Medical & Biological Engineering & Computing (2021) 59:1993–2017
https://doi.org/10.1007/s11517-021-02392-0

ORIGINAL ARTICLE

Text-based multi-dimensional medical images retrieval according to the features-usage correlation

AliAsghar Safaei[1]

Received: 7 July 2020 / Accepted: 13 June 2021 / Published online: 20 August 2021

Abstract

Emerging medical imaging applications in healthcare, the number and volume of medical images is growing dramatically. Information needs of users in such circumstances, either for clinical or research activities, make the role of powerful medical image search engines more significant. In this paper, a text-based multi-dimensional medical image indexing technique is proposed in which correlation of the features-usages (according to the user's queries) is considered to provide an *off-the content* indexing while taking users' interestingness into account. Assuming that each medical image has some extracted features (e.g., based on the DICOM standard), correlations of the features are discovered by performing data mining techniques (i.e., *quantitative association pattern discovery*), on the history of users' queries as a data set. Then, based on the pairwise correlation of the features of medical images (a.k.a. *Affinity*), set of the all features is fragmented into subsets (using method like the *vertical fragmentation* of the tables in distribution of relational DBs). After that, each of these subsets of the features turn into a hierarchy of the features (by applying a hierarchical clustering algorithm on that subset), subsequently all of these distinct hierarchies together make a multi-dimensional structure of the features of medical images, which is in fact the proposed text-based (feature-based) multi-dimensional index structure. Constructing and using such text-based multi-dimensional index structure via its specific required operations, medical image retrieval process would be improved in the underlying medical image search engine. Generally, an indexing technique is to provide a logical representation of documents in order to optimize the retrieval process. The proposed indexing technique is designed such that can improve retrieval of medical images in a medical image search engine in terms of its *effectiveness* and *efficiency*. Considering correlation of the features of the image would semantically improve precision (effectiveness) of the retrieval process, while traversing them through the hierarchy in one dimension would try to optimize (i.e., minimize) the resources to have a better efficiency. The proposed text-based multi-dimensional indexing technique is implemented using the open source search engine *Lucene*, and compared with the built-in indexing technique available in the Lucene search engine, and also with the Terrier platform (available for the benchmarking of information retrieval systems) and other the most related indexing techniques. Evaluation results of memory usage and time complexity analysis, beside the experimental evaluations efficiency and effectiveness measures show that the proposed multi-dimensional indexing technique significantly improves both *efficiency* and *effectiveness* for a medical image search engine.

Keywords Indexing · Information retrieval · Medical images · Vertical fragmentation · Text-based retrieval · Features-usage correlation · Association pattern discovery · Query expansion

1 Introduction

Medical imaging is one of the best methods for medical diagnostic tasks and even interventional procedures and is growing in number and use cases. As an example, chest CT-scan is more accurate for detection of COVID-19 rather than the PCR test [1, 2]. As the number of medical images increases dramatically (such in the COVID-19 pandemic), the need for tools to search the information needed for clinical or research activities while considering users preferences (relevancy, efficiency, etc.) would be more important.

Search engines as the tools for performing the *information retrieval* (IR) process, either general or vertical domain-specific search engines (e.g., medical or healthcare), are

✉ AliAsghar Safaei
aa.safaei@modares.ac.ir

[1] Department of Medical Informatics, Faculty of Medical Sciences, Tarbiat Modares University, Tehran, Iran

empowered recently (in various directions such as *beyond the text IR*, *semantic web IR*, and use of data mining and machine learning techniques) [3].

Essentially, for evaluation of information retrieval systems, these two categories of metrics should be considered: *effectiveness* and *efficiency*:

a. *Effectiveness*: This criterion is used to measure the correctness and accuracy of retrieval. The key criterion for determining the quality of the information retrieval process is relevance. The relevance shows the rate of correctness in the retrieval. To measure relevance, measurements are introduced, which usually measures the relevance of a set of documents and queries. These criteria (metrics) are precision and recall.

The *precision* (p) is the fraction of the retrieved documents that are related to a query and provides a degree of soundness for the system. Precision does not care about the total number of documents considered relevant by the information retrieval system. This aspect is defined by the *recall* criteria (R), which is defined as the fraction of the truly relevant documents that are properly retrieved, and therefore regarded as a measure of the completeness of the system. Generating a fast, but inefficient response often does not seek user satisfaction, and certainly, the ultimate goal of data retrieval is to satisfy user satisfaction [4].

b. *Efficiency*: The efficiency criterion evaluates the use of resources (by the information retrieval system). Efficiency includes performance (e.g., processing speed or *response time*) and also system resource utilization. As information resources increase, the need to reduce retrieval time increases rapidly. The translation of the user's information needs is not easy. For this reason, they must find solutions that reduce the search time and thus improve the matching step. Increasing performance and overcoming uncertainties during the extraction and translation of information in documents and queries.

Indexing is one of the most important parts of search engines' procedures in which the documents selected by the *crawler* are processed and analyzed (by *tokenizing*, elimination of *stop words*, *stemming*, term *weighting*, and so on) to find for each document some keywords that can describe the document in a relatively unique manner (a.k.a. *index*). Of course, at the final step of this procedure, indexes of different document files are managed as *inverted file* in which all of the indexes are listed once indication to which documents each index is referring to Ceri et al. [4].

Indexing has very serious impacts on effectiveness and efficiency of the search engine since it contains metadata about the documents content (to use the relevant ones to the users query), and to retrieve documents faster.

Although, essentially, an index entry consists of the two main parts of *value* and *pointer*, but some more details could be used in index data structure (e.g., weight of the term in each of the documents *i.e.,* $t_f id_f$). So, to design a proper indexing technique, both the *index data structure* and the required *operations* (e.g., *create*, *update*, *delete*, *fetch*.) must be designed properly.

For indexing of image documents, generally there are two methods: *text-based* and *content-based* methods [5]. In text-based methods, indexes are expressed with the use of text descriptor (i.e., features of the image) and annotations. Searching for images in these systems is based on the text [6]; while in the content-based image retrieval, the image is retrieved based on features such as color, texture, shape and the like, which are extracted from the image's content, itself. So far, various systems for image retrieval have been developed using both methods (text-based or content-based) or by combining two methods; an initial classification of methods for medical images retrieval is shown in Fig. 1 [5].

Although, many of the research publications have focused on the content-based for medical image retrieval (e.g., [6–8]), but text-based medical image indexing techniques also have the following reasons to be used and emerging recently [9–11]:

- Computation overheads required for image content processing and analyzing can be reduced by annotating medical images and using such annotations as its features.
- One drawback of content-based indexing and weighting the documents based on the frequency of term or pattern is spam misuse in SEO (search engine optimization) topic (a technique wherein a word or pattern is repeated hundreds of times on a page in order to increase the frequency and propel the page higher in the listings). Off-the page retrieval (such as the *link analysis* or *click-through*

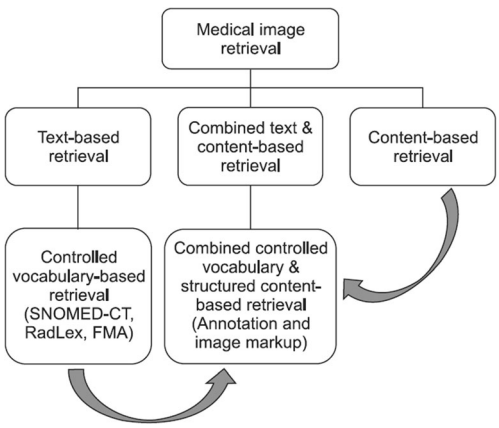

Fig. 1 Medical images retrieval methods classification [5]

measurement) or text-based techniques can be a good solution.
- Recent data mining and machine learning techniques and tools can be used to effectively and efficiently extract the features of the images' content

Considering the advantages and disadvantages of these two approaches, in order to increase the precision of retrieving medical images, the semantic retrieval of medical images, auto-annotation of images, and the use of a variety of methods for extracting content (color, shape, and texture) have been exploited.

In the study of Ayadi et.al. [11], a modality feature-based re-ranking model is proposed for medical image retrieval based on medical image–dependent features. These features are manually selected by a medical expert from imaging modalities (e.g., image modality and image scale) and medical terminology. The motivation is the large influence of image modality in medical image retrieval and they evaluate their approach via a series of experiments on the medical ImageCLEF data sets.

Galshetwar et.al. proposed a multi-dimensional multi-directional method for biomedical image retrieval taking into consideration the fact that, biomedical images have dominant spatial information. So, it encodes relationship of neighbor pixels in adjacent planes of a multi-dimensional image, in three stages; first of all, five sub images are formed by traversing in five different directions on three planes of a multi-dimensional image, then directional masks are applied on each sub image to find directional edges of the image, and finally, maximum edge patterns are found based on the directions of the directional edges. Relationship of neighbor pixels with center pixel is encoded by standard local binary patterns and local the relationship between adjacent pixels surrounding the center pixel is encoded by mesh patterns [12].

Tseng et al. [13] presented the multi-dimensional indexing structure called D-tree for access to business intelligence information. From a multi-dimensional point of view, point locations (objects) are important information for managing spatial data that should be well documented. The first two categories have similar features in representing spatial data. But for the management and processing of documents, words are the most important objects. But their location in a document depends on the context in which it is hardly possible to record with current language perceptual technologies. For online analytical processing, the traditional index structures for spatial data indexing may be very complex and inappropriate for text documents. Therefore, indexing structures for textual documents should be re-examined.

Image retrieval systems that have been used in the field of medicine are Automatic Search and Selection Engine with Retrieval Tools (ASSERT), CasImage, medGIFT, VisMed, BRISC, IRMA, the second National Health and Nutrition Examination (NHANES II), and FSSEM. Also, there are seven online medical image retrieval systems: figure search, BioText, GoldMiner, Yale Image Finder, Yottalook, IRMA, and iMedline that belongs to NLM (Open-i, Open Access Biomedical Search Engine) [14, 15].

Böhm et al. [16] implemented an indexing structure on a commercial relational database system, which showed that this could easily be done for a large class of multi-dimensional index structures. To prove this, they implemented an X-tree on Oracle 8 and ran several experiments on large databases. The upgraded performance is very high compared to the continuous scan of the database.

In the study of Safaei and Habibi-Asl [15], a multi-dimensional indexing techniques is proposed for medical images retrieval in which set of standard features of medical images (e.g., the DICOM format) are partitioned through a relational DBs normalization-like approach. Then, a hierarchical structure of such partitioned features is constructed as a multi-dimensional index structure and used for future search and retrieval of indexed medical images.

In order to clarify the differences of the proposed technique with the abovementioned works, it should be noted that although in the study of Ayadi et.al. [11], a feature-based medical image retrieval is issued, there is some major differences with the proposed text-based (feature-based) multi-dimensional indexing technique; first of all, it is re-ranking model but not an indexing technique (no index data structure to construct, store and using for retrieval), features are manually determined by medical human experts (not standard features such as in the DICOM). There are many other researches working on getting the relevance feedback through medical social networks for example [17]. Of course, Ayadi et. al. have also used the ImageCLEF data set for their experiments (it has been used it for a part of data set in this research).

Also, the multi-dimensional multi-directional mask maximum edge pattern medical image retrieval approach presented in Galshetwar et al. [12] in fact analyzes the content of medical images in three levels of dimensions, and also in multi direction (i.e., a multi-dimensional indexing of medical images based on their content pixels' patterns and in different directions).

The multi-dimensional indexing technique proposed in [15] in which *Normalization* of relational database is used for partitioning the set of feature for medical images is another similar and recent related work that would be compared and discussed.

The rest of the paper is organized as follows: the proposed text-based multi-dimensional indexing technique including the proper data structure and the required operations are presented in section 2. Complete evaluation of the proposed indexing technique, both analytical and experimental evaluation, is provided in section 3. Finally, the paper is concluded in section 4.

2 The proposed medical image retrieval

The proposed indexing technique is a text-based, multi-dimensional indexing in which correlation of the features' usage (according to the users' queries) is considered for the construction of the index structure.

In fact, assuming that each medical image has some extracted features (e.g., based on the DICOM standard), correlation of the features are discovered by performing data mining techniques (e.g., *association pattern discovery)*, on the history of users' queries as a data set. Then, based on the correlation of the features of the medical images, set of the all features is fragmented into subsets (using method like the vertical fragmentation of the tables in disturbing of relational DBs). After that, each of these subsets of the features turn into a hierarchy of the features (by applying a hierarchical clustering algorithm on that subset), subsequently all of these distinct hierarchies together make a multi-dimensional structure of the features of medical images, which is in fact the text-based (feature-based) multi-dimensional index structure.

Constructing and using such text-based multi-dimensional index structure via its specific required operations, medical image retrieval process would be improved in the underlying medical image search engine. Generally, an indexing technique is to provide a logical representation of documents in order to optimize the retrieval process. The proposed indexing technique is designed such that can improve retrieval of medical images in a medical image search engine in terms of its *effectiveness* and *efficiency*. Considering correlation of the features of the image would semantically improve precision (effectiveness) of the retrieval process, while managing them together in one dimension would try to optimize (i.e., minimize) the resources to have a better efficiency.

In order to describe the proposed text-based multi-dimensional indexing technique (its data structure and operations), it is necessary to provide a method for constructing the data structure; first, the text-based multi-dimensioning of the features in the creation of the data structure is explained. Then, the creation of the multi-dimensional structure is presented and after that, the main operations designed for this data structure will be explained.

Definition1: Multi-dimensional indexing technique

The multi-dimensional indexing technique, designed to facilitate information retrieval, consists of two basic pillars: (1) a multi-dimensional *data structure* for storing indexes; and 2) a *set of operations* for working with it, such as insertion, deletion, and search (query processing, matching, and representation).

Using the concept of *multi-dimensional* is to look at a document from all aspects. Previously, the search for documents was always done in one-dimensional form. In that way, for each index term assigned to a document had a referral at the time of retrieval, which meant that not all aspects of a document were considered and this would reduce the precision in retrieval; because it might have retrieved documents by simply having an index word in the query, without considering the other important aspects of those documents.

In the proposed multi-dimensional indexing, it is assumed that all indexes assigned to a document are properly retrieved in a multi-dimensional structure and unrelated images are not retrieved.

The 'S' medical document collection has 'M' descriptive features as M: $\{m_1, m_2, ..., m_k\}$, which is included in the dictionary that is created as explained in the introduction of this section. In the proposed data structure, the features are divided into "n" dimensions: D: $\{d_1, d_2, ..., d_n\}$. In this divide, $k \leq n$. If $k = n$, each dimension has one descriptive attribute, and it will not be any different from the one-dimensional indexing. Previously it was explained that one-dimensional indexing is not suitable for efficiency and effectiveness and reduces the speed and precision of retrieval.

Definition 2: A n-dimensional structure

"A polygon document DM = (S, $(A_1, A_2, ..., A_n)$) where S is the document set defined in n dimension $(D_1, D_2, ..., D_n)$."

Such a structure optimizes time and precision in the retrieval system by creating various dimensions that can be searched in the retrieval of a particular type of document, such as medical images. The only rational assessment is a statistical analysis of empirical behavior of different techniques and their response to real performances on real systems and on real issues [18].

2.1 2-1. Index construction (create)

In order to design a text-based multi-dimensional index based on the features-usage correlation, it is first necessary to compile the features associated with the medical image. These features will be used as content candidates in creating the index. A medical image with a DICOM format has some data. The contents of the DICOM header, each with the following features, are candidates for being in a multi-dimensional structure. DICOM is used in radiology, mammography, cardiology, radiotherapy, cancer, ophthalmology, dentistry, pathology, surgery, veterinary medicine, neurology, and pneumonia. The data assigned to each image on the DICOM header describes that image, which can be used to retrieve the image.

The following algorithm is used in the proposed indexing technique to extract features from the DICOM header.

```
1  Sequence sq = file.GetJointDataSets().
GetJointSubsequences();
2  string tag = string.Empty;
3  string description = string.Empty;
```

```
4   foreach (DataElement element in sq)   {
5     tag = element.Tag.ToString ();
6       description = element.VR.Tag.GetDictionaryEntry ().
Description;
7     Console.WriteLine (tag + " " + description);   }
```

The DICOM standard is the standard used for managing, storing and sending information specifically for medical images. Some of the images that will be used in the proposed system for retrieving the medical images will be images stored in the hospital and with the DICOM standard. The DICOM standard provides information on the description of all items, such as the name of the patient, the time of birth, age, sex, weight, smoking status and type of image.

The DICOM header has many features: (1) they are not all used to retrieve images; (2) many of these features are not filled by the radiology and hospital staff. According to research on the retrieval behavior of the group using these images [19, 20], at least two, or at best, three of the following axes were found in specific queries:

1. Anatomy area
2. Image type
3. Pathology

As a result, these three items were selected as the main features of retrieval. Other features such as age, gender, side (right or left), weight, and type of device were added to the DICOM header.

2.1.1 Step 1: Assigning features to each of dimensions

After the special features of medical images are collected, it is time to choose the appropriate features. For a more precise explanation of the choice of features, it is necessary to state that an attribute has two features of type and value. For example, the various types of features that are presented in the case study of medical images, along with their values, can be found in Table 1.

In this step, the proper features are chosen for each dimension. Features that can increase recall (relevant retrieved ratio to existing relevant) and precision (relevant retrieved ratio to total retrieved) in the multi-dimensional structure. It is assumed that M features have been identified in the collecting of features. Features that are in separate dimensions are subsets of the M set. These dimensions should have the following basic features:

- Covering: Includes all possible queries. Including all potential probable queries, by taking into account all applications and, for example, taking into account queries raised in an image retrieval system of a hospital or a medical search engine.
- Non-overlapping: The selected features are complementary and do not overlap.

In other words, the subsets of the selected features (whose union are null and their intersection makes the entire collection) divide the entire set of features should be useful for all users of medical search engines, including diagnosis, treatment, education, and research.

To do such a division of medical features, using features application, the vertical fragmentation will be used. Vertical fragmentation is done by partitioning the database tables containing descriptor features, in parts that are formed by using the affinity of the features. The criterion for affinity of features is their application. Finding the frequency of document descriptors application can be done in two ways:

1. Considering all types of queries that users are submitting to the retrieval system. For example, if the list or user query logs are available from the image retrieval system, this method is used. By analyzing queries, you can identify the features that are being asked together.
2. Based on the analysis of descriptor features from the collection of images available. For example, in a hospital, depending on the amount of the different types of medical images stored with the DICOM standard header, one can find out what the needs of patients and medical personnel are for various uses, including diagnosis, monitoring, etc. in that hospital. As a result, if the DICOM descriptors and their data are extracted from a collection of hospital images taken during a given period from the patient, then proper categorization take place on the descriptors of the images; descriptor collections can be obtained.

Table 1 Differences in the type and value of features

Type	Value
Image modality	MRI, CT, X-ray, radiology
Anatomy area	Heart, nervous system, stomach, spine
Injury	Inflammation, clotting, vascular rupture, Fractures
Sickness	Diabetes, kidney failure, arthritis, Parkinson, Alzheimer's

Definition1: *Vertical fragmentation*

Vertical fragmentation in a relation A creates parts A_1, A_2, ..., A_r, each of which has a subset of features A, which are categorized according to the number of uses of the values of specific features (used in queries). The purpose of the vertical fragmentation, dividing a relationship into a set of small relationships, so that many user applications can only be executed on one piece. In this case, an "optimal" piece creates a plot of fragmentation that minimizes the runtime for the user's application on which The parts are executed [21].

Vertical fragmentation has been investigated in the area of centralized and distributed database systems. In centralized systems, it is a design tool that allows user queries to work with smaller relationships, thus giving access to a small number of pages. It has also been suggested that very "active" sub-relationships can be identified and, in cases where there is a hierarchy of memory, it can be embedded into the subsystem with faster memory [21].

This concept of vertical fragmentation defined in the two paragraphs above is applicable to the definition of optimal dimensions. If a relation A is a collection of descriptor features of the retrieved documents, vertical fragmentation can divide the descriptor features set into smaller sets in order to create optimal dimensions that minimize runtime execution.

The information requirements are described below. The most important information needed as input for vertical fragmentation is the application. Because vertical fragmentation puts those features into a piece that are accessed together, it is necessary to define a different criterion to define the concept of "being together." This criterion is "affinity" of features that indicates how much features are related.

The important information regards the applications is the frequency of access to them. But as already mentioned, at the beginning of creating a fragmentation, in the absence of access to the application, it is logically possible to refer to the frequency of features in the list of writings. If $Q = \{q_1, q_2, ..., q_q\}$ is a set of references (applications) to the relation A (a_1, a_2, ..., a_n). For each qi reference and any Aj attribute, an attribute usage value is used which is referred to as use (qi, Aj) and is defined as follows:

$$\text{Use}\left(q_i.A_j\right) = \begin{cases} 1 & \text{if qi uses } Aj \\ 0 & \text{otherwise} \end{cases}$$

In this regard, if the A_j attribute is referenced by q_i, the value of the relationship is equal to 1 and otherwise 0.

Use vectors (q_i, 0) for each application, if the designer knows the queries to be executed on the database, it is easy to define.

Example: By having the following relation, queries have been raised:

PROJ (P NO, P NAME, BUDGET, LOC)

By having the identification number of an image, find the type of sickness it describes:

q_1 = *SELECT SICK FROM PHOTO WHERE PNO=Value*

Find the types and anatomy described in all images:

q_2 = *SELECT PNAME, ANATOMY FROM PHOTO*

Find types of images that have the considered sickness:

q_3 = *SELECT PNAME FROM PHOTO WHERE SICK=Value*

Find all images that have a specific sickness in various anatomical areas of the body:

q_4 = *SELECT ANATOMY FROM PHOTO WHERE SICK = Value*

In summary:

A_1 = *P NO*, A_2 = *PNAME*, A_3 = *ANATOMY*, A_4 = *SICK*

And the matrix for using the features is as following Fig. 2:

The values of the use of features don't have suitable generality to form the basis of separation and fragmentation. Because these values do not specify the frequency of applications, frequency criteria can include the definition of the affinity criteria of features af f (A_i, A_j), which measures the relationship between the two features of a relationship based on their usage [21].

The affinity criterion of the features between the two features A_i and A_j of the relation R (A_1, A_2, ..., A_n) is defined for the set of applications $Q = \{q_1, q_2, ..., q_q\}$ as follows:

$$aff\left(A_i, A_j\right) = \sum_{k|use(q_k,A_i)=1 \wedge use\left(q_k,A_j\right)=1} \sum_{\forall S_l} ref_l(q_k) acc_l(q_k) \quad (1)$$

where re $f_l(q_k)$ is the number of accesses to the features (Ai, Aj) for each execution of the q_k application in the search engine S_l and acc_l (q_k) is the predefined access frequency rate criterion.

The result of this calculation is an n × n matrix, each element of which is one of the criteria defined above. This matrix is called the features affinity matrix of (AA).

Example: In the following example it is considered that for simplicity, $ref_l(q_k) = 1$ is considered for all q_k and S_ls. If the frequency of applications is as follows:

$acc_1(q_1) = 15$ $acc_2(q_1) = 20$ $acc_3(q_1) = 10$

$acc_1(q_2) = 5$ $acc_2(q_2) = 0$ $acc_3(q_2) = 0$

$acc_1(q_3) = 25$ $acc_2(q_3) = 25$ $acc_3(q_3) = 25$

$acc_1(q_4) = 3$ $acc_2(q_4) = 0$ $acc_3(q_4) = 0$

	A_1	A_2	A_3	A_4
q_1	1	0	1	0
q_2	0	1	1	0
q_3	0	1	0	1
q_4	0	0	1	1

Fig. 2 An example of an features usage matrix

The affinity criterion between the A_1 and A_3 features is calculated as follows:

$$aff(A_l, A_3) = \sum_{k=1}^{1}\sum_{l=1}^{3} acc_l(q_k) = acc_1(q_1) + acc_2(q_l) + acc_3(q_i) = 45 \quad (2)$$

Since the only query (application) that accesses both features is q_1, the response is calculated as follows. The matrix of the affinity of the features is shown in Fig. 3. Diameter values are not computed because they are meaningless [21].

Features affinity matrix is used to guide fragmentation. This process first involves the clustering of high-affinity features together and then the separation of the relationship on this basis [21]. The result of the fragmentation resulting from this process is the creation of subsets of the large M set of medical images features and each subset forms one dimension. The Bond Energy Algorithm (BEA) is a fundamental work in the design of a vertical fragmentation algorithm for grouping features of a relationship based on the affinity values of features in AA.

2.1.2 Computation of the affinity matrix (correlation of the features)

Someone may ask the question, how the affinity matrix can be computed or generated to be used as the input of the described vertical fragmentation step.

Different approaches are used for affinity computation, ranging from mathematical computation (such as in [22]), or feedback through medical social network (e.g., in [17, 23]), to using data mining techniques such as the *Random forest* and *Gaussian process regression* algorithms [24].

The most proper approach for computation of the affinity matrix is the *Association Rule Mining* [25]. Association rule (pattern) mining (discovery), provides interestingness relationship between data attributes (i.e., features) and has application in market basket analysis Web usage mining, intrusion detection, continuous production, and bioinformatics.

But, classic *Boolean Association* pattern discovery would not be suitable for our case, and *Quantitative Association* pattern discovery [26] must be used, since we need the numerical value of the pairwise interestingness of features, to compute the affinity matrix's entries.

	A_1	A_2	A_3	A_4
A_1	-	0	45	0
A_2	0	-	5	75
A_3	45	5	-	3
A_4	0	75	3	-

Fig. 3 Features affinity matrix

So, the quantitate association rule mining approach presented in [26] has been applied for computation of entries of the affinity matrix of the images' features, to be used for the vertical fragmentation step (described in section 2-1-1).

After computing the features affinity (correlation) matrix and using it to vertically fragment and partition the set of the features into subsets, each with the maximum affinity between its features, each of these subsets of features (as one *dimension*) will be reshaped into a *hierarchy* by applying a *hierarchical clustering algorithm* (described in the next section).

Note that, as stated before, the proposed indexing technique aims to improve retrieval of medical images in a medical image search engine in terms of *effectiveness* and *efficiency*. Considering correlation (i.e., pairwise affinity) of the features of images would semantically improves the precision (effectiveness) of the retrieval process, while managing them together in one dimension would try to optimize (i.e., minimize) the resources to have a better efficiency.

2.1.3 Step 2: Hierarchical clustering within each subset (dimension)

To create the right structure for each dimension, we need to examine different methods of processing and categorizing information so that, by creating the proper structure, it becomes possible to perform retrieval on features in an optimal manner in terms of time and memory usage, as well as in terms of retrieval evaluation criteria (precision and recall). Data processing is one of the most important issues in the information world. Clustering is one of the best ways to work with data. Clustering makes it possible to enter the data space and recognize its structure, so it is considered as one of the most ideal mechanisms for working with a huge data world.

The number of clustering methods currently used to analyze data is very high. The two common types of clustering methods are as follows [27]:

1. Hierarchical clustering (has two types of agglomerative and divisive)
2. Non-hierarchical clustering (includes five types of partitioning (*K-means*, *K-medoids*, Fuzzy c-means), density based (DBSCAN, OPTICS), based on the grid, network-based algorithms (STING, CLIQUE) and graph-based)

Non-hierarchical partitioning clustering algorithms are less costly in terms of computational time than hierarchical algorithms [27]. On the other hand, hierarchical algorithms provide more qualitative results than partitioning [28].

Hierarchical clustering algorithms are used to divide or merge a given dataset into a sequence of nested partitions. The hierarchy of these nested parts is of two types:

agglomerative or down-to-top or divisive or up-to-down. Usually, their results are displayed by a dendrogram tree.

The tree map or dendrogram (hierarchical tree diagram): The final output of both hierarchical agglomerative and divisive hierarchical methods is a dendrogram. A dendrogram is a 2-D diagram that can be plotted both vertically and horizontally. The results are the same in either case. To determine the number of clusters, a dendrogram can be cut at an appropriate point. In this diagram, what matters is height. As clusters formed at lower altitudes, clusters or observations are more similar to each other and vice versa.

To form the tree structure required for the proposed multi-dimensional indexing technique, the clusters that are created at each stage and displayed in the tree structure of the dendrogram are shown in Fig. 4 will be kept and will be used to create the tree structure in each dimension of the proposed data structure.

Comparison and selection of appropriate clustering The purpose of clustering is to find similar clusters of objects among input samples, but which clustering method is appropriate and which is not suitable is a controversial issue. It can be shown that there is no absolute criterion for the best clustering, but it depends on the issue and the user's point of view. By estimating or looking at the output, the user can determine how much the precision of data clustering is. However, there are various criteria for the goodness of a cluster that can guide the user to achieve a proper clustering. Some of these criteria are presented in later sections. One of the important issues in clustering is the selection of clusters. In some algorithms, the number of clusters is already specified, and in others, the algorithm decides itself to divide the data into how many clusters. An agglomerative hierarchical clustering that has a down-to-top operation shows a better performance for the data structure. Web clustering has become a topic for researchers in the field of information retrieval for many years [29]. Aggomolative hierarchical clustering is often used more than divisive in data retrieval [30].

The reason for choosing Aggomolative hierarchical clustering as the priority over non-hierarchical clustering algorithms in retrieval is as follows [29]:

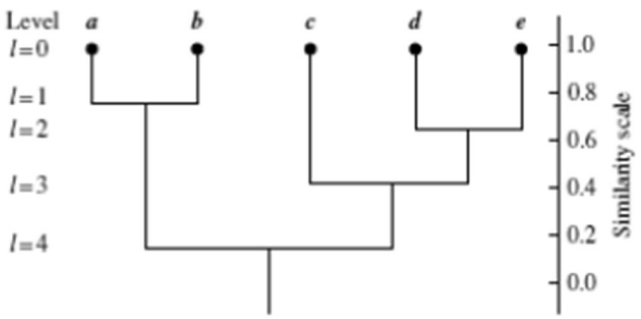

Fig. 4 Example of a dendrogram tree from agglomerative clustering

1. For non-hierarchical algorithms, the number of clusters as inputs is required. But getting this number is very hard. On the other hand, hierarchical clustering does not require this information.

2. Non-hierarchical clustering algorithms are uncertain and unstructured. While the agglomerative hierarchical clustering algorithm is certain, it returns a hierarchy that contains useful information.

3. By hierarchical data, the readability is more informative, but in a non-hierarchical algorithm it is necessary to study all the clusters for finding information [29].

The simple algorithm of agglomerative hierarchical clustering is as follows:*SIMPLEHAC (d1, . . ., dN)1* ***for n ← 1 to N2*** ***do for i ← 1 to N3*** ***do*** *C[n][i] ← SIM (dn, di)4 I[n] ← 1 (Considers active clusters)5 A ← [] (Maintains clusters as a sequence of merges)6* ***for*** *k ← 1* ***to*** *N − 17* ***do*** *< i, m > ← argmax* $_{\{<i,\ m>:\ i \neq m \wedge I\ [i]\ =\ 1 \wedge I\ [m]\ =\ 1\}}$ *C [i][m]8 A. APPEND(<i, m>) (Saves merge)9* ***for*** *j ← 1* ***to*** *N10* ***do*** *C [i][j] ← SIM (i, m, j)11 C [j][i] ← SIM (i, m, j)12 I [m] ← 0 (Inactivating the cluster)13* ***return*** *A*

First, the matrix N × N is calculated for the similarity "C." Then, the algorithm, N-1, executes the merging step for clusters that are very similar. In each replication, the two clusters that are very similar are merged, and the columns and rows of the cluster merged into "C" are updated. Clustering is stored as a list of mergers in "A." "I" shows which cluster is still active for merge. The function of SIM (i, m, j) calculates the similarity of the j cluster in the merge with clusters i and m. In some agglomerative hierarchical algorithms, SIM (i, m, j) is just a function of C [j] [i] and C [j] [m], [31].

Choosing the criterion of similarity or lack of similarity (distance) A criterion of similarity criteria is used to determine whether an instance of data belongs to a cluster or not. The function of many algorithms depends on choosing a good similarity criterion for its intended data set and changing the quality of the final results. Similarity criteria are chosen based on the application and type of algorithm.

The Rand index (RI) is often used to measure the cluster quality and is an agreed criterion between two set of objects: the first is the set that has been created in the clustering process and the other is defined by the external standard. While there are various clustering criteria, such as total square error, entropy, purity, jacquard, etc., the RI is probably the index that is most often used to check the accuracy of clustering [32]. Assume S = $\{o_1, o_2, ..., o_n\}$ is a set of 'n' elements and two divisions of S are compared. C = $\{c_1, c_2, ..., c_r\}$, which is a partition of "S" with "r" subset and G = $\{g_1, g_2, ..., g_s\}$ is a division of S with "s" subset, the RI index is defined as follows:

$$\mathrm{RI} = \frac{a+b}{a+b+c+d}$$

where in:

- a is the number of pairs of data in S that are in C in the same set and in G in the same set.
- b is the number of pairs of data of S that are in C in different sets and in G in different sets.
- c is the number of pairs of data of S that are in C in the same set and in G in different sets.
- d is the number of pairs of data of S that are in C in different sets and in G in the same sets.

In the study [32], similarity criteria were compared for numerical data clustering in distance-based algorithms and benchmarked using 15 sets of data. The accuracy of similarity criteria is calculated using the Rand index and is discussed for the best similarity criterion for a set of small and large dimensional data and for four well-known distant-based algorithms. The result shows that the 'average distance' is among the highest accuracy in the criteria for clustering algorithms. Based on the results of the study Shirkhorshidi et al. [32], Pearson coefficients are generally not recommended for a low-dimensional data set. It also does not work with center-based algorithms (algorithms that include central data, representing each cluster that does not necessarily belong to the dataset). This criterion is recommended for a large dimensional data set using hierarchical approaches.

$$Pearson(x,y) = \frac{\sum_{i=1}^{n}(x_i-\mu_x)\left(y_i-\mu_y\right)}{\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}}$$

According to the results of the studies mentioned, Pearson's coefficient is used as the appropriate similarity criterion for selected clustering, i.e., agglomerative hierarchical clustering that is used to construct multi-dimensional data structure's dimensions.

2.2 The Multi-dimensional index data structure

In this section, comparisons were made with the study on various categorization, classification and clustering methods to create the appropriate data structure for each dimension of the data structure of the proposed indexing technique. The goal is to have a data structure that brings together the most closely related data. The time and space used for this structure should be as optimal as possible. Finally, a structured and stable agglomerative hierarchical clustering was selected to create a tree structure in each dimension.

Hierarchical clustering displays the clustering result in which similar data are put together in a structured way in the dendrogram tree. Since agglomerative hierarchical clustering is used more in information retrieval 37] and is not sensitive to pertinent data, this clustering was selected. The reasons for this selection are given in Section 2-1-2-1. The algorithm suitable for this clustering, due to the low time and space complexity, was chosen for the complete linkage algorithm.

The similarity criterion that will is used in this algorithm, according to the practical results of the different distance criterion, was determined as the Pearson coefficient. Because the practical results indicated by the RI showed that the Pearson coefficient exhibits a better hierarchical clustering than other results.

By clustering for the normalization approach, as in Fig. 5, a tree structure is constructed for each dimension of the proposed multi-dimensional data structure. The final result is the creation of a forest. The final structure for the proposed multi-dimensional construction is the structure that connects these trees and creates a multi-dimensional. To connect these dimensions, the use of a circular or rotational link list is proposed and discussed below:

In the proposed structure, a root node collects the root of all other dimension trees (Summit) and creates a meta-tree. The root of each tree is the representative of that tree, which is the structure of each dimension. Obviously, to create a Summit structure, firstly trees of each dimension must be created to form the structure with the aggregation of the roots of each dimension. For the proposed data structure, the place where the dimensional connection is made is called the Summit. The Summit in the proposed structure contains the roots of the trees forming each dimension. In order to be able to access the root nodes that represent the dimensions in a dynamic and fast way, they are structured into a double link circular link list at the Summit. The reason for using a rotating or circular link list is quick access to root nodes at the Summit.

Using arrays is a method of storing such data that has some disadvantages. For example, adding and removing elements in an array is relatively costly. Moreover, since each array usually occupies a block of memory space, the number of stored elements in an array is limited to the size of the array, and the array size cannot be increased when it is necessary to store more elements than the size of the array. For this reason, arrays are called compact or dense lists. In addition, arrays are also called the static data structure.

Another way to save a list in memory is to put each element in a node, which contains the information fields and the next node's address in the list. In this way, it is not necessary to occupy successive elements in the list of adjacent spaces in memory. This makes it easy to add and remove list elements. This is known as building a link list.

The link list is a dynamic data structure. The number of nodes in the list is constantly changed by inserting and deleting the elements. The dynamic nature of the list is controversial with the static nature of the array whose length remains constant. One of the reasons for selecting the link list to implement the Summit in the data structure of the proposed

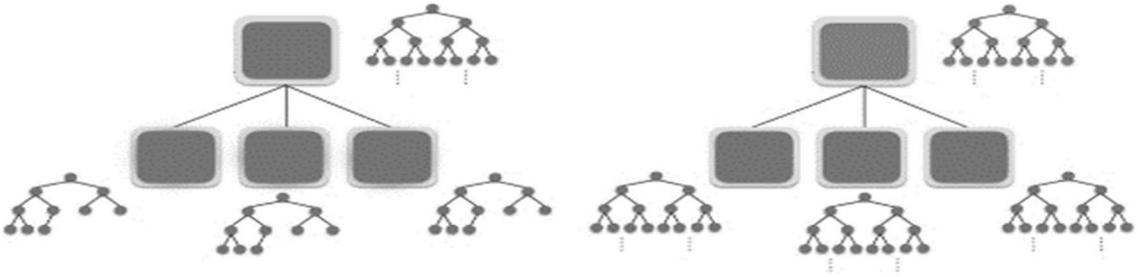

Fig. 5 Outputs resulted from hierarchical clustering: dimensions in a forest structure

technique is its dynamic structure. Another reason is the optimal use of storage space using pointers and has a positive effect on the system's efficiency, which is one of the goals of designing a multi-dimensional indexing technique. In addition, the cost of insertion and removal in the array is very costly.

Since each node in link list has the next node address, it is not necessary to put the list elements in memory beside each other. Because each node defines its next element, to access the link list elements, an external pointer refers to the first node in the list. In the Summit data structure, the pointer is stored at the main root. This pointer contains the address of the first node of the list. The first node in the list points to the first dimension that has a high priority, and scrolling always begins with this entry.

The circular link list is similar to the one-way list, with the difference that the last node's address field, instead of referring to NULL, refers to the list's first node. In the one-way list, we must always have the first node of the list. But in the circular link list, you can access all the nodes with the address of each node you want. An example of a circular link list is shown in Fig. 6.

2-2-1. Nodes The node is considered as the smallest unit of the data structure. After explaining the types of nodes, the space complexity of the proposed data structure is presented. The following section describes the operations and pseudo-code for these operations in the proposed data structure. The time complexity of each operation is given at the end of each operation.

- *Node*: The smallest unit is a data structure. In the data structure, which is constructed using an agglomerative hierarchical clustering in the form of a tree in each dimension, then the root nodes of the trees, by connecting to the root mother in the structure of the summit, form the proposed data structure. There are three types of nodes:

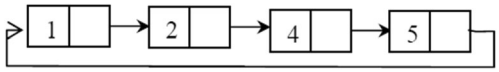

Fig. 6 An example of a circular link list

1. Root node
2. Middle node
3. Leaf node

Each node has two fields including data and pointer which in different types of leaf, root, and middle nodes, each field has different parts as shown in Fig. 7. First, all the fields that can be added to each part of a node are explained, and then, explaining each node, the reason for using or not using each part is explained.

The data part can include the following:

1) The frequency of any attribute
2) The frequency of the value of that attribute
3) The attribute
4) The value of the attribute itself
5) Data structure information
6) Information about each dimension

The pointer part can include the following:

1) pointer to the next node of the same level
2) pointer to the previous node of the same level
3) Pointer to a parent node
4) Pointer to a child node
5) A document pointer

Parts that are not generally used in the proposed data structure nodes include the frequency field of features, the pointer to the previous node, and the pointer to the parent node.

The frequency of features is used to create the data structure and is the input of the first step of dimensioning and also the second step in creating a hierarchical structure. Since the creation operation occurs only once and has no function in the main operation, i.e., search, also to optimize the efficiency of the space usage criterion of the data structure, it is not necessary to use it in the original data structure.

The pointer to the same level node is used in the link list section of the proposed data structure, where the root nodes of all dimensions are located. Because the dimension scrolling in the search operation is forward, it is not necessary to use the

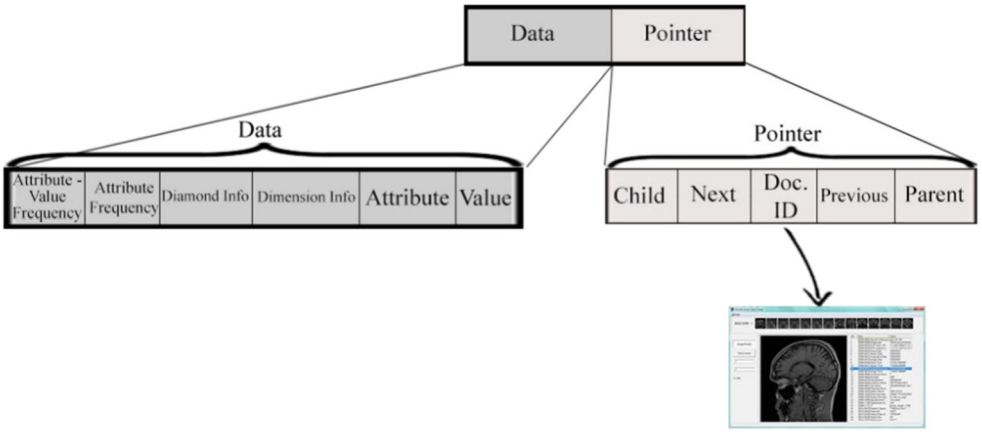

Fig. 7 View of potential parts in a general node

pointer to the previous node, and only the pointer to the next node is used. Not using the pointer to the parent has a similar reason. As the navigation in the data structure is from top to down, it is enough to have a pointer to the child. Not using parts that are not used in operations improves the efficiency criterion for memory usage. The following sections describe the parts that are used in the proposed data structure in detail.

Each node can be defined as a structure that has the fields of data and the address field. In the following definition of the node, the types of parts of each field in the general structure of a node in the proposed data structure are presented. The reason for not using some parts of the general structure in the definition of nodes is explained above.

```
1  Struct Node
       {
2          elementtype info {
3        Attribute
4        Value
5        Info
6        Info
                 };
7          Node * next {
8        Node → Child
9        Node → Doc ID
10       Node → Next                 };
           };
```

As shown in the pseudo-code above and in Fig. 8, each node has two general parts: the data part and the pointer part. The data part contains information about the entire structure of the data, information about each attribute, the attribute itself, and the attribute value. The pointer part contains the pointer to the child, the pointer to the image document and the pointer to the next node (in the root nodes of each dimension).

The content of these two parts is different in the root, middle, and leaf nodes.

Leaf node: The leaf node in the data part contains the attribute and its value. Maintaining the attribute type in each node is due to the multi-dimensionality of the data structure. Because in the case of one-dimensionality, there is no need to express the type of attribute. In the pointer, the leaf node points to the image document but does not point to the child. The pointer to the same level node is also not used because of the of the top-down scroll. An example of a leaf node is given in Fig. 9.

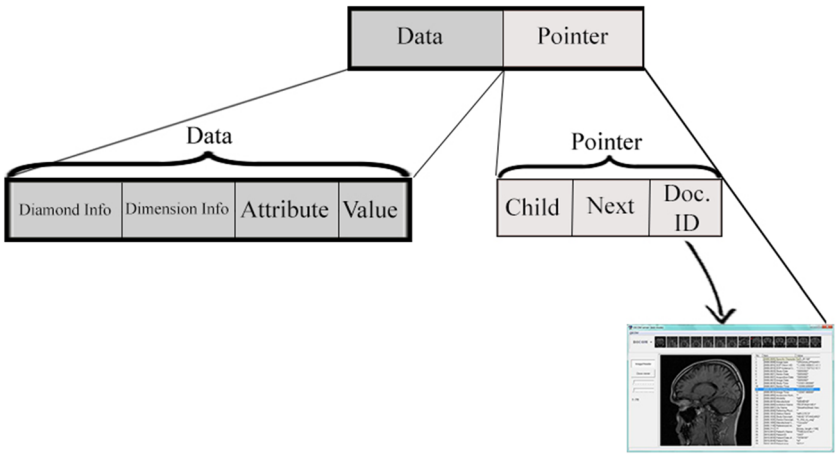

Fig. 8 The general structure of a proposed multi-dimensional indexing node

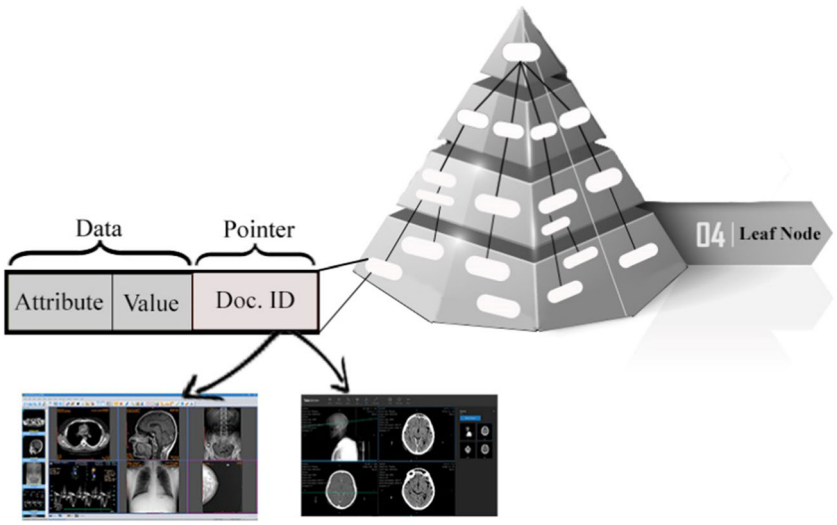

Fig. 9 A leaf node in the proposed multi-dimensional structure

Middle node: The middle node shown in Fig. 10 is the same as the leaf node in the data part containing the attribute and its value. In its pointer section, it does not refer to the same level node; because the navigation in this data structure is from top to down and there is no need to refer to the node at the same level. In the pointer, the middle node contains a pointer to the child and one to the image document. The direct pointing of the middle nodes to the document is to make data structure flexible for comprehensive queries. This will improve the recall and causes precision retrieval for any incomprehensive query. It also improves performance by boosting retrieval speed. For a clearer explanation of these reasons, consider the following example:

The S document set contains images with features A. A contains the attribute type and the attribute value. For example, for the type of attribute, you can name image modality or disease. If we consider the image modality as an attribute type, the attribute value for it will be MRI, CT, and X-ray. All features A are located on the middle and leaf nodes. A node in the data section may have only image modality attribute. If this attribute is located in the middle node and the user uses the keyword "all modalities" of images in their queries, all of these images will be retrieved more quickly with a pointer to the documents containing this attribute. The opposite of this process is that all the nodes that have modality in their attribute type part of the data part are separately found and retrieved, which is required a long time for it. As a result of the presence of a pointer to the document in the middle nodes, speed and recall will increase, especially for comprehensive queries.

Root node: In the proposed data structure, as shown in Fig. 11, there are two types of root nodes. Mother root node and the root node of each dimension. The Mother root node, which is located above the data structure and is the beginning point of scroll through its structure, contains data about its dimension in the data section and a pointer to the children in the pointer section. These children are root nodes in dimensions. The root node of each dimension, in the data section, contains the information for each dimension and in the pointer, contains the pointer to the child and the pointer to the next node of the same level.

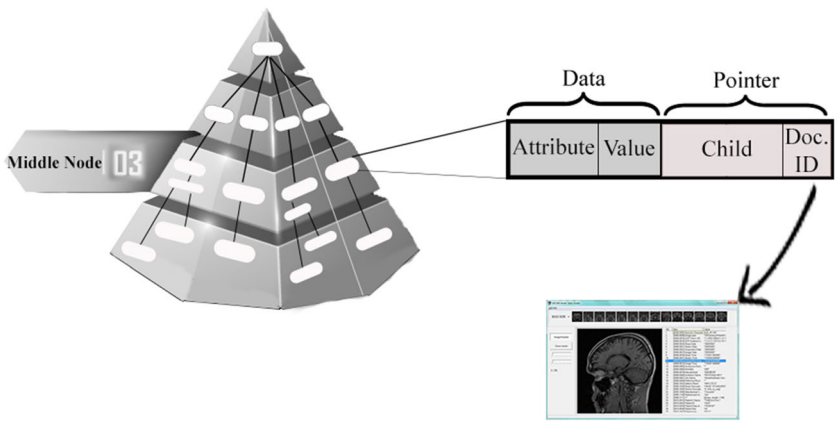

Fig. 10 middle node in the proposed multi-dimensional structure

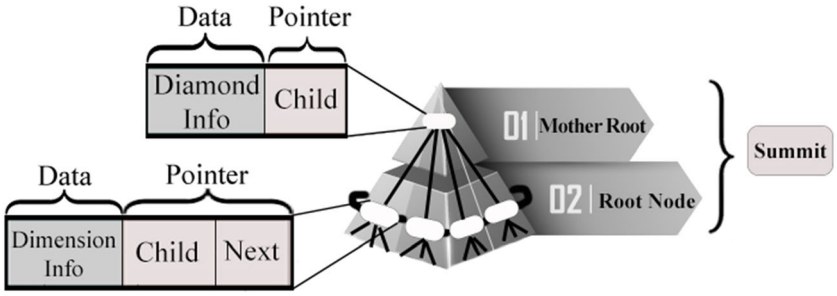

Fig. 11 The root node in the proposed multi-dimensional structure

Connection of the dimensions: In the proposed structure shown in Fig. 12, a root node collects the root of all other dimension trees (Summit) and creates a meta-tree. The root of each tree is the representative of the same tree, which is the structure of a dimension. Obviously, in order to create the Summit structure, we must first create trees of each dimension, to form the structure by aggregating the roots of each dimension. For the proposed data structure, we call the point where the dimensional connection is made, the Summit. The Summit in the proposed structure contains the roots of the trees forming each dimension. In order to be able to access the root nodes that represent the dimensions in a dynamic and fast way, they are structured in form of the circular link list at the Summit. The reason for using a rotating or circular link list is quick access to root nodes at the Summit.

The general form of the proposed multi-dimensional data structure is shown in Fig. 13. Due to the similarity of the structure created to proposed s.

2.3 *2-3*. Other operations and the time complexity analysis

2.3.1 ■*Create*

This operation is described in detail in Section 2-1. To create a multi-dimensional index, first of all, special features are derived from medical images. Then, these features are divided according to the normalization approach based on functional dependence. Next, the data structure is created. Agglomerative hierarchical clustering creates the data structure for each dimension. Then, by connecting the roots of each data structure, the Summit is formed, which is the dimensional connection point.

In the following pseud-ocode, the create operation is presented based on the previous sections:

Create index ()
{
1 Extract Features from Images;
2. Separate Features in Dimensions (
3 Normalization (with Functional Dependency))
4 Data Structure for each Dimension (Hierarchical Aggomolative Clustering)
5 Bind Dimensions in Summit;
}

2.3.2 ■*Search (fetch)*

Search operations are done in the following steps:

1) It starts with a query by the user. This query contains specific features or their values (for example, the image modality is a specific attribute and the CT image is a value). Scrolling the data structure is happening for the search.
2) Scrolling starts from the root.
3) Gets to circular link list.
4) Then runs parallel on the trees of the dimensions at the same time.
5) If reaches the specific attribute in the query, it returns the pointer to the document.

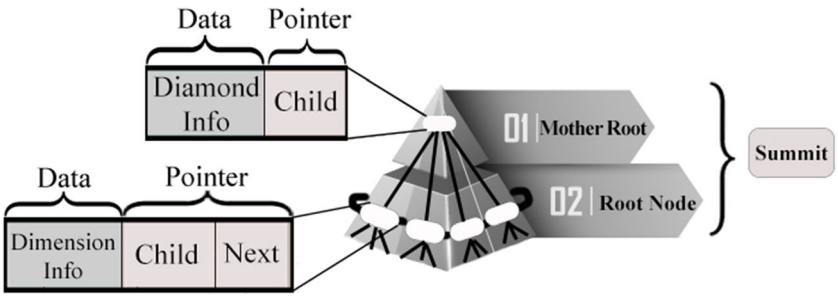

Fig. 12 The Summit structure in the proposed multi-dimensional structure

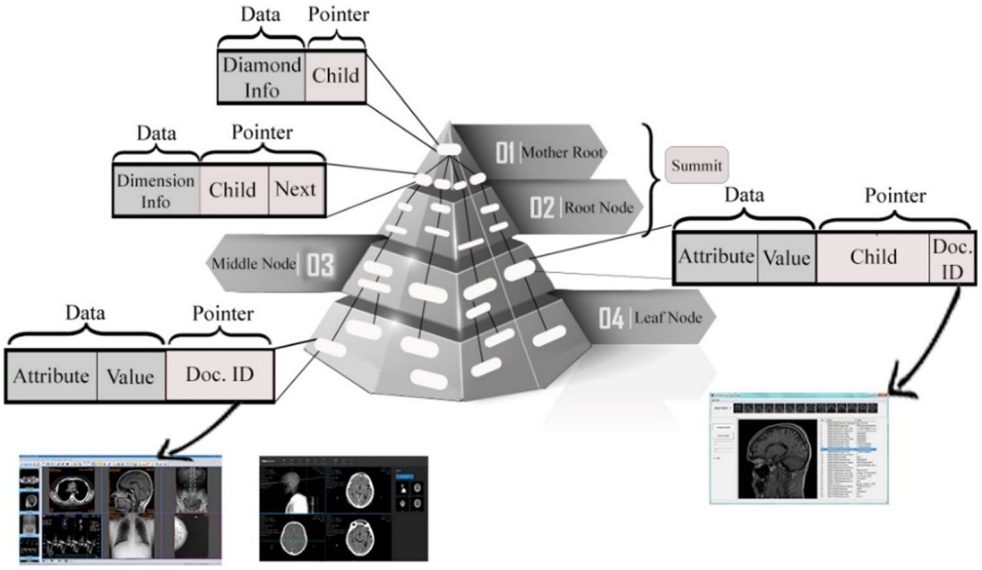

Fig. 13 indexing data structure

```
1 Search index (void Search_In_All_LinkedLists (List l, string search_val)
{
2  Position p = l- > next;
3      while (p! = NULL)
{
4     agg_tree my_tree = search_tree(p- > tree_node, search_val);
5    if (my_tree! = null) // The search term was found in this tree
6    print_preorder (my_tree);
7 else // That is, the search term in this tree is not found and should go to other trees
8       p = p- > next;
{
9 Return my_tree- > image
   }
```

To synchronize search in all trees, a multi-threaded system is implemented as follows:

Thread th = new Thread (() = > Search_In_All_LinkedLists (List l, "Query")

- Rating the retrieved results:

The search result is usually presented to the user as a list of images. These images are sorted by relevance to the user's query. Any document that has more words of the query will have more privileges. When a free text query in the form of a set of words is entered into the search engine interface without the use of any particular operator (such as Boolean operators), an acceptable scoring mechanism calculates the privilege that is counting number of the words in the query, the total amount of matching between each query word and the words in the document index privileges the documents in result [31].

After obtaining the features used in the user's query, the identifier of the document that has any attribute is recalled, and the result of comparing the document identifiers will be the document that has all the features to be queried. These image documents have a higher rating and are presented to the user at the beginning of the list.

Example:

Query: The image has the features $\{a_i, ..., a_2, a_1\}$

Navigating the structure and finding features,

Finding the identifier of the documents that contain these features from the written list

Group by for DocIDs

If Count DocID is higher, will be retrieved as the first one.

Select (Attribute, DocID)

From index_data_structure

Where (Attribute = a_1 OR a_2 OR a_3 … OR a_i)

GroupBy DocID

Order by (Count DocID) DESC

■Insert

To insert a new image, you need to do the following:

The features associated with the new image are found in the data structure and for this purpose, the navigation operation (same as search) is performed.

Then, from those features, a pointer is created to the new document.

Table 2 The time complexity of the main operations of the index data structure

Operation	Time complexity
Creation	$O(n + n^2)$
Search	O(n /d)
Insert	O(n /d)
Delete	O(n /d)

Insert index (Image (Doc ID), Features){ {
1 Search index_data_structure (Features);
2 For Each (Nod [D] = Attribute)3 Nod [P] → Doc ID;
4 Next
}
}

■**Delete**

1) Delete operation, such as insert, requires scrolling (the same search).

2) Finding the features of the image, the pointer of that features to the image should be removed.Delete index (Image (Doc ID), Features){ {1 Search index_data_structure (Features);2 For Each (Nod [D] = Attribute)3 Nod [P] → NULL;4 Next }}

Accordingly, the time complexity of the main operations of the data structure are shown in Table 2.

Some of the most important scientific contribution and novelty of the proposed indexing technique are as follows:

A text-based multi-dimensional index structure for medical images instead of one or multiple distinct indexes, in order to provide an integrated multi-aspect perspective (e.g., tissue, modality and format of the image, sickness or trauma) to a medical image

Text-based (image's features) instead of content-based indexing that relax using of image processing techniques and its overheads and accuracy issues

considering affinity of the features to each other (based on their correlation according to users' queries statistics) to cluster them (so, is users-centric indexing)

Improvement of effectiveness (i.e., precision and recall) as well as efficiency (i.e., response time and memory usage) which is illustrated via analyzing and experimental evaluation in the next section

3 Evaluation

3.1 3-1. Experimental evaluation setup

Each of the requirements for evaluating the multi-dimensional indexing technique will be explained in more detail in this section. This evaluation is performed on a system with the following specifications and in the Windows operating system: Intel Core i5-4200M @ 2.5.0 GHz RAM: 6 GB

The used computer has an Intel Core i5-4200M processor with a frequency of 2.5.0 GHz and 6 GB of RAM.

To evaluate the multi-dimensional indexing technique, this indexing technique was implemented using the Lucene open source search engine. As the proposed indexing technique replaced the Lucene's indexing technique. The implementation of the proposed indexing technique begins by obtaining image information. Due to the fact that the format of the images is of DICOM type, at this stage, the text information of the entire image is extracted. From each image, information is extracted in the following order:

1.Image ID: Each image is given a unique number. This number is used in vertical fragmentation.

2.Disease area: like chest, head and neck, pancreas and ...

3.Photo type (modality): Like CR, CT, MRI and ...

4.Type of disease: divided into benign and malignant tumors.

5.Age of the patient: for example, 58y is 58 years.

6.The patient's weight: for example, 78 kg.

7.Gender of the patient: It is divided into two groups of men and women (male and female).

8.Other information: Depending on the area of the image, it is different:

PET Prostate: The location of the imaging is divided into two categories: hfs and ffs.

Mammography: Divided into two groups to the right and left.

Breast: Divided into two groups to the right and left.

4D Lung: The location of the shooting is divided into two categories: hfs and ffs.

For other areas, this part is empty.

3.1.1 3-1-1. Data set

For this study, in order to approximate the result of its proposed approach for retrieving medical images to reality, it was decided to use the hospital's medical images stored in DICOM format. Because the selected hospital might not have all kinds of images, the effort was to use some of the image types not provided by the hospital from open source databases of medical images that were created for the purpose of the research and set up a complete set.

The medical images collection that was used to evaluate the proposed medical images retrieval system in this study included images of the Tabriz "Behboud" Hospital. To use these images, medical ethics certificates and authorization of the board of the Tabriz "Behboud" Hospital obtained. They accepted to give patient images with considering privacy and removal of patient identification information and their demographic. Because of this removal of information and the lack of proper insertion of the remaining features that were intended for this research, a series of other images included in the collection to complete the collection of images.

As a result, by presenting the variety of images that were used in the standard Evaluation of ImageCLEFmed 2016, (shown in Fig. 14) for Behboud hospital, 10,000 images were requested. The hospital's information officer provided about 1000 images of mammography images, CTs, and radiology due to the timely exclusion of information including patient private information.

Images are stored in DICOM format, a standard for storing and exchanging medical images. The DICOM format has a

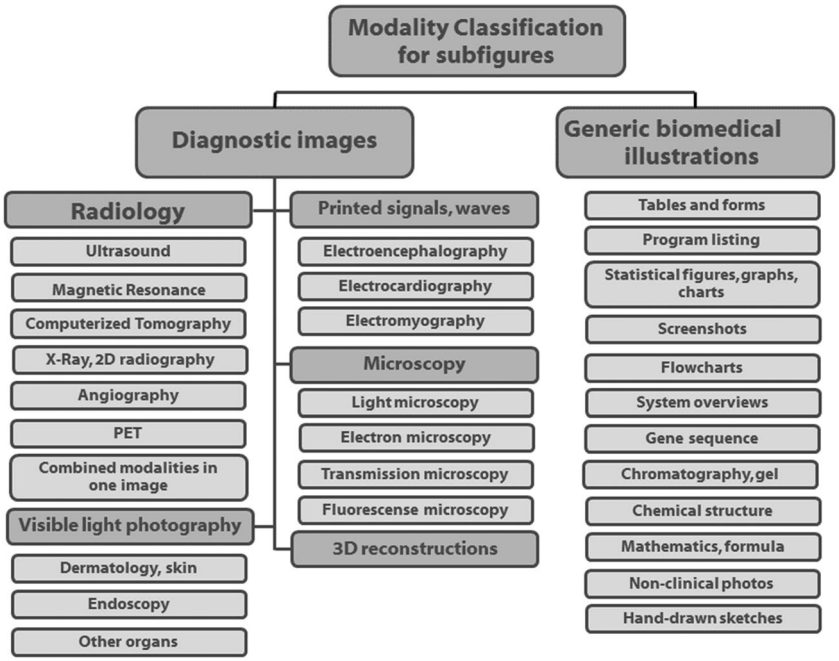

Fig. 14 Hierarchical classification of images that are freely available in biomedical subject literature [33]

standard in the type of features to store the required metadata for storing images, which is based on the work of this research in the division of dimensions. Different types of features associated with a medical image are stored in the DICOM header.

According to a systematic review of 66 articles on the retrieval of medical images [10], out of a total of 50 articles referring to a specific set of images for evaluating their proposed approach, 26 articles (52%) set their images according to their own research goal, 12 articles (24%) used the ImageCLEFmed medical event collection, and the rest of them used other benchmark sets such as IRMA, OASIS, ELCAP, ADNI, and some of them used a composition of these collections.

With the aim of using a benchmark of medical images in DICOM format, to evaluate the suggested indexing technique, various benchmark sets were reviewed.

OASIS: According to the descriptions posted on http://www.oasis-brains.org/, The Open Access Series of Imaging Studies provides access to MRI brain image collections for everyone.

ELCAP[1]: This collection has CT images of the early stages of lung cancer.

ADNI: Alzheimer's disease Neuroimaging Initiative

These three sets are not diverse in the modality of the image and anatomy area.

IRMA: Articles reviewed in the systematic review [10] and used the IRMA collection were studied. All articles were referred to the http://www.irma-project.org/ to access this collection. This website was not available at the time of this research. The following was said about this collection in one of the articles [34]:

"The ImageCLEF Medical Imaging Database is accessed by the IRMA Group from the University of Aachen, Germany. This collection collects anonymous radiographs that are conventionally selected from the Diagnostic Radiography Department of the German University of Technology (RWTH) in Germany. These images show various Ages, Genders, Views, and Damages. In this collection, which is intended for content-based images retrieval application, each image has 120 × 120 pixels. "

Table 3 Types and numbers of images in the collection

Modality	Anatomy area	No. of patients	No. of images
Radiology	Chest	53	53
CT	Lung	1	500
	Chest	4	777
	Lymph nodes	1	661
	Colon	1	1232
	Head and neck	1	1830
	Pancreas	1	240
Mammography	Chest	100	155
MRI	The brain	3	158
	Prostate	1	348
	Chest	3	150
PET/CT	prostate	1	255
Total		170	6355

[1] Early Lung Cancer Action Program

Table 4 The most frequent queries and vocabulary and the number of their repetitions [20]

#	Query	Repetitions	#	Word	Repetitions
1	Mega cisterna magna	118	1	cyst	801
2	Baastrup disease	80	2	mri	545
3	Limbus vertebra	74	3	disease	463
4	Negative ulnar variance	67	4	ct	447
5	Toxic	65	5	syndrome	438
6	Cystitis cystica	50	6	fracture	404
7	Throckmorton sign	46	7	sign	359
8	Double duct sign	45	8	tumor	322
9	Riedel lobe	40	9	bone	294
10	Splenic hemangioma	40	10	pulmonary	293

As a result, a collection of other open source images that were used in numerous medical articles was selected as part of the evaluation, which is explained as follows.

TCIA[2] Collection: Image data in The Cancer Imaging Archive (TCIA), located at http://www.cancerimagingarchive.net, is organized in purposeful subject collections. The total number of images in this collection is 11,518,783. These subjects usually have a common type of cancer and/or anatomy (lung, brain, etc.) [16].

The reasons for choosing this collection are as follows:

1. The images in this collection are all in DICOM format.
2. Has a sufficient variety and number to evaluate this project.
3. The appropriate text descriptors in the DICOM header are suitable as inputs for proposed indexing techniques.

Behboud Hospital images with a subset of the images from the TCIA collection created a set of images with the characteristics and variety needed to evaluate the proposed indexing technique. The types and number of images in the collection of images that are used to evaluate the proposed indexing technique are shown in Table 3.

As shown in Table 3, a total of 6355 DICOM medical images were used to evaluate the search engine implemented using the proposed indexing technique.

3.1.2 3-1-2. Query set

In order to create queries, it is necessary to use the actual information needs of users. To do this, it is necessary to carry out studies on clinical experts in order to identify their important information needs and translate them into queries for the retrieval system [35].

[2] The Cancer Imaging Archive

In spite of the limitations that exist in the estimation of true user behavior, search logs are widely used to understand the search behavior of users, especially in the field of biomedical web search engines. In an article [20], the logs of queries recorded in a medical image retrieval system have been analyzed. This article follows two main objectives: (1) identifying the information needs of medical practitioners in search of radiological resources to create a realistic setting for the ImageCLEFmed event; (2) assessing the query structure that users use in medical search engines to serve as information for designing and promoting the effectiveness of search in medical systems.

The logs of this article have been extracted from a medical search engine called Goldminer. This search engine indexed 250,000 radiological images. To create a query list after pre--processing, there were 23,033 queries in the log, with 14,413 (63%) unique ones. The average number of words per query was 2.24 words. This average number is one word less than PubMed queries and is closer to the number of search terms in the Web search engine. In total, queries are short, and about 90% of them have 3 words or less [20]. Table 4 shows the most frequent queries and vocabulary, and the number of their repetitions.

According to a similar study on the PubMed search engine, and after removing the medical retrieval related queries, the result was that at least two, or at best, three of the following axes were found in specific queries:

1. Anatomy area
2. Modality
3. Pathology

The top ten queries of the study [19] are shown in Table 5. The subject of these queries is anatomy or pathology.

In Table 5, frequent queries have been presented with issues related to the visualization of information needs. In order for the selected queries to be suitable for benchmarking the proposed retrieval system, there must be at least two axes of four axes. As a result, a number of queries from Table 6 should be removed from the list.

Table 5 Top ten queries in the PubMed search log file [19].

Query	Repetitions
Finasteride	3601
Ibuprofen and toxicity not gastrointestinal	3421
One and a half syndrome	1751
#1 and #2	1242
Hypertension	801
Osteoclast tab12	767
Influenza	765
Diabetes	640
Cancer	552
Heart	481

Table 6 Frequent queries with issues related to the visualization of information needs [19]

Query	Repetitions
MRI	58
Ultrasound	42
Otitis media	37
fMRI	33
Cardiac MRI	20
Endoscopy	20
Walsh CT	18
Lung ultrasound	15
Capsule endoscopy	15
Ultrasound for thyroid disorders	15

In this study, the log file referred to in these articles and the top queries in them were used as templates to create a set of queries that could be used to evaluate the proposed indexing technique. The reason for using this method was to get the queries closer to the reality of the search for images in this area.

Using two research [19, 20], based on the combination of words in the query, 22 queries were designed in three different types to measure the power of retrieval function in the proposed retrieval system. These queries include 14 single-word queries to measure retrieval speed according to the single-dimensional and multi-dimensional structure in the image retrieval system, 7 double-word queries, and 1 three-word query.

In two types of 2-word and 3-word combinations, each word must be of a different axis. As previously mentioned, a variety of features include anatomy, modality, pathology, and visual recognition.

3.1.3 3-*1-3. Indexing techniques to compare*

a. *Terrier* Standard Benchmark Platform [36]: This platform is an open source search engine, very flexible, efficient, and effective, which is ready to be implemented on a large set of documents. This search engine implements indexing and retrieval capabilities in the best possible way and provides an ideal platform for rapid development and evaluation of retrieval functions in large collections. This platform is used for research and testing in text retrieval. Research on the TREC standard test kits and CLEF has already been done using this platform. The language of the Terrier is Java, developed by the University of Glasgow, School of Computer Science.
b. *Lucene* the open source search engine [37]: The indexes in Lucene are stored in vector form and the final index is

Table 7 Queries used in the evaluation

Query category	Query type	Row	Query
Simple	Single word (anatomy area)	1	Lung
		2	Chest
		3	Lymph nodes
		4	Pancreas
		5	Abdomen
		6	Head and neck
		7	Brain
		8	Prostate
	Single word (modality)	9	Radiology (CR)
		10	CT
		11	Mammography
		12	MRI
		13	PET
	Single word (pathology)	14	Cancer
Medium	Two words (anatomy area + modality)	15	Chest CR
		16	Lung CT
		17	Brain MRI
	Two words (pathology+ modality)	18	CT Cancer
		19	MRI Cancer
	Two words (anatomy area + pathology)	20	Lymph nodes cancer
		21	Prostate cancer
Complicated	Three words (anatomy + modality + pathology)	22	CT pancreas cancer

created in a vector space model. At the time of the search, a single score is given to each document compared to its vicinity of the query. This score is generated based on the bag of words. In the bag of words method, first, a dictionary is created from the keywords in the search terms. For example, "man, arthritis, fracture, old" are a sample of words that are in most of the documents and form part of the dictionary. Suppose the query is as follows: "Arthritis in the elderly man." In the first step, the words that exist in the dictionary are found. Then for each document, based on the number of repetitions of these keywords in them, a score will be obtained that will score that document. It has already been said that documents in the Lucene method are indexed in vector space. Therefore, in order to obtain time complexity, the analysis should be done in the vector space. The time complexity of search is equal to *O(n)*, where n is the number of dimensions of the vector.

c. Multi-dimensional multi-directional method for biomedical image retrieval biomedical images have dominant spatial information called ***MD²MaMEP*** proposed in [12]:
d. Multi-dimensional indexing techniques proposed in [15] for medical images retrieval in which Relational DBs ***Normalization***-like approach is applied.
e. The proposed text-based multi-dimensional medical image indexing technique in which ***Correlation*** of the features-usages is considered.

3.1.4 3-1-4. Evaluation criteria

As mentioned in the introduction to the evaluation of the performance of information retrieval systems, these characteristics should be considered: *efficiency* and *effectiveness*; effectiveness includes *precision* and *recall* and efficiency includes *execution time* and *memory usage*.

3.2 3-2. Evaluation results

In the following, the two main criteria, namely effectiveness and efficiency, have been evaluated and the results are presented. The collection of images used in the assessment is 6355 images, which is from Tabriz Behboud hospital and a subset of images from the TCIA collection. A set of queries was created with a template of two articles [19, 20] and includes 22 queries.

The proposed indexing technique was implemented using Lucene's open source search engine. In comparison, the proposed technique is compared with the multi-dimensional indexing using the normalization approach [15], with the Lucene search engine's built-in indexing technique and the Terrier evaluation platform. In the following, the results of these conditions for the stated criteria are presented.

3.2.1 3-2-1. Efficiency

The efficiency or productivity criterion evaluates the performance of the retrieval system. Performance includes processing speed and system resource efficiency, or memory usage.

The number of medical images databases in clinical procedures, medical research, and so on is increasing rapidly [38]. An increasing number of clinical experts, researchers, students, and patients are using search engines to search for related medical images [39]. In order to evaluate the performance of an information search system in general, the following should be provided and considered:

- Experimental evaluation environment
- Data set (images)
- Query set (along with the selection of images that have relevancy rate for each query: At least three related images)
- Evaluation criteria
- Evaluation scenarios

To create an environment for evaluating the proposed multi-dimensional indexing technique, this technique was implemented using Lucene the open source search engine. The collection of images that was evaluated as a dataset consisted of medical images stored at the Tabriz "Behboud" Hospital in East Azarbaijan province, as well as a subset of the TCIA collection. For Behboud hospital images, in agreement with the board of directors of this hospital, it was intended that medical images would be selected by the hospital itself, and after patient's personal information was removed from the images, they are used for this research purposes. The composition of the collection of images is explained in Section 3-1-5.

A set of queries was created by using the idea of articles from the same fields [38,3 9]. The multi-dimensional proposed indexing technique is compared with conventional single-index indexing techniques in terms of evaluation criteria that demonstrate the efficiency and effectiveness of the system (Section 3-1-4). In this measurement, the precision (average of precision) and recall are compared as efficiency measures, speed and memory usage as effectiveness criteria in the two systems.

Analyzing memory usage

- In total, from the Behboud Hospital System and the TCIA images, 6355 documents were compiled, at the stage of creation using the vertical fragmentation approach, 13 dimensions were created, and while there is 7 type of features and the number of features is 44,485 numbers. In the dispersion of features in the data structure, if there are 23,450 features in the middle nodes and 21,035 features in the leaves, and the data fields receive an average of 8 bytes and pointer fields of 2 bytes of space, the space

usage complexity for each index node is calculated as follows:

The value of t is 6355 and the M_{total} is calculated with the formula.

t = 6355 k = 21,035n = 23,450d = 13

$$M_{total} = \sum_{i=1}^{n}\left(\overline{M}_{(L)leaf\ node}\right) + \sum_{i=1}^{n}\left(\overline{M}_{(d)middle\ node}\right) + \sum_{i=1}^{n}\left(\overline{M}_{(r)\ root\ node}\right) + M_{Images}$$

$$M_{total} = \sum_{i=1}^{k}(M_{D_a} + M_{D_v} + M_{P_{ID}}) + \sum_{i=1}^{n}(M_{D_a} + M_{D_v} + M_{P_c} + M_{P_{ID}}) + \sum_{i=1}^{d}(M_{D_d} + M_{P_c} + M_{P_n}) + M_{D_m} + M_{P_c} + M_{Images}$$

M_{Images} is 5500 megabytes (5,500,000,000 bytes). Data fields with α and pointer fields with β are given in the following formula. Total memory usage is 444.99 KB.

$$\begin{aligned} M_{total} &= 21,035\ (\alpha+\beta) + 23,450\ (\alpha+\beta) + 13(\alpha+\beta) \\ &\quad + \Big)\alpha+\beta(+5.500.000.000 = 44,499\ )\alpha+\beta\Big(\\ &\quad + 5.500.000.000 \\ &= 44,499\ (8+2) + \ \ 5.500.000.000 \\ &= 5,500,444,990\ \text{Bytes} = 5500\ \text{Megabytes} \end{aligned}$$

This value should be divided by the number of documents, so that the average memory usage should be the amount of memory used for each document.

$$\overline{M}_{D} = \frac{M_{total}}{t} = \frac{5500}{6355} = 0/8654\ \text{MegaByte} = 865\ \text{kilobytes}$$

– By the Normalization approach [15], from the Behboud Hospital System and the TCIA images, 6355 documents were compiled, at the stage of creation using the vertical fragmentation approach, 13 dimensions were created, and while there is 7 type of features and the number of features is 44,485 numbers. In the dispersion of features in the data structure, if there are 34,123 features in the middle nodes and 10,362 features in the leaves, and the data fields receive an average of 8 bytes and pointer fields of 2 bytes of space, the space usage complexity for each index node is calculated as follows:

t = 6355k = 10,362n = 34,123d = 4

$$M_{total} = \sum_{i=1}^{n}\left(\overline{M}_{(L)\ leaf\ node}\right) + \sum_{i=1}^{n}\left(\overline{M}_{(d)\ middle\ node}\right) + \sum_{i=1}^{n}\left(\overline{M}_{(r)root\ node}\right) + M_{Images}$$

$$M_{total} = \sum_{i=1}^{k}(M_{D_a} + M_{D_v} + M_{P_{ID}}) + \sum_{i=1}^{n}(M_{D_a} + M_{D_v} + M_{P_c} + M_{P_{ID}}) + \sum_{i=1}^{d}(M_{D_d} + M_{P_c} + M_{P_n}) + M_{D_m} + M_{P_c} + M_{Images}$$

M_{total} = 10,362 (α + β) +34,123 (α + β) + 4(α + β) +)α + β(+5,500,000,000 = 444.9)α + β(+ 5,500,000,000 = 44,490 (8 + 2) + 5,500,000,000 = 5,500,444,900 Byte = 5500 MB

$$\overline{M}\ D = \frac{M_{total}}{t} = \frac{5500}{6355} = 0.8654\ \text{MB} = 865\ \text{Kilobytes}$$

– To calculate the average memory usage per document, for the Lucene search engine, its index memory should be added to the memory usage of the documents and divided by the number of documents. The amount of space used for the Lucene index is 2500 KB, which is divided by the number of documents, 6355 documents, will be 0.393 KB per document.

$$\frac{2500+5.500.000}{6355} = 865\ \text{kilobytes}$$

4 3. For Terrier, the index size was 418 KB, which is divided by 6355 and is 0.065 KB per document.

$$\frac{0.065+5.500.000}{6355} = 865\ \text{Kilobytes}$$

Average memory usage is compared in Fig. 15. By comparing the results presented in Fig. 15, the superiority of the proposed indexing technique can be observed to be compared to the two control techniques. The multi-dimensional technique compared to Terrier has been about three times better and for Lucene, it is about 5 times higher.

Execution time The measurement of this criterion should be made for four operations that are used in an indexing technique. These operations include creating the index data structure, inserting a new image, deleting the existing image, and searching. In Section 2-3, along with the definition of operations, the complexity of the time of each operation was expressed theoretically. In this section, the time of each operation is calculated and presented here.

Figure 16 shows the measured time for creating, insertion, and deletion operations for the proposed indexing technique with the proposed approach, the indexing of the Lucene and Terrier search engine.

It can be clearly seen that Lucene is leading the way in creating the index, then the proposed technique. Terrier has been working at a slower pace. Because creation is done only

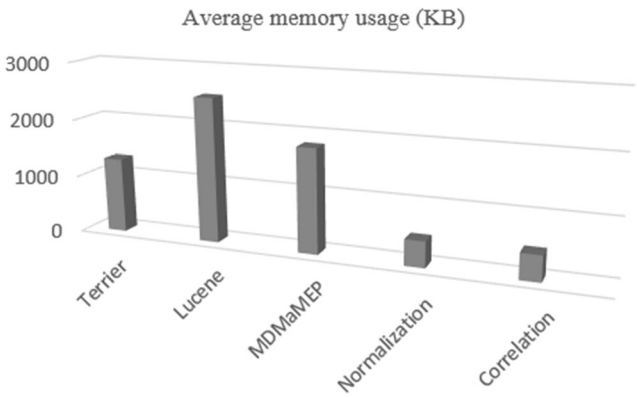

Fig 15 Memory usage in each indexing technique in the kilobyte

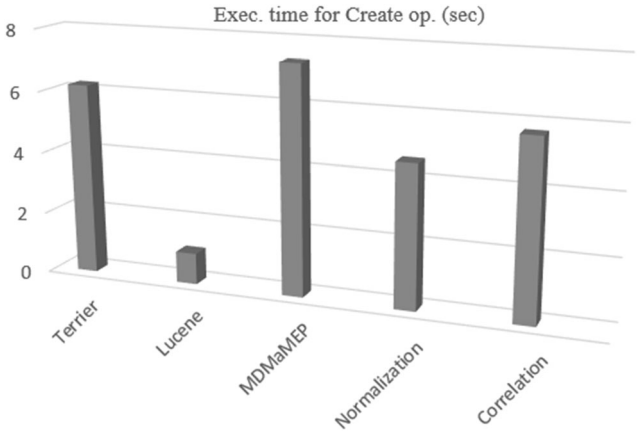

Fig. 16 Creation time through the second

once, it is less important than a search operation, and if the created structure can improve retrieval speed, then this time can be considered a good time in terms of better retrieval speed.

In Fig. 16, the time of creation in the proposed multi-dimensional indexing techniques can be compared with other relative techniques and also engines such as Lucene indexing and indexing in the Terrier platform.

Figures 17, 18, and 19 show the insertion and deletion time comparison.

In Fig. 18, the average search for the tested techs is shown, in which it is better to see the difference in the time of the searches.

For the comparison of different level of complexities, the retrieval speed difference for simple, moderate, and complex categorization in queries is given in Fig. 19.

Figure 19 shows faster retrieval speeds in all techniques in a simpler query. After that, the medium and then the complex query, respectively, have faster retrieval rates.

Effectiveness This criterion is used to measure the accuracy of retrieval. The key criterion for determining the quality of the information retrieval process is relevance. The relevance

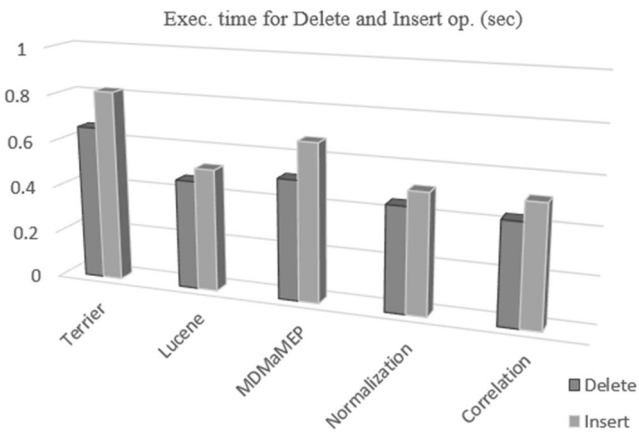

Fig. 17 Insert and delete execution time

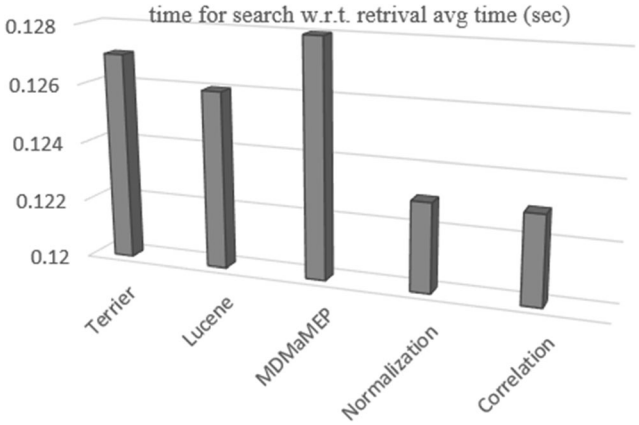

Fig. 18 The difference in search time with regard to the average retrieval speed

shows the correctness rate of retrieval. In order to formulate the relevancy with considering different aspects of it, it can be said that an information retrieval system can be evaluated in terms of relevance, only by obtaining the following information [4]:

1. A benchmark of documents
2. A benchmark set of queries
3. A binary judgment of the relevance of the document to the query

Secondly, in order to assess the relevance of the criteria introduced for measuring, they usually calculate the true value of relevance based on a set of documents and queries. These criteria include precision and recall.

The precision (p) is the fraction of the retrieved documents that are related to a query and provides a degree of soundness for the system. Precision does not care about the total number of documents considered relevant by the information retrieval system. This aspect is defined by the recall criteria (R), which is defined as the fraction of the truly relevant documents that are properly retrieved, and therefore regarded as a measure of the completeness of the system. Generating a fast, but inefficient response often does not seek user satisfaction, and

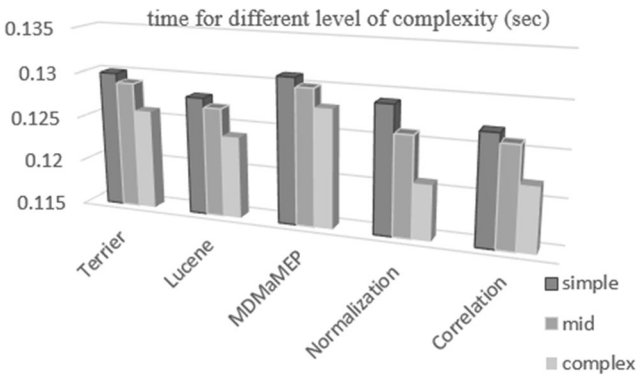

Fig. 19 Comparison of speed in the category of queries

certainly, the ultimate goal of data retrieval is to satisfy user satisfaction [4].

$$P = \frac{|TP|}{|TP| + |FP|} = \frac{|TP|}{Retrieved} \qquad R = \frac{|TP|}{|TP| + |FN|} = \frac{|TP|}{Relevants}$$

Precision According to the results of measuring precision, the proposed multi-dimensional indexing technique with the vertical fragmentation approach provides the highest precision in retrieval and, with the distance from both control search engines, results in the superiority of multi-dimensional indexing.

Figure 20 shows the average precision in the diagram for a better comparison between indexing techniques. The proposed multi-dimensional indexing technique with the vertical fragmentation approach has shown the highest precision with 97.6 in retrieval. This technique has been able to show the most relevant items in retrieval results.

For the comparison shown in Scenario 5, the difference of precision in retrieval for simple, moderate, and complex categorization in queries is given in Fig. 21.

Figure 21 shows as a query is simple, the precision of answers gets better no matter what the technique is. After that, with a more complex query, the precision of retrieval has dropped.

Recall The measured average recall represents a better recall for multi-dimensional indexing. That is, from the images related with each query this technique has been able to retrieve more images. Figure 22 shows the average recall chart in the evaluated techniques.

As shown in Fig. 23, there is a clear difference between the proposed multi-dimensional technique and single-dimensional techniques that were selected for comparison. The multi-dimensional technique with the proposed approach can provide a better recall than one-dimensional techniques.

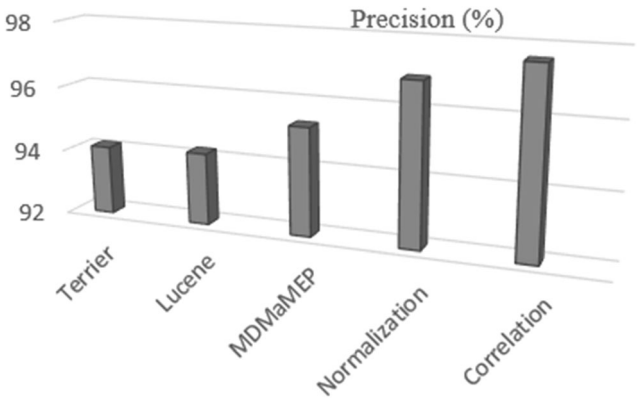

Fig. 20 Average precision in the search operation of indexing techniques

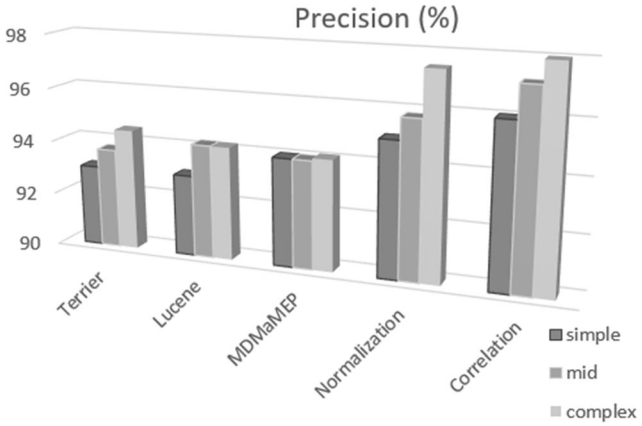

Fig. 21 Comparison of the precision in the categories of queries

For the comparison of different level of complexities, the difference in retrieval recalls for simple, moderate, and complex categorization in queries is shown in Fig. 23

Figure 23 shows a higher retrieval recall in all techniques in complex queries. As the query is simplified, the retrieval recall is reduced.

The evaluation of the proposed multi-dimensional indexing technique explained in detail. Settings considered for this evaluation, including evaluation environment, set of images, queries set, indexing techniques used for comparison, evaluation criteria, and scenarios designed for evaluation. Finally, the results from the use of these settings were included in the evaluation of the proposed multi-dimensional indexing technique.

A brief qualitative comparison of the proposed text-based multi-dimensional indexing technique with the other relent ones is depicted in Table 8.

5 Conclusions and future work

Medical imaging is a popular and profitable method in healthcare application. Number of medical images is accumulatively growing and this huge amount of medical image

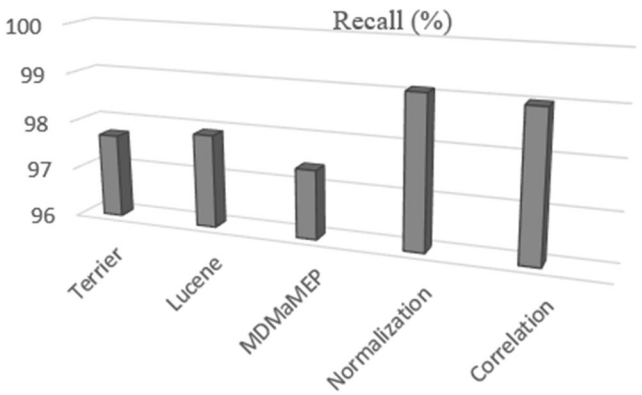

Fig. 22 The average recall in the search operation of indexing techniques

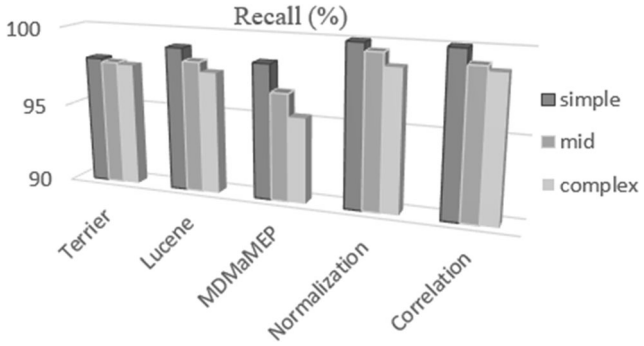

Fig. 23 Comparison of the recall in the categories of queries

documents needs powerful search engine systems to be utilized. Indexing, as an important part of information retrieval systems can have serious impact on search engines' performance.

Most of the medical indexing techniques are content-based that need to process and analyze medical image content which has some drawbacks. In contrast, text-based indexing of medical images can provide some benefits that are discussed in this paper and issued in some are publications.

In this paper, a text-based multi-dimensional indexing technique (both its data structure and operations) is designed for medical image retrieval in which correlation between image features-usage (based on users' queries) is taken into account. At the first step, set of image features (based on the standard format such as the DICOM) is fragmented into some subsets; pairwise correlation of the features are computed through performing *Quantitative Association Rule Mining* (QARM) technique on data set of previous users queries on the set of image features. The result of this QARM algorithm is Affinity matrix of features, used as the input for the *vertical fragmentation*-like procedure (used in distributed relational database design). So, the set of image features are fragmented and partitioned into subsets (in each features with the most affinity reside).

In the next step, a hierarchical clustering algorithm is applied for each of the subsets (a.k.a. a *Dimension*) to convert the set of features in a dimension into a tree (i.e., hierarchy).

Considering features-usage correlation besides traversing the hierarchy of them in a dimension cause to have precision (semantic correlation of the features) as well as performance (response time), together.

Evaluation results (memory usage and time complexity analysis in addition to the experimental evaluations) show that, in terms of efficiency, the proposed text-based multi-dimensional indexing technique occupy less memory and has a good rate of search and insertion and removal operations. It should be noted that in the area of improving the power of retrieval systems, time is an important factor trying to improve it in milliseconds or even more precisely. As a result, even a few milliseconds represents a significant improvement. In the creation operation, Lucene's technique was pioneered. But given that the creation occurs only once and is not prioritized to other operations, it can be considered appropriate if the technique has better retrieval time. The criterion that compared to efficiency has significant importance in the field of evaluation of retrieval systems, is the effectiveness and more important the precision. Regarding the effectiveness criterion, the multi-dimensional indexing technique showed good superiority in precision and recall. Since queries that are getting submitted to an information retrieval system in real life, are not all of a unique form and are diverse and complex, the queries that were selected for evaluation also have this feature of diversity. Given this, it is tried to measure the effect of the simple and complex query on retrieval. Comparison of simple, moderate and complex queries at different criteria showed that as for the speed, as simple as a query is, the faster retrieval takes place. The precision criterion is increased in complex queries, and about the recall criterion, complex query provides a better recall. In comparison to the proposed multi-dimensional indexing technique and two single-dimensional control techniques, at each of three levels of query complexity, the multi-dimensional proposed technique was superior in *precision* and *recall*.

Some future researches can be as follows:

- Since the proposed indexing technique is sensitive to dirty data (e.g., missing value or noisy data), proper data cleansing method (as the step zero), or a dirty data resilient version is required
- Adaptive indexing and retrieval that dynamically re-calculate affinity of the features based on the users' query statistics and its enforcement

Table 8 Brief qualitative comparison of the proposed text-based multi-dimensional indexing techniques

Critera	Terrior	Lucene	Mdmapet	Normalization	Correlation
Average memory usage	Mid	Very high	High	Low	Low
Average execution time (create)	High	Low	Very high	Mid	High
Average execution time (search)	Very high	High	Very high	Mid	Mid
Precision	Low	Low	Mid	High	Very high
Recall	Mid	Mid	Low	High	High

– Use of deep-learning techniques for multiclass classification of the medical images regardless of the predefined features for the images

References

1. Herpe G, Lederlin M, Naudin M, Ohana M, Chaumoitre K, Gregory J, Vilgrain V, Freitag CA, De Margerie-Mellon C, Flory V, Ludwig M (2020) Efficacy of chest CT for COVID-19 pneumonia in France. Radiology
2. Xie X, Zhong Z, Zhao W, Zheng C, Wang F, Liu J (2020) Chest CT for typical coronavirus disease 2019 (COVID-19) pneumonia: relationship to negative RT-PCR testing. Radiology 296(2):E41–E45
3. Guo J, Fan Y, Pang L, Yang L, Ai Q, Zamani H, Wu C, Croft WB, Cheng X (2020) A deep look into neural ranking models for information retrieval. Inf Process Manag 57(6):102067
4. Ceri S, Bozzon A, Brambilla M, Della Valle E, Fraternali P, Quarteroni S (2013) Web information retrieval. Springer Science & Business Media
5. Hwang KH, Lee H, Choi D (2012) Medical image retrieval: past and present. Healthc Inform Res 18(1):3–9
6. Kumar A, Kim J, Cai W, Fulham M, Feng D (2013) Content-based medical image retrieval: a survey of applications to multidimensional and multimodality data. J Digit Imaging 26(6):1025–1039
7. Owais M, Arsalan M, Choi J, Park KR (2019) Effective diagnosis and treatment through content-based medical image retrieval (CBMIR) by using artificial intelligence. J Clin Med 8(4):462
8. Das P, Neelima A (2017) An overview of approaches for content-based medical image retrieval. Int J Multimed Inform Retr 6(4): 271–280
9. Ghosh P, Antani S, Long LR, Thoma GR (2011) Review of medical image retrieval systems and future directions. In2011 24th International Symposium on Computer-Based Medical Systems (CBMS). IEEE:1–6
10. Habibi Asl S, Safaei AA (2016) Medical image retrieval approaches, methods and systems: a systematic review. Pajoohandeh J 21(2):61–73
11. Ayadi H, Torjmen KM, Daoud M, Huang JX, Ben Jemaa M (2018) MF-Re-Rank: a modality feature-based Re-Ranking model for medical image retrieval. J Assoc Inf Sci Technol 69(9):1095–1108
12. Galshetwar GM, Waghmare LM, Gonde AB, Murala S (2018) Multi-dimensional multi-directional mask maximum edge pattern for bio-medical image retrieval. Int J Multimed Inform Retr 7(4): 231–239
13. Tseng FS, Lin WP (2006) D-tree: A multi-dimensional indexing structure for constructing document warehouses. J Inf Sci Eng 22 (4):819–842
14. Laal M (2013) Innovation process in medical imaging. Procedia Soc Behav Sci 81:60–64
15. Safaei Ali A (2021) Habibi-Asl Saeedeh, Multidimensional indexing technique for medical images retrieval, accepted to be published in the Journal of Intelligent Data Analysis
16. Böhm C, Berchtold S, Kriegel HP, Michel U (2000) Multidimensional index structures in relational databases. J Intell Inf Syst 15(1):51–70
17. Ayadi MG, Bouslimi R, Akaichi J (2016) A medical image retrieval scheme with relevance feedback through a medical social network. Soc Netw Anal Min 6(1):1–23
18. Nieuwenhuis R, Hillenbrand T, Riazanov A, Voronkov A (2001) On the evaluation of indexing techniques for theorem proving. In: International Joint Conference on Automated Reasoning. Springer, Berlin, Heidelberg, pp 257–271
19. Muller H, Kalpathy-Cramer J, Hersh W, Geissbuhler A (2008) Using Medline queries to generate image retrieval tasks for benchmarking. Stud Health Technol Inform 136:523–528
20. Tsikrika T, Müller H, Kahn Jr CE (2012) Log analysis to understand medical professionals' image searching behaviour. Quality of Life through Quality of Information. IOS Press 1020–1024
21. Özsu MT, Valduriez P (1999) Principles of distributed database systems. Prentice Hall, Englewood Cliffs
22. Muthuganesan R, Chandrasekar VK (2019) Characterizing non-classical correlation using affinity. Quantum Inf Process 18(7):1–3
23. Anandh A, Mala K, Suresh BR (2020) Combined global and local semantic feature–based image retrieval analysis with interactive feedback. Meas Control 53(1-2):3–17
24. Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP (2003) Random forest: a classification and regression tool for compound classification and QSAR modeling. J Chem Inf Comput Sci 43(6):1947–1958
25. Zhang C, Zhang S (2003) Association rule mining: models and algorithms. Springer
26. Ke Y, Cheng J, Ng W (2008) An information-theoretic approach to quantitative association rule mining. Knowl Inf Syst 16(2):213–244
27. Gülagiz FK, Sahin S (2017) Comparison of hierarchical and non-hierarchical clustering algorithms. International Journal of Computer Engineering and Information Technology 9(1):6
28. Kaur M, Kaur U (2013) Comparison between K-mean and hierarchical algorithm using query redirection. Int J Adv Res Comput Sci Softw Eng 3(7):1454–1459
29. Park H, Kwon K, Khiati AI, Lee J, Chung IJ (2015) Agglomerative hierarchical clustering for information retrieval using latent semantic index. In 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity) (pp.426–431). IEEE
30. Rafsanjani MK, Varzaneh ZA, Chukanlo NE (2012) A survey of hierarchical clustering algorithms. J Math Comput Sci 5(3):229–240
31. Manning CD, Raghavan P, Schütze H (2008) Xml retrieval. In: Introduction to information retrieval. University Press, Cambridze
32. Shirkhorshidi AS, Aghabozorgi S, Wah TY (2015) A comparison study on similarity and dissimilarity measures in clustering continuous data. PLoS One 10(12):e0144059
33. De Herrera AG, Bromuri S, Schaer R, Müller H (2016) Overview of the medical tasks in ImageCLEF 2016. CLEF Working Notes, Evora
34. Srinivas M, Naidu RR, Sastry CS, Mohan CK (2015) Content based medical image retrieval using dictionary learning. Neurocomputing. 168:880–895
35. Müller H, Clough P, Hersh W, Deselaers T, Lehmann T, Geissbuhler A (2005) Evaluation axes for medical image retrieval systems: the imageCLEF experience. In Proceedings of the 13th annual ACM international conference on Multimedia Nov 6 (pp. 1014–1022)
36. Ounis I, Amati G, Plachouras V, He B, Macdonald C, Johnson D (2005) Terrier information retrieval platform. In: European Conference on Information Retrieval. Springer, Berlin, Heidelberg, pp 517–519
37. Goetz B (2000) The Lucene search engine: Powerful, flexible, and free. JavaWorld. Available http://www.javaworld.com/javaworld/jw-09-2000/jw-0915-lucene.html
38. Siong LC, Zaki WM, Hussain A, Hamid HA (2015) Image retrieval system for medical applications. In 2015 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 73–77). IEEE

39. Murphy SN, Herrick C, Wang Y, Wang TD, Sack D, Andriole KP, Wei J, Reynolds N, Plesniak W, Rosen BR, Pieper S (2015) High throughput tools to access images from clinical archives for research. J Digit Imaging 28(2):194–204

Ali Asghar Safaei received the B.Sc. degree in computer software engineering from Shahid Sattari Air University, Tehran, Iran, in 2001, the M.Sc. degree in computer software engineering from Ferdowsi University, Mashhad, Iran, in 2004, and the Ph.D. degree in computer software engineering from the Iran University of Science and Technology, Tehran, in 2011. He is currently Assistant Professor of the Department of Computer Engineering,Tarbiat Modares University, Tehran, Iran. His research interests include medical software system design, Big Data, information retrieval, IoT, and also VR/AR in healthcare.