Update($\Omega'$)

input: $\Omega'$: the CISs buffered at $N_E$
output: the updated $\Omega'$

1: for $i = 1$ to $MaxN - |\Omega'| - 1$
2: remove a CIS$_j$ whose URS is the largest from $\Omega'$;
3: $\Omega' \leftarrow \Omega' - CIS_j$;

Table 2. Ranking of the AT.

| CIS   | AT  | AT_ID | WAT   |
|-------|-----|-------|-------|
| CIS1  | 2.50| 4     | 4/6   |
| CIS2  | 3.40| 2     | 2/6   |
| CIS3  | 1.48| 6     | 6/6   |
| CIS4  | 4.21| 1     | 1/6   |
| CIS5  | 2.32| 5     | 5/6   |
| CIS6  | 3.11| 3     | 3/6   |

https://doi.org/10.1371/journal.pone.0274507.t002

Table 3. Ranking of the AF.

| CIS   | AF  | WAF  |
|-------|-----|------|
| CIS1  | 1   | 4/5  |
| CIS2  | 3   | 2/5  |
| CIS3  | 1   | 4/5  |
| CIS4  | 4   | 1/5  |
| CIS5  | 2   | 3/5  |
| CIS6  | 3   | 2/5  |

https://doi.org/10.1371/journal.pone.0274507.t003
new retrieval to the fact that the answer decreased with the help of the buffering scheme since a portion of answer will greatly shorten the long-distance transmission delay and improve the retrieval efficiency.

Unlike the traditional image retrieval methods, which directly obtain data from the remote cloud, if the answer CISs can be directly obtained from the edge without accessing the cloud, it will greatly shorten the long-distance transmission delay and improve the retrieval efficiency. Based on the above motivation, we propose an edge buffering scheme by analyzing the user historical retrieval (HR) log file. The refinement cost of the candidate CISs can be significantly decreased with the help of the buffering scheme since a portion of answer CISs can be retrieved directly without any refinement processing.

Specifically, assume that n HRs have been successfully completed with accurate results. Due to the fact that the answer CISs provided by each HR have been verified, when a user submits a new retrieval CIS (i.e., $X_R$), it is highly possible that $X_R$ may be similar or even the same as the HR one (i.e., $X_R^n$). As a result, the retrieval efficiency and accuracy can be greatly improved if the HR results in $N_E$ can be carefully reused as a part of the current results.

Definition 7(CRS). Given a retrieval CIS $X_R$ and a retrieval radius $r_R$, their corresponding CIS retrieval sphere (CRS) is a high-dimensional sphere with a centre $X_R$ and a radius $r_R$, denoted as $\Theta(X_R, r_R)$.

Definition 8(HCRS). Given a HR CIS $X^n_R$ and a retrieval radius $r^n_R$, their corresponding historical CIS retrieval sphere (HCRS) is a high-dimensional sphere with a centre $X^n_R$ and a radius $r^n_R$, denoted as $\Theta(X^n_R, r^n_R)$.

Definition 9(AA). Given a CRS $\Theta(X_R, r_R)$ and a HCRS $\Theta(X^n_R, r^n_R)$, their corresponding affected area (AA) is the intersection part of the two spheres, formally denoted as: $AA(\Theta(X_R, r_R) \cup \Theta(X^n_R, r^n_R))$.

For example, as shown in Fig 6, there are three HCRSs, i.e., $\Theta(X^1_R, r^1_R), \Theta(X^2_R, r^2_R)$ and $\Theta(X^3_R, r^3_R)$. The current CRS is represented as: $\Theta(X_R, r_R)$. For $X_R$, it’s corresponding 1st, 2nd and 3rd nearest neighbor CISs are $X^1_R$, $X^2_R$ and $X^3_R$, respectively. Therefore, the HR of $\Theta(X^2_R, r^2_R)$ can be safely discarded since its corresponding HCRS does not intersect with $\Theta(X_R, r_R)$. The CISs falling in the AA (i.e., $\Theta(X_R, r_R) \cup \Theta(X^1_R, r^1_R)$ and $\Theta(X_R, r_R) \cup \Theta(X^3_R, r^3_R)$) can be a part of the answer CISs of $\Theta(X_R, r_R)$.

Next, given two CIS retrieval spheres: $\Theta(X_R, r_R)$ and $\Theta(X^n_R, r^n_R)$, there exists two cases on the basis of the two retrieval CISs (i.e., $X_R$ and $X^n_R$), which are shown in Figs 7 and 8, respectively.

In Fig 7, if $X_R = X^n_R$, then there exists two cases in terms of the retrieval radii (i.e., $r_R$ and $r^n_R$).

4: LNode(CIS$_i$).CType ← ‘F’;
5: LNode(CIS$_j$).EType ← ‘F’;
6: end for
7: return the updated $\Omega$

**Table 4. Uniform ranking score.**

|   | CIS$_1$ | CIS$_2$ | CIS$_3$ | CIS$_4$ | CIS$_5$ | CIS$_6$ |
|---|---------|---------|---------|---------|---------|---------|
| WRA | 4/6     | 2/6     | 6/6     | 1/6     | 5/6     | 3/6     |
| WAF | 4/5     | 2/5     | 4/5     | 1/5     | 3/5     | 2/5     |
| URS | 44/30   | 22/30   | 54/30   | 11/30   | 43/30   | 27/30   |

https://doi.org/10.1371/journal.pone.0274507.t004

**Edge buffering**

Edge buffering

Privacy-preserving batch retrieval of lung CT image sequences
Fig 6. An example of the edge buffering scheme.

https://doi.org/10.1371/journal.pone.0274507.g006

Fig 7. \( X_s = X_0^s \).

https://doi.org/10.1371/journal.pone.0274507.g007

Fig 8. \( X_s \neq X_0^s \) (a). \( AA(\Theta(X_s, r_s), \Theta(X_0^s, r_0^s)) = \text{NULL} \), (b). \( AA(\Theta(X_s, r_s), \Theta(X_0^s, r_0^s)) \neq \text{NULL} \), (c). \( AA(\Theta(X_s, r_s), \Theta(X_0^s, r_0^s)) = \Theta(X_s, r_s) \), (d). \( AA(\Theta(X_s, r_s), \Theta(X_0^s, r_0^s)) = \Theta(X_s, r_s) \).

https://doi.org/10.1371/journal.pone.0274507.g008
• For case (b) which is formally represented as: $AA(\Theta(X_R, r_R), \Theta(X'_R, r'_R)) = \Theta(X_R, r_R)$, the answer CISs for $\Theta(X_R, r_R)$ can be derived from the CISs in $\Theta(X'_R, r'_R)$.

In Fig 8, if $X_R \neq X'_R$, there are four cases according to the placement of the two spheres (i.e., $\Theta(X_R, r_R)$ and $\Theta(X'_R, r'_R)$):

• In case (a), as the AA of the above two spheres does not exist, formally represented as: $AA(\Theta(X_R, r_R), \Theta(X'_R, r'_R)) = NULL$, the answer CISs need to be calculated sequentially in $\Theta(X_R, r_R)$;

• In case (b), as the AA of the above two spheres exists, formally represented as: $AA(\Theta(X_R, r_R), \Theta(X'_R, r'_R)) \neq NULL$. Since the CISs falling in $\Theta(X'_R, r'_R)$ have been verified previously, they can be regarded as a part of a candidate CIS set of $\Theta(X_R, r_R)$;

• In case (c), as the AA of the above two spheres is $\Theta(X'_R, r'_R)$, formally represented as: $AA(\Theta(X_R, r_R), \Theta(X'_R, r'_R)) = \Theta(X'_R, r'_R)$. As the CISs that fall in $\Theta(X'_R, r'_R)$ have been verified previously; they can be regarded as a part of an answer CIS set of $\Theta(X_R, r_R)$;

• In case (d), as the AA of the above two spheres is $\Theta(X_R, r_R)$, formally represented as: $AA(\Theta(X_R, r_R), \Theta(X'_R, r'_R)) = \Theta(X_R, r_R)$, the answer CISs are contained by the CISs falling in $\Theta(X'_R, r'_R)$.

The BRS algorithm

Before introducing the algorithm, a pre-processing step is first conducted. Algorithm 11 summarizes the detailed steps of our proposed BRS method in which $ESearch(X_R, r_R)$, $CSearch(X_R, r_R)$ and $DSearch(X_R, r_R)$ correspond to Algorithms 6-8, respectively. As illustrated in Fig 9, first
of all, when a retrieval lung CIS \((X_R)\) is submitted to the edge node level \(N_E\) from the user one \(N_U\), then the eIndex scheme in \(N_E\) is adopted to quickly judge whether there are some answer CISs similar to \(X_R\). If exists, then the high-dimensional similarity retrieval is carried out with the support of the cIndex scheme at the cloud to obtain some partial retrieval answer CISs; otherwise, the similarity retrieval of all CISs supported by the cIndex is performed directly through the cloud, and finally the answer CISs are returned to the receiver node. Note that, in line 9, before transmitting the answer CISs to the receiver, the decryption processing of the CIBs in the CISs need to be performed to ensure the accurate reconstruction and display of the answer CISs. Compared to NIBs, the CIBs have higher transmission priorities. In accordance with the various priorities of the IBs, they can be transmitted in descending order of priority, which not only assures the security of data transmission but also ensures that the critical information can be transmitted first.

Algorithm 11 \(BRS(X_R, r_R)\)

input: \(X_R\): a retrieval CIS, \(r_R\): a retrieval radius

output: \(\Psi\): the answer CISs

1: a retrieval CIS \((X_R)\) is submitted from \(N_U\);
2: \(\Psi' \leftarrow ESearch(X_R, r_R)\); \(\triangleright\) obtain answer CISs \((\Psi')\) based on the eIndex at the edge node \(U_E\);
3: \(\Psi' \neq \text{NULL} \) then
4: \(\Psi'' \leftarrow CSearch(X_R, r_R)\); \(\triangleright\) obtain the partial answer CISs \((\Psi'')\) based on cIndex at the cloud \(U_C\);
5: \(\Psi \leftarrow \Psi' \cup \Psi''\);
6: else
7: \(\Psi \leftarrow DSearch(X_R, r_R)\); \(\triangleright\) obtain the complete answer CISs based on cIndex at the cloud \(U_C\);
8: end if
9: transmit the CISs in \(\Psi\) to the receiver node level with different transmission priorities

Experiments

To verify the efficiency of the proposed BRS method, extensive simulation experiments are conducted to demonstrate the retrieval performance.

Experimental setup. In the experiments, the image receiver client is equipped with a 5.9-inch, full HD 1080p screen and a Qualcomm® Snapdragon™ 650 processor running at 1.7GHz. The client system is developed in Java and operates on the Android operating system [37]. The edge node and the cloud one are connected via 1Gbps network links. In the cloud node, the IBs (i.e., CIB and NIB) with various transmission priorities are kept in a file system and some structured data is recorded by the MySQL [38]. Each node contains a 2.7 GHz quad-core Xeon processor, 2.0 Gigabyte memory, and 1.0 Terabyte hard disk. The maximum data communication rate is 150 Mbps in the wireless network.

We selected the LUNA16 dataset [39], which contains 239,232 lung CISs, as our experimental dataset. There are 888 lung CISs in this database, with an average of 336 lung CISs each set. The lung CISs in each set range in level from 200 to 600.

A prototype retrieval system. Fig 10 depicts a demonstration of the prototype system. An example of the CIS pre-processing backend interface is shown in Fig 10(a) in which a POA as been marked by a blue rectangle line. In Fig 10(b), a CIS with the category 'lung' has been inputted as a retrieval sequence. Four result CISs were quickly retrieved, and their matching IBs are restored and shown.

Effectiveness of the BRS method. The first experiment testifies the effectiveness of our BRS method by using the lung CISs randomly selected as experimental data. The recall and
precision achieved by this retrieval method can be defined as:

\[
\text{recall} = \frac{|\text{rel} \cap \text{ret}|}{|\text{ret}|}, \quad \text{precision} = \frac{|\text{rel} \cap \text{ret}|}{|\text{rel}|}
\]

where \( \text{rel} \) means the set of ground-truth, and \( \text{ret} \) refers to the set of results returned by a similarity range search.

As shown in Fig 11, performance comparisons of the retrieval effectiveness of the 10 CISs with the same organ (i.e., lung) that are randomly selected from the database are conducted. As can be observed from the figure, precision steadily declines as recall ratio rises. The reason is that when the recall rate is low, it’s highly possible that the correctness rate of the result CISs is high. Meanwhile, the high recall rate can not guarantee that the retrieval results contain the correct CISs.

**Effect of data size.** In this experiment, we investigate the effect of data size (i.e., the number of the lung CISs) on the retrieval efficiency by using the two methods: 1) our proposed BRS method; 2) The MIRC method in [25]. In this experiment, the network bandwidth is 100Mbps and the number of edge nodes is 15, and the UECI framework is used. In Fig 12, with the increase of the CISs, the BRS method is superior to the MIRC since the edge buffer is verified to significantly reduce the retrieval computation cost and transmission delay. Meanwhile, it is interesting to observe that as the data size increases, the overall response time first grows rapidly and then gradually. This is because the index performs better when there is more data.

**Effect of \( \varepsilon \).** The experiment evaluates the effect of \( \varepsilon \) on the retrieval performance. Similar to the above experiment, the network bandwidth and the number of cloud(edge) nodes are fixed, and the edge buffering scheme and the indexing mechanism are adopted. As demonstrated in Fig 13(a), with the increase of \( \varepsilon \), the CPU cost for the similarity computation is decreasing due to the decrease of the number of the RCIs in each CIS. Meanwhile, it’s interesting to note in Fig 13(b) that the precision ratio increases rapidly first and then decreases gradually. The reason is that too many or too few RCIs will make it difficult to accurately and completely measure the similarity of the CISs. Therefore, an optimal \( \varepsilon \) is set to be 0.65.
Effect of edge buffering scheme. In this experiment, we proceed to study the effect of the edge buffering scheme on the retrieval performance. Method 1 adopts the edge buffering scheme and method 2 do not use it. Fig 14 demonstrates that the overall response time using method 1 is faster than method 2 when the bandwidth is stable and the retrieval radius \( r_R \) is fixed. Meanwhile, the performance gap widens as \( r_R \) steadily grows while the band-width...
remains constant. This is because with the increase of $r_0$, the probability of obtaining the result CISs in the edge buffering is also increasing.

**Effect of indexing scheme.** The final experiment examines how the index framework (i.e., $eIndex$ and $cIndex$) affects retrieval efficiency. Here, method 1 uses the aforementioned two indexes, whereas method 2 does not (i.e., it sequentially searches each cloud node to find the answer CISs). In Fig 15, when the data size and the network bandwidth are fixed, the number of the cloud nodes varies from 10 to 50, the response time for the method 1 (i.e., index-based retrieval) is growing with the number of the cloud nodes increases. Meanwhile, the performance gap of the two approaches becomes smaller since the response time for method 2 is relatively stable and locating the corresponding candidate CISs based on the index is faster.
than that of no index. It’s interesting to notice that the retrieval response time is the smallest when the number of cloud nodes is 10. The larger the number of cloud nodes involved in retrieval, a large amount of data exchange and transmission will occur, resulting in retrieval delay.

Conclusion
In this paper, we introduced the BRS method—a privacy-preserving batch retrieval of the lung CISs in edge-cloud collaborative computing environment. The goal of our proposed BRS is to provide a safe and efficient retrieval of the lung CISs in resource-constraint network with low and unstable network bandwidth. To enable the efficient BRS processing, four supporting techniques are proposed, namely, 1) batch similarity measure for CISs, 2) CIB-based privacy preserving scheme, 3) uniform edge-cloud index framework, and 4) edge buffering scheme. The experimental results reveal that the efficiency of the BRS method is more than 200% higher than that of the sequential retrieval with the aid of the supporting techniques, especially when the number of cloud nodes is smaller.

Acknowledgments
The authors would like to thank the editors and anonymous reviewers for their helpful comments. Special thanks to Dr. Yujia Ge for her editing.

Author Contributions
Methodology: Nan Jiang.

Writing – original draft: Yi Zhuang.
References

1. Welter P, Fischer B, Günther RW, Deserno TM. Generic integration of content-based image retrieval in computer-aided diagnosis. Computer Methods & Programs in Biomedicine, 108(2):589–599. 2012. https://doi.org/10.1016/j.cmpb.2011.08.010 PMID: 21975083
2. Alzubi J, Alzubi O, Singh A, et al. Cloud-IIoT Based Electronic Health Record Privacy-Preserving by CNN and Blockchain-Enabled Federated Learning. IEEE Transactions on Industrial Informatics. pp:1-8. July 2022
3. Razaque A, Aloqaily M, Almiani M, et al Efficient and reliable forensics using intelligent edge computing. Future Generation Computer Systems. pp.230–239. 1999. https://doi.org/10.1016/j.future.2021.01.012
4. Alzubi O, Alzubi J, Shankar K, Gupta D. Blockchain and artificial intelligence enabled privacy-preserving medical data transmission in Internet of Things. Transactions on Emerging Telecommunications Technologies. Vol.32, Issue12. December 2021. e4360
5. Nazir S, Alzubi O, Mohammad K, et al. Image subset communication for resource-constrained applications in wireless sensor networks, Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 28: No. 5, Article 21. 2020. https://doi.org/10.3906/elk-2002-169
6. Philbin J, Chum O, Isard M, et al. Object retrieval with large vocabularies and fast spatial matching. International Conference on Computer Vision and Pattern Recognition. 2007.
7. Flickner M, Sawhney H, Niblack W, Ashley J. Query by image and video content: The QBIC system. Computers. 28(9). 23–31. 1995. https://doi.org/10.1109/2.410146
8. Rui Y, Huang T.S, Chang S.F. Image Retrieval: Current Techniques, Promising Directions and Open Issues, Journal of Visual Communication and Image Representation, 10. pp.39–62. 1999. https://doi.org/10.1006/jvci.1999.0413
9. Smith J, Chang S.F. VisualSEEK: a fully automated content-based image query system. ACM Multimedia. pp.89–98, 1996.
10. Smith J, Chang S.F. Visually Searching the Web for Content, IEEE Multimedia. 4(3): 12–20, 1997. https://doi.org/10.1109/93.621578
11. Shyu C.R, Brodley C.E, Kak A.C, et al. ASSERT : a physician-in-the-loop content-based retrieval system for HRCT image databases. Computer Vision and Image Understanding. 75. pp.111–132. 1999. https://doi.org/10.1006/cviu.1999.0768
12. Deselaers T. Features for Image Retrieval [dissertation] Aachen, Germany: Rheinisch-Westfälische Technische Hochschule Aachen. 2003.
13. Deselaers T. Fire [Internet] Tomas Deselaers; Available from: http://thomas.deselaers.de/fire. 2009.
14. Huang YG, Zhang J, Huang HY, Wang DF. Medical image retrieval based on unclean image bags. Multimedia Tools and Applications. 72(3), pp 2977–2999. 2014. https://doi.org/10.1007/s11042-013-1589-3
15. Huang YG, Huang HY, Zhang J. A noisy-smoothing relevance feedback method for content-based medical image retrieval. Multimedia Tools and Applications. 73(3), pp. 1963–1981. 2014. https://doi.org/10.1007/s11042-013-1685-4
16. Kitanovski I, Strezoski G, Dimitrovski I. Multimodal medical image retrieval system. Multimedia Tools and Applications. 76(2), pp. 2955–2978. 2017. https://doi.org/10.1007/s11042-016-3261-1
17. Lan RS, Zhong S, Liu ZB, Shi Z. A simple texture feature for retrieval of medical images. Multimedia Tools and Applications. 77(9), pp. 10853–10866. 2018. https://doi.org/10.1007/s11042-017-5341-2
18. Ali M, Dong L, Akhtar R. Multi-panel medical image segmentation framework for image retrieval system. Multimedia Tools and Applications. 77(16), pp. 20271–20295. 2018. https://doi.org/10.1007/s11042-017-5453-8
19. Kasban H, Salama D.H. A robust medical image retrieval system based on wavelet optimization and adaptive block truncation coding. Multimedia Tools and Applications. 2019. https://doi.org/10.1007/s11042-019-08100-3
20. Tuyet V.T.H, Binh N.T, Ouoc N.K, et al. Content Based Medical Image Retrieval Based on Salient Regions Combined with Deep Learning. Mobile Network and Application, 26, pp.1300–1310. 2021. https://doi.org/10.1007/s11036-021-01762-0
21. Elkamiem A.F.A, Bashir M.B, Ahmed T.H, et al. Distributed medical image retrieval techniques: A review. 2017 Sudan Conference on Computer Science and Information Technology. 2017.
22. Anbarasi M.S, Mehata K.M, Sandhya S, Suganya V. Medical image retrieval from distributed environment. International Conference on Intelligent Agent & Multi-Agent Systems. 2009.
23. Charisi A, Megalooikonomou V. Content-based medical image retrieval in peer-to-peer systems. ACM International Health Informatics Symposium (IHI’10). 724-733. 2010.
24. Depeursinge A, Duc S, Eggel I, et al. Mobile medical visual information retrieval. IEEE Trans. on Information Technology in Biomedicine. 16(1), pp. 53–61, Jan. 2012. https://doi.org/10.1109/TITB.2011.2173585 PMID: 22157061

25. Zhuang Y, Jiang N, Wu Z.A., Li Q, et al. Efficient and robust large medical image retrieval in mobile cloud computing environment. Information Sciences. Vol.263. pp.60–86, 2014. https://doi.org/10.1016/j.ins.2013.10.013

26. Zhuang Y, Jiang N, Li Q, Chen L, Ju C.H. Progressive batch medical image retrieval processing in mobile wireless network. ACM Trans on Internet Technology. 15(3), Article 9, 2015. https://doi.org/10.1145/2783437

27. Cruz A.L, Medina R, Vega F, et al. Mobile teleradiology system suitable for m-health services supporting content and semantic based image retrieval on a grid infrastructure, 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2016.

28. Chitra S, Kumaratharan N, Ramesh S. Enhanced brain image retrieval using carrier frequency offset compensated orthogonal frequency division multiplexing for telemedicine applications. International Journal of Imaging Systems and Technology. 28(3), pp. 186–195. 2018. https://doi.org/10.1002/ima.22269

29. Jiang N, Zhuang Y, Chiu D. K.W. Towards Effective and Efficient Framework for Crowd-Assisted Mobile Similarity Retrieval of Medical Images in Mobile Telemedicine Systems. Multimedia Tools and Applications. 79, pp.19893–19923. 2020. https://doi.org/10.1007/s11042-020-08755-3

30. Lei Y, Xu D, Zhou Z-Y, Higgins K, et al. High-resolution CT image retrieval using sparse convolutional neural network, SPIE 10573, Medical Imaging 2018: Physics of Medical Imaging, 105733F (9 March 2018)

31. Yu M, Lu Z-T, Feng Q-J, Chen W-F. Liver CT image retrieval based on non-tensor product wavelet. International Conference of Medical Image Analysis and Clinical Application. 2010.

32. Hatibaruah R, Nath V. K. and Hazarika D. Computed tomography image retrieval via combination of two local bit plane-based dissimilarities using an adder. International Journal of Wavelets, Multi-resolution and Information Processing vol. 19, no. 01 2021

33. Hwang H-J, Seo J-B, Lee S-M, et al. Content-Based Image Retrieval of Chest CT with Convolutional Neural Network for Diffuse Interstitial Lung Disease: Performance Assessment in Three Major Idiopathic Interstitial Pneumonias. Korean Journal of Radiol. 22(2); pp.281–290. 2021. https://doi.org/10.3348/kjr.2020.0603 PMID: 33169547

34. Alzubi J.A, Bharathikannan B, Tanwar S, Manikandan R, Khanna A, Thaventhiran C, Boosted neural network ensemble classification for lung cancer disease diagnosis, Applied Soft Computing, Vol. 80, pp. 579–591, July 2019. https://doi.org/10.1016/j.asoc.2019.04.031

35. Jagadish H.V, Ooi B.C, Tan K.L, et al. iDistance: An Adaptive B+-tree Based Indexing Method for Nearest Neighbor Search. ACM Trans on Database Systems, 30(2), 364–397. 2005. https://doi.org/10.1145/1071610.1071612

36. Frey B.J, Dueck D. Clustering by passing messages between data points, Science. 315 (5814), 972–976. 2007. https://doi.org/10.1126/science.1136800 PMID: 17218491

37. Android. The Android platform. http://code.google.com/intl/zh-CN/android/. 2010

38. MySQL. http://www.mysql.com/. 2011.

39. The Luna16 Dataset. https://luna16.grand-challenge.org/. 2016