International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 25, No. 2, April, 2020, pp. 215-225
Journal homepage: http://iieta.org/journals/isi

Efficient Platform as a Service (PaaS) Model on Public Cloud for CBIR System

Fairouz Hadi, Zibouda Aliouat*, Sarra Hammoudi

Computer Science, Laboratory LRSD, Ferhat Abbas University, Sétif 19000, Algeria

Corresponding Author Email: zaliouat@univ-setif.dz

https://doi.org/10.18280/isi.250209

Received: 4 December 2019
Accepted: 19 February 2020

Keywords:
clouds, CBIR, computer network reliability, decision trees, DICOM, wireless communication

ABSTRACT

Medical image processing requires handling a huge amount of data. Unstructured big data can create some issues related to latency. Distributed architectures based on parallelism can alleviate a latency problem. Also, achieving image availability in case of a server failure in such a huge system increases the latency time. To solve the challenges of latency in image processing in an enormous system, we propose a new platform for information retrieval in databases consisting of digital imaging communication in medicine (DICOM) files. The platform is based on a Decision Tree. The servers in this platform are distributed and work in a parallel way. Also, a fault tolerant system based on time triggered protocol is proposed to ensure image availability and minimize image recovery latency in the case of a server failure. The main goal of this proposal is to select images from DICOM files similar to an image proposed in a query, using the principle of content based image retrieval (CBIR). Also, this platform helps radiologists with the diagnosis of medical images.

1. INTRODUCTION

For several years, researchers have been interested in content search methods that would make it possible to find objects in databases using only the digital content of images or signals. These search methods function by describing the objects sought, the objects in the databases that have previously been automatically processed, or directly using the digital objects in demand. This is a very active area of research, particularly in the medical field [1].

Currently, managing a medical image database is a very difficult task. A search and automatic retrieval system is required to manage medical image databases [2]. The management and indexing of these large image databases are becoming more complex [3].

Content-based image retrieval (CBIR) is a growing field with the advancement of new capabilities in diagnostic-image technology. CBIR systems have absolutely changed the way pathologists diagnose a patient, facilitating computer-assisted diagnosis by retrieving images that are similar to a query image. Retrieved images are classified according to their relevance [4].

The common file format for medical images is Digital Imaging and Communications in Medicine (DICOM). The DICOM format could contain additional information regarding image modality, patient identification, and raw image data, and so forth [3].

The increasing volume of medical images leads to the need to design accurate systems for efficient storage and indexing of these images. Improving the techniques of analysis and recovery of medical images is one of the biggest challenges. The annual image production of the major centers of radiology is voluminous. Implementing computer techniques for efficient indexing management, automated processing, and relevant research requires efficient, fast and accurate development of decision support tools. The deployment of such tools should be carried out carefully due to the particular characteristics of the medical field [5].

Big Data is used as a generic name for complex and huge amount of data collected from different sources. Big Data is not just about the size of the data itself, but also addresses the management and storage of data for a possible analysis. As humans digitize, computing includes data with greater volume, velocity and variety [6, 7].

A system works properly even if a server is out of order needs data replication, so it is important that data be organized when stored on distributed servers [8].

Cloud Computing provides on-demand resources and services delivery over the Internet. It allows the storage and access to data via the Internet rather than an individual PC's resources. Cloud computing provides unlimited storage and processing resources for users [8, 9].

Among the most popular data mining algorithms are those for building decision trees used in statistics and machine learning [1]. A decision tree can analyze the information included in a vast data source and identify valuable rules and relationships. Usually, decision trees are deployed for the purpose of prediction /classification [10]. A decision tree is a classifier uttered as a recursive partition of the instance space [11]. The success of decision trees lies largely in their readability, as opposed to "black box" algorithms such as neural networks [1]. One of the biggest advantages is that users don't need to know a lot of background knowledge in the process of learning, which is also the biggest drawback [12]. Big Data classification can be performed using the decision tree approach [7].

In this work, we propose a novel Platform as a Service (PaaS) Model on public Cloud for CBIR System. The PaaS-CBIR organizes the storage of our images to minimize the latency time. PaaS–CBIR provides monitoring, backup and recovery services that ensure data availability with minimum latency time.

The rest of the paper is organized as follows: Section 2 presents related work and classification of these works. Section 3 gives a CBIR description. Section 4, presents the Cloud Computing overview. Section 5 discusses the aim of the PaaS-CBIR. Section 6 explains the proposed methodology. In section 7, system monitoring, backup and recovery purposes in PaaS-CBIR are described. The formal analysis of our PaaS-CBIR is presented in section 8. Finally, we conclude our paper and provide future perspectives in section 9.

2. RELATED WORK

This section explains the different tools and methods used to implement a CBIR system. The visual descriptors such as texture, color, and shape constitute the basis of almost all the CBIR systems [4].

Dimitrovski et al. [3] proposed a web-based system for medical image retrieval that utilized the Oracle Multimedia features for image retrieval. The main objective was to show the use of the Oracle Multimedia descriptors for data management and retrieval. In this work, Dimitrovski et al. manipulated the DICOM format, which could contain additional information regarding image modality, patient identification and raw image data ... etc. Dimitrovski et al. combined both content-based and text-based data. To calculate similarity between query image and images within the database, and also to extract features of an image such as color, texture and shape, they used a solution based on the methods provided by Oracle. In this work, the authors did not indicate the methods they employed for the extraction of characteristics (color, texture, shape), nor did they indicate the method used for similarity measure.

Ramamurthy and Reddy [4] proposed a CBIR system for a medical images database. The CMBIR aids the pathologists in the patient's diagnosis. The edge-based texture feature extraction technique is used to extract texture. The edge histogram descriptor is used for extract shape. The authors combined the two techniques to yield a single feature vector. The authors used the Euclidean distance formula to calculate the similarity between the query images and images within the database. The main goal of the system was to improve the efficiency of image retrieval compared to the use of a single feature. Because this system combined shape and texture, it provided roughly 33%-91% precision and 33%-75% recall.

Manoj and Manglem [2] proposed a CBIR system using medical images. To implement this system, the authors began with feature extraction such as texture and intensity. They used the Local Binary Pattern (LBP) Algorithm for Texture Extraction. The authors focused their work on images of the eye. To enhance retrieval, they also used Euclidian Distance to calculate the similarities between query image and database images. Because this system combined color and texture, it provided roughly 40.86%-89.18% precision and 34.86%-78.12% recall.

Kusrini et al. [13] proposed extracting the characteristics of the image to make a classification using k-means and a decision tree algorithm. The authors used statistical moments for the color, and Haralick's co-occurrence matrix for texture. To classify a query image, the authors extracted its features and compared them with the rules constructed in the training process. Unfortunately, this method failed to retrieve similar images.

Table 1. Classification of the works described in the Section 2

			Meena et al. [14,15]	Kusrini et al. [13]	Paiz-Reyes et al. [16]	Mahmoudi et al. [17]	Ramamurthy et al. [4]	Manoj et al. [2]	Ivica et al. [3]
Content-based			X	X	X	X	X	X	X
Text-based					X				X
QBE(Query By Example)			X	X	X	X	X	X	X
Global properties of the image	color	Intensity			X	X		X	Extraction of the attribute without any detail
		Histogram	X			X			
		Color moments (μ, σ,θ)		X					
	Texture	Matrix of co-occurrences		X					Extraction of the attribute without any detail
		LBP			X			X	
		Wavelet transform	X			X			
		Edge based feature extraction					X		
Local properties of the image	Division of an image in the region								
	Segmentation				X				
	Wavelet transform		X			X			
Shape	Boundary-based descriptor						X		Extraction of the attribute without any detail
	Region-based descriptor				X				
Medical images	DICOM								X
	Other						X	X	
General images			X	X	X	X			
Similarity measure	Euclidean distance		X		X		X	X	Without any detail
	K-NN (K- Nearest Neighbor)					X			
Cloud	SaaS		X			X			
	PaaS								
	IaaS								

Meena et al. [14], Meena and Bharadi [15] proposed a CBIR system that is implemented on Cloud as SaaS services using the Windows Azure platform. The authors used the wavelet transform to extract digital features from images such as color, texture and shape. The authors proposed a hybrid method, applying the Kekre transform for the global properties of the image, and the Haar, Walsh, DCT, Hartley transforms for the local properties of the image. The combination of all these transformations generated good results. The similarity between the database image and the query image is performed using Euclidian Distance.

Paiz-Reyes et al. [16] have proposed GIF (Graphics Interchange Format) Image Retrieval in Cloud Computing Environment. The data collection is manually classified into four categories. The authors used the 3D color histogram in the HSV color space for the color-based image descriptor, the LBP (Local Binary Patterns) is applied for Texture-Based Descriptor and the Zernike moments are applied for Shape-Based Descriptor. Feature vectors are saved in a hash table as a dictionary. LSH (locality sensitive hashing) is used for indexing. The LSH aims to maximize the probability of "collision" for similar elements. The Euclidean distance is used to find similar images.

Mahmoudi et al. [17] proposed a CBIR system implemented on the Cloud using the GPU (Graphical Processing Unit) platform. The CUDA (Compute Unified Device Architecture) API (Application Programming Interface) is used to exploit NVIDIA cards. The OpenCL (Open Computing Language) framework is used to exploit ATI (Array Technologies Incorporated) / AMD (Advanced Micro Devices) graphics cards. Combining the two SIFT (Scale Invariant Feature Transform) descriptors with the SURF (Speeded up Robust Feature) provides good results for extracting image features, since both approaches are most useful for detecting and matching image characteristics. SIFT and SURF methods are invariant in terms of rotation, scaling, illumination, and translation etc …. Regarding the comparison, Mahmoudi et al., use BF (Brute Force) Matcher and FLANN (Fast Library for Approximate Nearest Neighbors) Matcher. To find similar images the KNN algorithm (k-Nearest Neighbors) is used.

According to our knowledge, our current paper is the first to tackle the Platform as a Service (PaaS) for CBIR system.

Table 1 presents a classification of the works described above. Each proposal has its own working principle, objective, and methodology.

3. CBIR DESCRIPTION

The indexing methods include the following steps to retrievel images among others:

1) A signature or index of the image serves as a characteristic to recognize and compare the image with other images.
2) A measure of similarity (or distance) that makes it possible to compare image signatures and to associate similar images.

Research algorithms which are based on the two previous tools and make it possible to quickly find the searched objects [1].

3.1 Symbolic characteristics

The principal features of the image are color, texture, and shape.

3.1.1 Color

Color is often the first descriptor (feature) utilized for image search. Although the majority of images are in the RGB (Red, Green, and Blue) color space, other spaces such as HSV (Hue, Saturation, Value) or the CIE Lab and Luv spaces are much better concerning human perception and are more frequently used [18]. In specialized fields, specifically in the medical domain, the color or grey-level characteristics are frequently of very limited expressive power [18].

Many attributes of color, such as histograms, moments, and more, can be used to characterize color.

- Color histograms: Color histograms are easy and quick to calculate and robust relative to rotation and translation (Eq. (1)) [19].

We have:

$c = I\ (i, j)$

I: image ($M \times N$) pixels

c: color belonging to the color space C

h: vector with n components ($h_{c1}, h_{c2}, \ldots, h_{cn}$)

h_{cj}: the number of color pixels c_j in image I.

$$\sum_{i=1}^{n} h_{ci} = MN \tag{1}$$

where, MN is the number of pixels in the image I.

- Color moments: The color histograms method uses full color distribution and required a large amount of data storage. Rather than calculating the complete distribution, some image search systems use only dominant color features such as the mean of c (symbolized as μ_c), the standard deviation(σ_c), and the skewness(θ_c). These features are calculated on each color component by the following formulas: Eq. (2), Eq. (3) and Eq. (4), respectively [20].

$$\mu_c = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^{c} \tag{2}$$

$$\sigma_c = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(p_{ij}^{c} - \mu_c\right)^2\right]^{\frac{1}{2}} \tag{3}$$

$$\theta_c = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(p_{ij}^{c} - \mu_c\right)^3\right]^{\frac{1}{3}} \tag{4}$$

where, M and N are the horizontal and vertical sizes of the image and p_{ij}^{c} is the value of color c in the image's row i and column j.

3.1.2 Texture

Because “visual texture” is only vaguely defined, texture measures possess an even larger variety than color measures. Texture measures attempt to capture the features of the image or image parts compared to changes in a certain direction and the changes of the scale. These measures are especially helpful for regions with homogeneous texture. Texture measures are invariant compared to image rotations and scale changes or shifts , can be enclosed into the attribute space yet information on the texture can get lost in this process [18].

In 1973, Haralick [21] proposed 14 statistical characteristics extracted from a co-occurrence matrix. Currently, only the four most appropriate characteristics are widely used. Energy, Entropy, Contrast and the Inverse Difference Moment (IDM) (homogeneity).

3.1.3 Shape

The shape is an essential visual attribute and one of the basic features for demonstrating the content of an image. The accuracy of shape descriptors is based largely on the segmentation scheme applied to split an image into meaningful objects. However, the description of forms is a difficult task. A good shape feature should be invariant to scaling, rotation, and translation.

The invariant moments of Hu [22] describe the shape through statistical properties. They are simple to manipulate and are robust to geometric transformations such as rotation, translation, and scaling.

3.2 Similarity measure between descriptors

Once the database is indexed based on characteristics and features such as color, texture, and shape, the most relevant images are searched. In most CBIRs, the distance between the features of any two images is used to measure their similarity. The smaller the distance value, the smaller the difference between images, and therefore, the more similar images [23]. According to the simplicity's Minkowski distance method and the need for representing each feature by normed vector space we choose the Minkowski distance.

This distance is calculated between the descriptor/feature vectors, it is based on L_r norm which is defined by Eq. (5):

$$L_r(V_1, V_2) = \left[\sum_{i=0}^{n} |V_1(i) - V_2(i)|^r\right]^{1/r} \tag{5}$$

V_1 and V_2 are the descriptor vectors.

When r=1 or r=2 or r=∞ , it is called the Manhattan distance, Euclidean distance, and the Chebyshev distance respectively as expressed below in Eq. (6), Eq. (7) and Eq. (8) accordingly:

$$L_1(V_1, V_2) = \sum\nolimits_{i=0}^{n} |V_1(i) - V_2(i)| \tag{6}$$

$$L_2(V_1, V_2) = \left[\sum_{i=0}^{n} |V_1(i) - V_2(i)|^2\right]^{1/2} \tag{7}$$

$$L_\infty(V_1, V_2) = \left[\sum_{i=0}^{n} |V_1(i) - V_2(i)|^\infty\right]^{1/\infty} \tag{8}$$

In this paper, Euclidean distance is used as a similarity measure between the descriptors of the compared images.

3.3 Performance evaluation

The most common measures for evaluating a system are the response time and the space used. In addition to these two measures, we are interested in measures such as precision and recall. These metrics are the most commonly used to evaluate the performance of image retrieval algorithms. The precision of retrieval is defined as the fraction of the retrieved images that are indeed relevant for the query compared to the total returned illustrated in Eq. (9); Recall is the fraction of relevant images returned by the query compared to the total number of relevant images in the database (see Eq. (10)) [23].

A good retrieval system ought to have high values for precision and recall.

$$Precision = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ Number\ of\ images\ retrieved\ from\ the\ database} \tag{9}$$

$$Recall = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ Number\ of\ relevant\ images\ in\ the\ database} \tag{10}$$

4. CLOUD COMPUTING OVERVIEW

Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources [24].

4.1 Components of cloud computing

Amin et al. [25] sort the Cloud Computing components into three main classes: distributed servers, clients, and data centers.

- ***Clients*** are the end-users' terminals that they use to manage the information located in the Cloud [6, 25].
- ***Datacenters*** are a set of servers that execute the required services [6, 25].
- ***Distributed servers*** are the physical servers situated in different areas. Distributed servers offer better accessibility [6].

4.2 Services of Cloud Computing

Cloud Computing can offer three important services [6] as follows:

- *Software as a Service (SaaS):* In the SaaS model, the provider remotely hosts complete applications and makes them available to customers over the Internet. These applications run on Cloud environments and are accessed over the Internet through a web browser or a client. Many SaaS applications are free to use, or at least offered through a freemium model, where entry-level features are available for free and premium features are offered for the price. SaaS applications also allow multiple end-users to share the same documents and services at the same time [24, 25].
- *Platform as a Service (PaaS)* is a software environment used to code and run applications. PaaS provides developers the online access to platform-layer resources. Applications developed using a PaaS model can be either private or free for any user on the web [24, 25].
- *Infrastructure as a Service (IaaS):* Users can store remotely their data, code, and run their applications. IaaS provides storage, processing, and network resources. IaaS also allows the clients to control the storage and the applications from anywhere at any time [24-26].

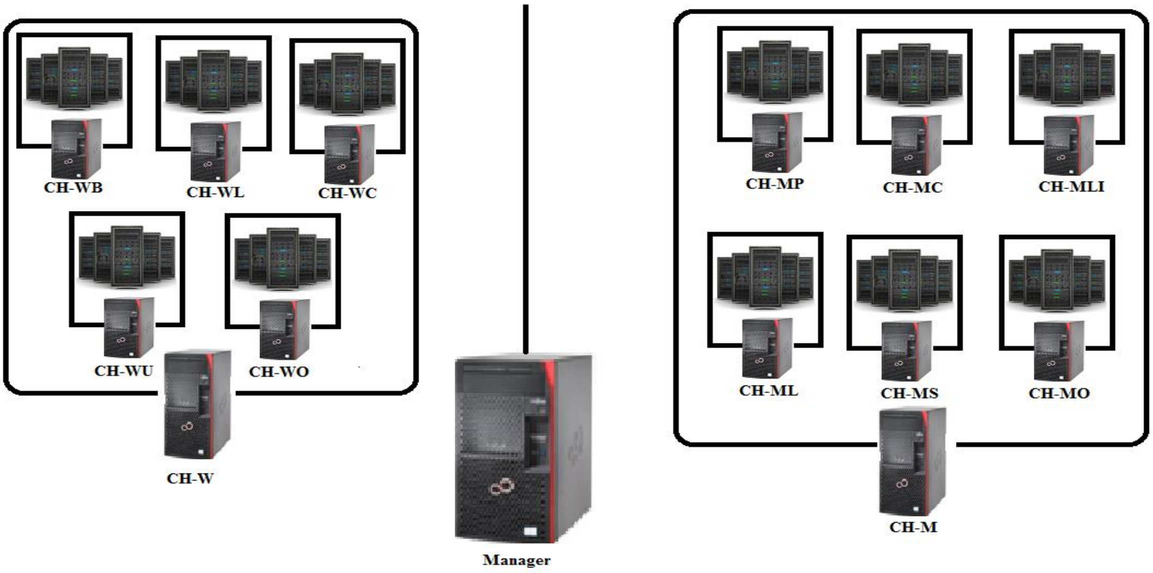

Figure 1. Clusterization of servers

5. THE PROPOSED PaaS-CBIR MODEL

This section discusses the aim of the work and details of the PaaS-CBIR architecture.

The PaaS-CBIR is proposed as a tool that can assist specialists to make an accurate and timely diagnosis. The PaaS-CBIR is a new platform as a service (PaaS) that provides software as a service with minimum latency at the online phase. During the offline phase, the PaaS-CBIR seeks to organize the database using a decision tree to ensure parallelism and collaboration.

The World Health Organization (WHO) estimates that the global cancer burden has now reached 18.1 million new cases and 9.6 million deaths in 2018 [27]. One in five men and one in six women in the world will develop cancer during their lifetime, and one in eight men and one in 11 women will die from this disease. Globally, the total number of people living with cancer within five years of diagnosis, known as five-year prevalence, is estimated at 43.8 million.

The WHO [27] also cited lung cancer as the most commonly diagnosed cancer in men (14.5% of the total for men and 8.4% for women) and the leading cause of cancer deaths in men (22.0%, or about one in five deaths). In men, lung cancer is followed by prostate cancer (incidence 13.5%) and colorectal cancer (incidence 10.9%) and, for mortality, liver cancer (10.2%) and stomach cancer (9.5%). Breast cancer is the most commonly diagnosed cancer in women (24.2%, or about one in four of the new cancer cases diagnosed in women worldwide). Breast cancer is the most commonly diagnosed cancer in 154 of the 185 countries covered by GLOBOCAN 2018. Breast cancer is also the leading cause of cancer deaths among women (15.0%), followed by lung cancer (13.8%) and colorectal cancer (9.5%), which are also the third and second most common types of cancer, respectively. Cervical cancer ranks fourth for incidence (6.6%) and mortality (7.5%).

Based on the statistics explained above, we constructed our Platform as a Service to efficiently organize our database.

5.1 PaaS-CBIR architecture

Based on statistics from the World Health Organization discussed above [27], Figure 1 describes the cloud servers according to the types of cancers.

The labels of the PaaS-CBIR architecture, described in Table 2 are based on the following hypothesis:

- During the offline phase, files are stored in DICOM format.
- All links between servers are robust.

Table 2. Notation used in PaaS-CBIR

Symbol	Description
CH-W	Cluster Head Women
CH-M	Cluster Head Men
CH-WB	Cluster Head Women Breast
CH-WL	Cluster Head Women Lung
CH-WC	Cluster Head Women Colorectal
CH-WU	Cluster Head Women Cervical
CH-WO	Cluster Head Women Other
CH-MP	Cluster Head Men Prostate
CH-ML	Cluster Head Men Lung
CH-MC	Cluster Head Men Colorectal
CH-MLI	Cluster Head Men Liver
CH-MS	Cluster Head Men Stomach
CH-MO	Cluster Head Men Other

5.2 Description of sub-cluster

Each sub-cluster represents a CBIR system. Each sub-cluster contains the database of images and the database of indexes. The sub-cluster contains only images of the specific organ. Each sub-cluster contains three agents, a color agent, a texture agent, and a shape agent. Each sub-cluster is independent of the other sub-clusters.

The function of each server is as follows:

- **Manager**
  The offline phase: Once all images are uploaded to the manager, the latter classifies images using the Decision Tree that is explained in section 6.1. Depending on the sex of the patient, the manager directs the image request either to the men sub-cluster or to the women sub-cluster.
  The online phase: The manager receives a request that contains the image, the sex, and the organ's type. Based on the received information (sex and organ), the manager orients the request to the appropriate cluster-head (CH-W

/ CH-M).

When the manager receives the resulting images of the request from either CH-W or CH-M, the manager orients these images to the appropriate user.

- **CH-W**

The offline phase: Once the CH-W receives all the pictures from the manager, the CH-W classifies them using a Decision Tree, explained in section 6.1. Depending on the type of organ, the CH-W orients the image to the appropriate sub-cluster (breast sub-cluster, lung sub-cluster, colorectal sub-cluster, cervical sub-cluster, or other sub-cluster).

The online phase: When the CH-W receives a request from the manager, based on the type of organ (breast, lung, colorectal, cervical), the CH-W forwards the request to the sub-cluster that stores images of the appropriate organ.

Once the CH-W receives the resulting images from one of (CH-WB, CH-WL, CH-WC, CH-WU, CH-WO), the CH-W delegates these images for the manager.

- **CH-M**

In the online and offline phases, the images request delegation process of the CH-M is similar to the process of the CH-W. The only difference is that the CH-M orients images toward different types of sub-clusters, including lung sub-cluster, prostate sub-cluster, colorectal sub-cluster, liver sub-cluster, stomach sub-cluster, and other sub-cluster.

- **CH-WX**

For each CH-WX (where X is breast, lung, cervical, colorectal, and other)

The offline phase: When the CH-WX receives all the images from the CH-W, the CH-WX extracts the characteristics of an image such as color, texture, and shape according to the details explained in section 3.1. The extraction introduces three demands: shape demand, color demand, and texture demand. These demands are sent respectively to agent-shape, agent-color, and agent-texture. All these agents work, in parallel, to extract the characteristics of the image. Each agent sends the results to the CH-WX. The CH-WX then aggregates the different features into a single features vector hat is stored in the index database. This treatment is applied for all the images of the database.

The online phase: When CH-WX receives a request from CH-W, it extracts the characteristics (shape, color, and texture) of the request image as explained in section 3.1. The extraction raises three requests: shape request, color request, and texture request which are sent respectively to agent-shape, agent-color, and agent-texture. All these agents perform, in parallel, the research of the corresponding request (shape, color, and texture). Each agent sends the results to the CH-WX. The CH-WX then aggregates the different results to have a features vector. The CH-WX compares the vector of the query image with each vector of the index database using Euclidean Distance as explained in section 3.2. This comparison produces a list of vectors most similar to the vector of the query image. The CH-WX looks for the appropriate images in the database and sends them to the CH-W. When the CH-W receives the selected images from the CH-WX, the CH-W then transmits the images to the manager.

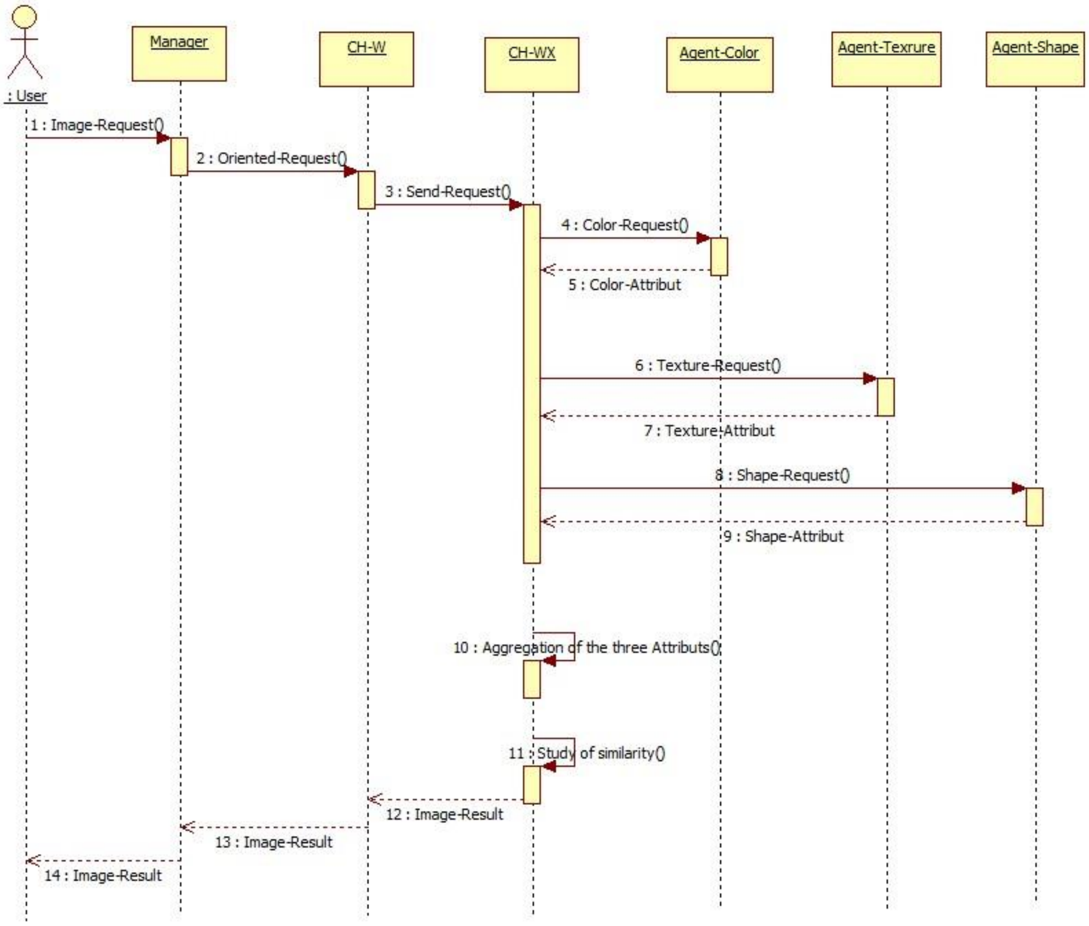

Figure 2. Sequence diagram of user request

- **CH-MX**

The CH-MX (where X is lung, prostate, colorectal, liver, stomach, and other) has the same process as the CH-WX.

The sequence diagram of a user request (e.g., CH-WX) is presented in Figure 2.

6. PROPOSED METHODOLOGY

Color, texture, and shape are the primary features of an image. These are extracted and put into a feature vector. When the user submits a request image (online phase), the same features extraction principle as is used in the offline phase is applied to this request. The Euclidean Distance as explained in section 3.2 is used to compare the similarity between the feature vector of the request image and the feature vectors of the database images. Our proposal goes through offline and online phases.

6.1 Offline phase

Step 1: Collection of the images in the database.

All images are stored with DICOM format, see Figure 3a. Each image is treated at the manager level to be oriented to the appropriate server (CH-M/CH-W), see Figure 3b.

Step 2: Building a decision tree

A decision tree consists of a set of rules that divides a case population (a patient database, in our case) into homogeneous groups. Each rule associates conjunction of tests on the attributes of a case to a group (for example: "if sex = male, then the case belongs to the group ‘men’"). Figure 4 describes these rules organized in the form of a tree whose structure is as follows.

- Each non-terminal node corresponds to a test on a descriptor (for example "Sex =?").
- Each arc corresponds to a response to a test (for example "male")
- Each sheet corresponds to a group of cases that provided the same answer to all the tests of a rule (example: "men with lung cancer").

To automatically build a decision tree, we need to look for the most discriminating attributes among all available attributes (images and contextual attributes, in our case) and build homogeneous case groups from tests on these attributes. The construction mechanism illustrated in Figure 3, is based on supervised learning. To start construction, we must have several classified examples. At the initialization of learning, the tree is simply a leaf, gathering the entire population as seen in Figure 3a. Recursively; we then divide each leaf F of the tree under construction. We are looking for the descriptor d most discriminant within the population P grouped in F. P is then distributed between new child nodes, one for each possible response to the test on d see Figure 3(b, c, d).

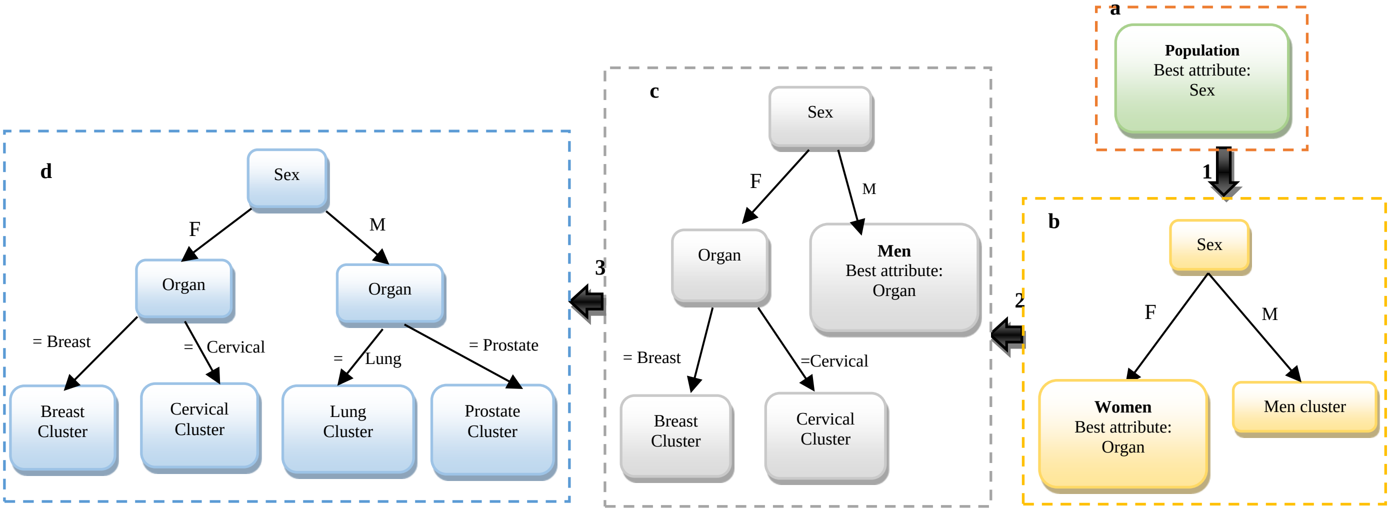

Figure 3. Learning a decision tree

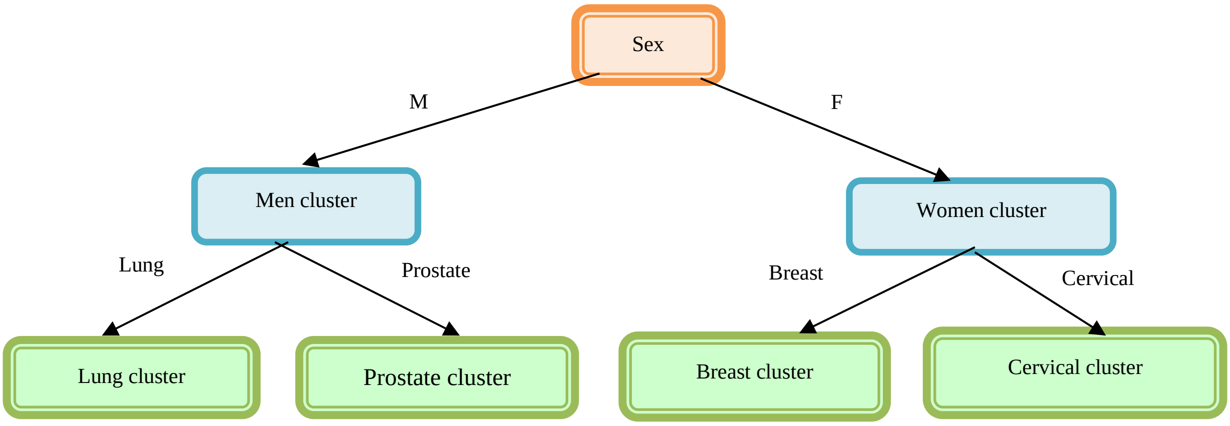

Figure 4. Example of a decision tree structure

In our proposal, each cluster represents a server (or even a set of servers). It is proposed that each cluster will have three agents to do the treatment; one agent responsible for the color, one agent responsible for the texture, and one agent responsible for the shape. These three agents are used in the extraction of the characteristics (features). Our architecture organizes 11 sub-clusters, and each sub-cluster deploys three agents which allow more rapid extraction of the characteristics (features) of the images compared to the classical architecture.

Step 3: The objective of this step is to construct a feature vector of each image. The image feature vector contains digital fields.

The choice of attributes is strongly dependent on the images of the database. Thus, attributes that give excellent results on one set of database images may give poor results on another set. There are no universal attributes that work properly on any image database. The attributes are context-sensitive. The general principle of our proposition is independent of the choice of attributes. We will propose a set of attributes to validate and illustrate our proposal, but the proposal will be as helpful using families of attributes different from those we use.

- ***Color feature extraction:*** The color is extracted using two attributes, such as the color histograms and the color moments. Color moments were divided into 3 low-order moments which are the mean of c μ_c, the standard deviation σ_c, and the skewness θ_c, as explained in section 3.1.1.
- ***Texture feature extraction:*** The texture is extracted using the co-occurrence matrix of Haralick. Only the four most appropriate characteristics are widely used (energy, entropy, the contrast, and the Inverse Difference Moment (IDM)) as explained in section 3.1.2.
- ***Shape feature extraction:*** The shape feature is extracted using the seven invariant moments in the rotation of Hu as explained in section 3.1.3.

The extracted features (color, texture and shape) are stored in the feature vector.

6.2 Online phase

In the first phase, the user submits an image request that contains an image, sex, and type of organ.

To orient the image request to the appropriate cluster, we follow the same principle of the decision tree mapping explained in the offline phase. The extraction of the features (color, texture, and shape) of the image request follows the same process as the extraction of image features from the database in the offline phase.

In the second phase, the Euclidean distance is used to compare the similarity between the feature vector of the request image and the feature vectors of the index database. The images are then ranked based on the similarity.

7. SYSTEM RECOVERY IN PAAS-CBIR

To assure image availability in PaaS-CBIR, we propose a mechanism based on the Time-Triggered Protocol (TTC/P) to ensure monitoring, backup, and recovery services for PaaS-CBIR.

In TTC/P, the servers are authorized to broadcast data according to a Time Division Multiple Access (TDMA) scheme. TTP splits a TDMA round into time slots (TS). It allocates each server in the communication system to one slot during which it broadcasts its data [8, 28]. The TTC/P is very important because it ensures safety-critical, fault-tolerant high-speed networks.

Assumptions:

- The server clocks are always synchronized.
- At most, one server (for an organ) will fail at a time.
- Only transient failures can happen.
- There are no links failures.
- Transient failure can only happen at the receiver server.

7.1 System monitoring

If one server fails, it could affect the time required to recover the images stored on that server. For that reason, an image replication is necessary. We proposed a solution that duplicates the content of each server (i) to a server (i+1). Thus, each server (i+1) contains the images of its neighbor server (i).

Each CH-X (where X is Men or Women), organizes the IP address of all servers in a round-robin order. Once the CH-X creates the list, it broadcasts it to all servers in its cluster. Based on this list, each server can communicate with its neighbor in the list (the last server's neighbor is the first server). A server detects its neighbor’s failure state by sending a heart-beat message to its neighbor.

Figure 5 shows that each TDMA round is split into N time slots, where N is the number of organ servers. Each server is assigned a fix and periodic duration (time slot *i mod N*) of unicast during which it can check its neighbor's health state. If the sender (i) receives an acknowledgment from its neighbor, then the sender deduces that its neighbor works properly; otherwise, it deduces that its neighbor is out of order. If server (i+1) is out of order, server (i) immediately informs the CH-X about server (i+1)’s failure.

Compared to Figure 5, we added the principle of Faults Tolerance schematized in Figure 6.

7.2 Data replication

On receipt of each new image (e.g., breast image), each server (e.g., breast server) duplicates the new received data on its neighbor (e.g., lung server) according to a round-robin order. This happens during the offline phase. Note that the original images of each server (server(i+1)) are stored separately from the replicated images of the server’s neighbor (server(i)).

7.3 Data recovery

When the CH-X receives an output request (i) stored on server(i), CH-X sends a hello message to server(i), to check whether it works properly or not. If server(i) replies to the CH-X with an acknowledgment, CH-X assigns it to the output request; otherwise, CH-X forwards the request to the server(i+1) (see Figure 6).

7.4 Data recovery complexity

The complexity is N x M, where N is the number of servers and M is the number of images on each server.

In PaaS-CBIR, N=1 and M=100. However, in the classical architecture N=11 and M=100

The complexity in PaaS-CBIR is 100, and in the classical architecture, 1100 units of time. As shown in Figure 7, the complexity is proportional to the number of images in each server.

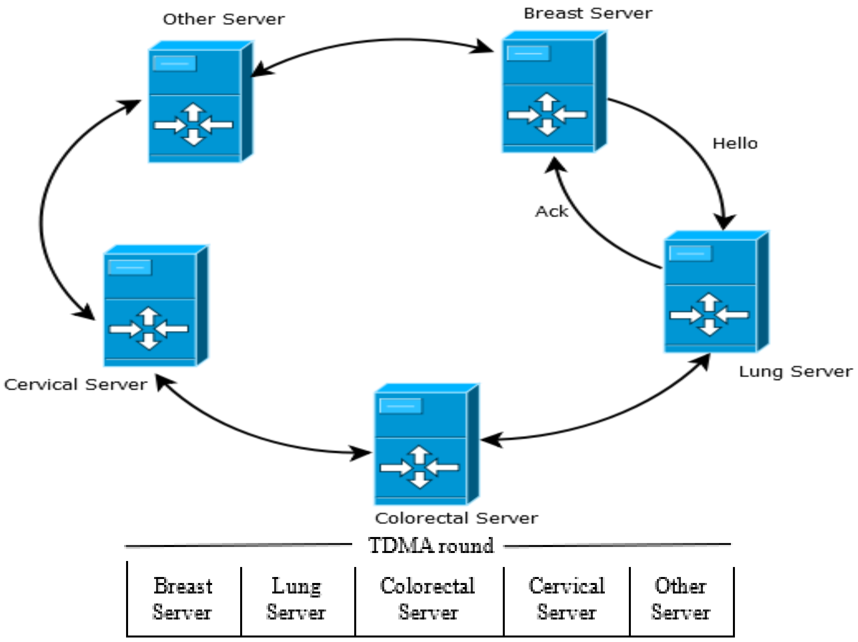

Figure 5. System monitoring

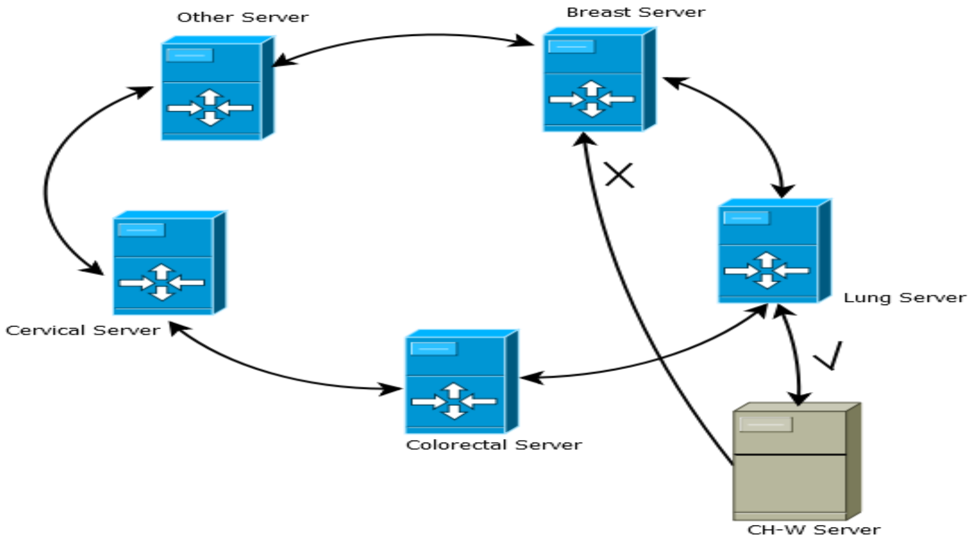

Figure 6. Recovery of the replicated image version

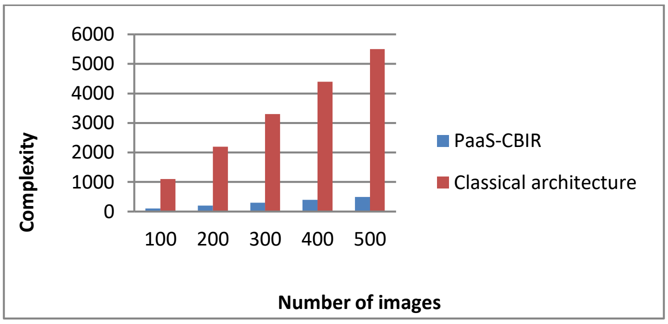

Figure 7. Complexity of PaaS-CBR & Classical architecture

8. PaaS-CBIR FORMAL EVALUATION

This section presents a formal evaluation of the PaaS-CBIR system.

Theorem 1: The PaaS-CBIR ensures parallel processing of eleven different types of requests at the same time.

Proof: PaaS-CBIR is based on the indexed tree that separates the database according to two sub-clusters (Men or Women). The sub-cluster that stores the men's images is made of six servers. Each server treats each type of organ independently. The sub-cluster that holds the women's database contains five servers. These servers work in parallel, which allows executing eleven different requests at the same time.

Theorem 2: PaaS-CBIR out-performs classical architecture by 89% in terms of response time.

Proof: *N*: total number of images

a: comparison time between the request image and an image stored in the database.

b_1: In the classical architecture, the extraction time of the characteristics of the request image is equal to extraction_time shape + extraction_time color + extraction_time texture.

b_2: In PaaS-CBIR, the extraction time of the characteristics of the request image = Max (extraction_time_shape, extraction_time_color, extraction_time_texture). We take the maximum value because three agents are executed in parallel. Therefore, $b_1 > b_2$

In the classical platform:

Response time_C=$\sum communication_{time} + b_1 + (N * a\ time)$

Communication$_{time}$=number of servers $* \alpha$

We consider α as the time required transmitting information from one server to another and *time* a unit of time, thus:

Response time_C=$11 * \alpha + b_1 + (N * a\ time)$

However, in PaaS-CBIR, the research will be at level of $\frac{N}{2} = X$ (according to the sex).

In the worst case of PaaS-CBIR, the research will be at the level of $\frac{X}{5} = X1$ (according to the organ).

Response time_PaaS-CBIR=$\sum communication_{time} + b_2 + X1 * a\ time$

$$X1 = \frac{X}{5} = \frac{\frac{N}{2}}{5} = \frac{N}{10}$$

Response time_PaaS-CBIR=$\sum communication_{time} + b_2 + (\frac{1}{10} N * a\ time)$

Communication$_{time}$= number of servers * α

α: time required to transmit information from one server to another, *time* is a unit of time.

***Response time_PaaS-CBIR* is:** $2 * \alpha + b_2 + (\frac{1}{10}\ N * a\ time)$

Theorem 3: PaaS-CBIR provides accurate results to the specialists using the database. It completely avoids the semantic fault problem.

Proof: Research is oriented toward a subset of the database that contains only relevant images pre-selected by sex and then, by organ, which limits the type of image (e.g., Breast). Because we are searching for a set of relevant images, only images related to the image request will be returned, thus the semantic fault is avoided.

Theorem 4: PaaS-CBIR decreases the amount of overhead, defined as the movement of tasks during inter-process and intra-processor communication, caused by the unbalanced delegation of requests.

Proof: In classical architecture, suppose that we have K servers and N requests. Each request should be verified sequentially until it will be found. In the worst case, where (K-1) servers do not contain the required request, there is a high probability that the request will be satisfied on the Kth server. Thus, (K-1) servers are uselessly occupied, which increases the overhead of these servers. PaaS-CBIR delegates the request directly to the appropriate server which decreases the overhead rate.

Theorem 5: PaaS-CBIR manipulates Big Data with high efficiency.

Proof: PaaS-CBIR provides unlimited storage processing capacity, which allows handling a myriad number of images. This utility satisfies the Volume characteristic. It also enables the incorporation of data from distinct sources and formats associated with different images. This benefit contents the Variety characteristic. PaaS-CBIR supplies fast requests response due to the distributed architecture and the parallelism designed in the proposal gratifies the third characteristic of Big Data, which is Velocity.

Theorem 6: PaaS-CIBR ensures image availability within minimum latency in case of a server failure.

Proof: If CH-X receives a request from the Manager, and CH-X detects that server(i) which stores the required images is out of order, then CH-X directly orients the image request to the server(i)'s neighbor (server(i+1)). Thus, it recovers the replicated versions of the similar images without fetching all the servers in the Cloud.

9. CONCLUSION

Constant technological advances in the field of archiving digital data allow us today to have access to a quantity of information unequaled in history. All areas of human activity are affected, and the problems are not simply the volumes of information archived, but also about the use of these data, the search for information relevant for a given use. This is particularly true in the medical field, where more and more information relating to patients, pathologies, and medical knowledge is recorded, archived in databases, and, in theory, of value for training and diagnosis.

In our work, we are interested in databases containing DICOM files that contain both digital images and semantic information, which requires the manipulation of big data. The approach that we have explored is based on Content Based Image Retrieval (CBIR).

Specifically, the contribution of this work is the proposition of an efficient Platform as a Service (PaaS) Model on a public Cloud for CBIR System. The proposed platform is fast because most of the computation time is spent during the offline phase.

PaaS-CBIR has been proposed to aid in medical diagnosis. PaaS-CBIR manipulates semantic information, rather than single images. By using the decision trees, the proposed architecture makes it possible to improve the calculation time for a medical diagnosis aid system. PaaS-CBIR can be implemented in a practical way to support the medical diagnosis. PaaS-CBIR handles big data with great efficiency. PaaS –CBIR creates intelligent links between servers that ensure data availability with a minimum latency time.

As future work, we plan to use other methods to extract image characteristics. We also plan to propose a contribution that enables the dynamic storage of our PaaS-CBIR system.

REFERENCES

[1] Gwénolé, Q. (2008). Indexation et fusion multimodale pour la recherche d'information par le contenu. Application aux bases de données d'images médicales. Thèse de doctorat, Télécom Bretagne.

[2] Manoj, K., Manglem, S. (2016). CBMIR: Content based medical image retrieval system using texture and intensity for eye images. International Journal of Scientific & Engineering Research, 7(9).

[3] Dimitrovski, I., Guguljanov, P., Loskovska, S. (2009). Implementation of web-based medical image retrieval system in oracle. 2009 2nd International Conference on Adaptive Science & Technology (ICAST), Accra, Ghana, pp. 192-197. https://doi.org/10.1109/ICASTECH.2009.5409726

[4] Ramamurthy, B., Reddy, K.R. (2013). CBMIR: Content based medical image retrieval using shape and texture content. Advance in Modeling B: AMSE Press, France, 56(2): 84-95.

[5] Trojacanec, K., Dimitrovski, I., Loskovska, S. (2009). Content based image retrieval in medical applications: an improvement of the two-level architecture. In IEEE EUROCON 2009. IEEE, St.-Petersburg, Russia, pp. 118-121. https://doi.org/10.1109/EURCON.2009.5167614

[6] Hammoudi, S., Aliouat, Z., Harous, S. (2018). Challenges and research directions for internet of things. Telecommunication Systems, 67(2): 367-385. https://doi.org/10.1007/s11235-017-0343-y

[7] Kaur, P., Sharma, M., Mittal, M. (2018). Big data and machine learning based secure healthcare framework. Procedia Computer Science, 132: 1049-1059. https://doi.org/10.1016/j.procs.2018.05.020

[8] Hammoudi, S., Aliouat, Z., Harous, S. (2018). A new infrastructure as a service for IoT-Cloud. 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, pp. 786-792. https://doi.org/10.1109/IWCMC.2018.8450503

[9] Mohana, R.M., Rama Mohan Reddy, A.(2015). CCBIR: A cloud based implementation of content based image retrieval. WSEAS Transactions on Computers, 14. http://www.wseas.org/multimedia/journals/computers/2015/a685705-840.pdf

[10] Batra, M., Agrawal, R. (2018). Comparative Analysis of Decision Tree Algorithms. Advances in Intelligent Systems and Computing, vol 652. Springer, Singapore, pp. 31-36. https://doi.org/10.1007/978-981-10-6747-1_4

[11] Rokach, L., Maimon, O. (2005). Decision Trees. In: Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA, 165-192. https://doi.org/10.1007/0-387-25465-X_9

[12] Yuan, L., Chen, H., Gong, J. (2018). Classifications based decision tree and random forests for fanjing mountains' tea. IOP Conference Series: Materials Science and Engineering, 394(5): 052002. https://doi.org/10.1088/1757-899X/394/5/052002

[13] Kusrini, K., Iskandar, M.D., Wibowo, F.W. (2016). Multi features content-based image retrieval using clustering and decision tree algorithm. Telkomnika, 14(4): 1480. http://dx.doi.org/10.12928/telkomnika.v14i4.4646

[14] Meena, M., Mallya, S., Talati, S. (2018). Effectiveness of SaaS cloud model for retrieving images from CBIR system. in Int. Conf. on Innovative and Advanced Technologies in Engineering, 13: 70-74.

[15] Meena, M., Bharadi, V.A. (2016). Hybrid wavelet based CBIR system using software as a service (SaaS) model on public cloud. Procedia Computer Science, 79: 278-286. https://doi.org/10.1016/j.procs.2016.03.036

[16] Paiz-Reyes, E., Nunes-De-Lima, N., Yildirim-Yayilgan, S. (2018). GIF Image Retrieval in Cloud Computing Environment. In: Campilho A., Karray F., ter Haar Romeny B. (eds) Image Analysis and Recognition. ICIAR 2018. Lecture Notes in Computer Science, vol 10882. Springer, Cham, 261-268. https://doi.org/10.1007/978-3-319-93000-8_30

[17] Mahmoudi, S.A., Belarbi, M.A., Dadi, E.W., Mahmoudi, S., Benjelloun, M. (2019). Cloud-based image retrieval using GPU platforms. Computers, 8(2): 48. https://doi.org/10.3390/computers8020048

[18] Müller, H., Mochoux, N., Bandon, D., Geissbuhler, A. (2004). A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. International Journal of Medical Informatics, 73(1): 1-23. https://doi.org/10.1016/j.ijmedinf.2003.11.024

[19] Swain, M.J., Ballard, D.H. (1991). Color indexing. International Journal of Computer Vision, 7(1): 11-32. https://doi.org/10.1007/BF00130487

[20] Dubey, R.S., Choubey, R., Bhattacharjee, J. (2010). Multi feature content based image retrieval. International Journal on Computer Science and Engineering, 2(6): 2145-2149.

[21] Haralick, R.M., Shanmugam, K., Dinstein, I. (1973). Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(6): 610-621. https://doi.org/10.1109/TSMC.1973.4309314

[22] Hu, M.K. 91962). Visual pattern recognition by moment invariants. IRE Transactions on Information Theory, 8(2): 179-187. https://doi.org/10.1109/TIT.1962.1057692

[23] Jobay, R., Sleit, A. (2014). Quantum inspired shape representation for content based image retrieval. Journal of Signal and Information Processing, 5(2): 54-62. http://dx.doi.org/10.4236/jsip.2014.52008

[24] Botta, A., de Donato, W., Persico, V., Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. Future Generation Computer Systems, 56: 684-700. https://doi.org/10.1016/j.future.2015.09.021

[25] Amin, Z., Singh, H., Sethi, N. (2015). Review on fault tolerance techniques in cloud computing. Int. Journal of Computer Applications, 116(18): 11-17. https://doi.org/10.5120/20435-2768

[26] Zhang, Q., Cheng, L., Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1): 7-18. https://doi.org/10.1007/s13174-010-0007-6

[27] World Health Organization, https://www.who.int, accessed on Jan. 10, 2019.

[28] Maier, R. (2002). Event-triggered communication on top of time-triggered architecture. Proceedings. The 21st Digital Avionics Systems Conference, Irvine, CA, USA, pp. 13C5-13C5. https://doi.org/10.1109/DASC.2002.1053011