International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 11 (2019) pp. 2636-2646

Design and Development of an efficient CBIR system using Hadoop to analyze large scale MRI images dataset for early disease diagnosis

[a]Harinder Singh, [b]Dr. Kulvinder Singh Mann
[a]Research Scholar, IKGPTU, Jalandhar, Punjab, India.
[b]Professor, Dept. of IT, GNDEC, Ludhiana, Punjab, India.

Abstract
An efficient Content Based Image Retrieval (CBIR) System using Hadoop is put forward to analyze large scale Magnetic Resonance Imaging (MRI) images dataset for early disease diagnosis. The proposed system helps to enhance the retrieval efficiency for large medical images database against the drawbacks of traditional File Management based single node retrieval systems. The local ternary co-occurrence patterns (LTCoP) method is used for extraction of texture features of MRI images. The extracted LTCoP features of each image are stored in Hadoop Distributed File system (HDFS) based HBase by using the MapReduce programming model of Hadoop. Then query MRI image LTCoP features are matched with features in feature vector database by using the Map function, while the calculated outcomes of each Map function are received by the Reduce function and results are ranked as per the size of similarity to obtain the best retrieval results. The proposed method is evaluated on massive MRI images dataset. The experimental results demonstrate that the method proposed in our work outperforms state-of-the-art methods in terms of storage time and average retrieval time.
Keywords: Content Based Image Retrieval (CBIR), Hadoop, Magnetic Resonance Imaging (MRI) dataset, local ternary co-occurrence patterns (LTCoP), Hadoop Distributed File System (HDFS), MapReduce, feature vector database.

1. INTRODUCTION

The development of digital imaging equipments leads to produce large amount of medical images. In 2002, more than 12000 images a day were produced by the Department of Radiology of a hospital in Geneva [1]. Medical images like magnetic resonance imaging (MRI), X-rays, computed tomography (CT), etc. are the important tools to help doctors for disease diagnosis. The effective management and fast access of such huge amount of medical images becomes an important field of research in the modern era. Traditional image retrieval systems retrieve the marked images using keywords. In these systems, the images are annotated by text. Due to the inaccuracy of manual annotation of images, the traditional systems are not capable to meet the needs for large scale images retrieval [2]. So the effective management of such large scale medical images becomes a problem for medical professionals to provide clinical diagnosis. The solution for this problem is Content Based Image Retrieval (CBIR) system which is widely applied in medical diagnosis and supports the process of clinical decision making [1].

CBIR is defined as the process of retrieving most relevant images to a given query image from a massive database of images using the visual features of images [2].Visual features can be extracted from image content such as texture, shape or color. CBIR system has three steps; First, selection of suitable image features; Second, selection of the effective method for feature extraction; third, use of the effective method for feature matching.

1.1. Motivation

CBIR is a data intensive computing process [2]. For large-scale image datasets, computational efficiency and accuracy have become the key challenges [3]. In generic image retrieval problems, the current methods for large scale datasets have achieved successes, but computational efficiency is still a challenge for massive medical image datasets [4].The main problems of massive medical image databases are: a) the inadequacy of traditional file management systems to handle massive databases [5]; b) the difficulty to analyze massive medical images as the large size of medical images, anatomic interactions and differences among different diseases produce a complex analysis [3]. To tackle the problem of retrieval for massive medical images, the direct applying traditional methods of CBIR may not appropriate [3].So the exponential growth of medical imaging datasets necessitates a paradigm shift in the way the data is managed and processed [5]. The retrieval of images from large-scale medical imaging datasets is very time-consuming [6].Therefore an efficient storage and retrieval model is required. This has motivated us to provide an efficient CBIR system to analyze massive MRI images dataset.

1.2. Related Work

Researchers have explored various methods for CBIR systems like Gabor filters [7] and discrete wavelet transform [8] for texture analysis. The further improvement to discrete wavelet transform was made in rotated complex wavelet filter [9], dual tree complex wavelet transform [10] and rotated wavelet filter [11] for texture analysis. In context of image retrieval, the local patterns are widely used for texture analysis. The most popular method for texture classification is Local Binary Pattern (LBP) [12]. The further improvements to LBP method was done in number of methods such as completed LBP [13], Dominant LBP [14], Multi scale block LBP (MB-LBP) [15] and Local Derivative Pattern (LDP) [16].The concept of LBP based on moment is proposed in [17]. Further, the edge information was used to extract the feature in Directional Local Extrema Pattern (DLEP) [18]. In this method, four directions in the image are utilized and the further enhancement was done in [19, 20]. Noise is a limitation of LBP. To reduce the noise of LBP, NR-LBP [21] and RLBP [22] are used. The further improvement to LBP is Local Ternary Pattern (LTP) [23] and is insensitive to noise. The further improvement to LTP is Improved LTP [24]. The concept of Local Tetra Pattern (LTrP) is presented in [25]. The Local Diagonal Extrema Pattern (LDEP) is put forward for retrieval of images of CT in [26].

The Local maximum edge binary pattern (LMEBP) is presented for pattern calculation in [27].Local ternary co-occurrence patterns (LTCoP) is presented in [28] for MRI and CT image retrieval. Co-occurrence of adjacent Sparse local ternary patterns (CoALTP) is proposed in [29] for texture classification. Spherical symmetric 3D Local Ternary Pattern (SS-3D-LTP) is presented in [30]. Local Mesh Patterns (LMeP) is proposed in [31] and Peak Valley Edge pattern is proposed in [32]. Local Mesh Peak Valley Edge Pattern combines these two methods in [33] for the retrieval of MRI and CT images. Center Symmetric Local binary co-occurrence pattern is proposed in [34]. Moreover, GLCM is used to extract feature vector in this method. Local directional number pattern (LDNP) is proposed in [35]. Local gradient hexa pattern (LGHP) is presented in [36]. The local neighborhood difference pattern (LNDP) is proposed in [37]. In this method, the combination of LNDP and LBP is applied for texture classification. The local directional ternary pattern (LDTP) is presented in [38]. The proposed LDTP is formed by the combination of LTP three level descriptions and the LDP directional features. Local derivative radial patterns (LDRP) is proposed in [39] for texture classification. Local concave-and-convex microstructure patterns (LCCMSP) is proposed in [40]. Repulsive and attractive local binary gradient Contours (RALBGC) is proposed in [41]. 3D Local Ternary Co-occurrence Patterns (3D-LTCOP) is presented in [42]. In this paper, 3D-LTCOP in combination with 3D-LTP is also applied for image retrieval purpose. Local Neighborhood Intensity Pattern (LNIP) is proposed in [43] for pattern calculation.

Current perspectives, challenges and solutions for analytics of big data in health have been introduced in [44]. An overview of big data has been presented in [45]. Challenges for big data analysis have been described in [46]. A survey of big data in health informatics has been elaborated in [47]. Review and open research issues for the big data rise on cloud computing has been discussed in [48]. An overview of big data challenges and technologies has been elaborated in [49]. In [50], an overview of machine learning tools in big data has been presented. A comprehensive review for analytics of big data has been provided in [51] and in [52], an overview on multimedia big data analytics has been presented.

1.3. Main Contribution of this work

An efficient CBIR system using Hadoop is presented to analyze large scale MRI images dataset for early disease diagnosis. The proposed CBIR system adopts Hadoop platform and uses LTCoPs descriptor for extraction of image features. The proposed method and traditional CBIR methods are evaluated on massive MRI images dataset and the experimental results obtained show that the proposed method performs better than state-of-the-art CBIR methods in terms of storage time and average retrieval time. The presented work is organized as: Section 1 includes introduction, related work and main contribution. Materials and Methods are elaborated in Section 2. In section 3, proposed system framework is presented. Experimental results are given in section 4 and finally, conclusion and future scope is provided in section 5.

2. MATERIALS AND METHODS USED

2.1 Hadoop Platform

Hadoop Platform is open source, reliable and scalable software platform that supports distributed computing for large dataset [2]. The following modules that form the base for Apache Hadoop are as follows:

HDFS: HDFS is designed to handle massive datasets [5]. HDFS provides the management and storage of data in the Hadoop cluster. HDFS structure makes up of a master node (NameNode), multiple slave nodes (DataNodes). The NameNode provides the management of the namespace of file system and reading of files by multiple clients. The DataNode is responsible for data storage of its nodes and handles read/write requests of clients to the file system. HDFS decomposes files into small data blocks and store these small data blocks in different DataNodes dispersedly. Each data block can be stored and copied in different DataNode. Due to this property, HDFS achieves high throughput and high fault tolerance.

MapReduce: MapReduce is a batch-based distributed programming model for computing massive amount of dataset [51]. It is based on parallel computing, in which a task is broken into subtasks. The different subtasks are assigned to different resource nodes by the system using appropriate strategies. After finishing all the subtasks, the processing of larger task is finished and result is delivered to the user. The MapReduce phase is splitted into the Map phase and the Reduce phase. In the Map phase, the data assigned in each subtask is calculated by each Map function and then the resultant data is mapped to the Reduce function as per <key, value> outcome by Map. During Reduce phase, the Reduce function performs next level processing to get the output results.

The Hadoop platform is also used for the collection of other software packages like Hive, Pig, Hbase, Spark, Phoenix, Zookeeper, Falume, Sqoop, Cloudera Impala, Oozie and Storm [5].

2.2 Feature Extraction by LTCoPs

The idea of LTCoP is proposed in [28] for MRI and CT image retrieval. In proposed LTCoP, for the center pixel (g_c), first-order derivatives are computed as follows:

$$\tilde{I}_{P,R}(g_i) = I_{P,R}(g_i) - I_{P,R}(g_c); i = 1,2,....,P \quad (1)$$

$$\tilde{I}_{P,R+1}(g_i) = I_{P,R+1}(g_i) - I_{P,R}(g_i); i = 1,2,..,P \quad (2)$$

After calculation, these first order derivatives are coded as follows:

$$I^1{}_{P,R}(g_i) = \tilde{f}_1(\tilde{I}_{P,R}(g_i)) \quad (3)$$

$$I^1{}_{P,R+1}(g_i) = \tilde{f}_1(\tilde{I}_{P,R+1}(g_i)) \quad (4)$$

where $$\tilde{f}_1(I(g_i), I(g_c), th) = \begin{cases} 1, & I(g_i) \geq I(g_c) + th \\ 0, & |I(g_i) - I(g_c)| < th \\ 2, & I(g_i) \leq I(g_c) - th \end{cases} \quad (5)$$

Here th is the user specified threshold, $I(g_c)$ represents the Center pixel (g_c) intensity and $I(g_i)$ represents the intensity of neighboring pixels to centre pixel (g_c).

The $I^1_{P,R}(g_i)$ and $I^1_{P,R+1}(g_i)$ form ternary codes. These ternary values are used to calculate LTCoP as follows:

$$\text{LTCoP} = \begin{bmatrix} f_2(I^1_{P,R}(g_1), I^1_{P,R+1}(g_1)), \\ f_2(I^1_{P,R}(g_2), I^1_{P,R+1}(g_2)), \\ ...., f_2(I^1_{P,R}(g_P), I^1_{P,R+1}(g_P)) \end{bmatrix} \quad (6)$$

$$\text{and } f_2(x,y) = \begin{cases} 1 & if\ x = y = 1 \\ 2 & if\ x = y = 2 \\ 0 & else \end{cases} \quad (7)$$

LTCoP form ternary patterns (0, 1, 2) and two binary patterns are formed from these ternary patterns by using the idea of LTP [23].The histograms constructed from these two binary patterns are then concatenated for whole the image to form feature vector.

More details can be found in [28].

2.3. Similarity measure

In database, each image is represented by a feature vector as $f_{DB_j} = (f_{DB_{j1}}, f_{DB_{j2}}, ..., f_{DB_{jLg}})$; where $j = 1,2,.....,|DB|$. After the feature extraction, the feature vector of query image Q is represented as $f_Q = (f_{Q_1}, f_{Q_2}, ..., f_{Q_{Lg}})$. The objective is to retrieve N images that most resemble with query image. For this, distance is measured between query image and image in database |DB|. The Euclidean Distance is used to calculate similarity measure as:

$$\text{ED (Q, DB)} = \sqrt{\sum_{i=1}^{Lg} (f_{DB_{ji}} - f_{Q_i})^2} \quad (8)$$

The Normalized Euclidean Distance can be calculated as:

$$\text{Normalized-ED(Q, DB)} = \frac{\sqrt{\sum_{i=1}^{Lg} (f_{DB_{ji}} - f_{Q_i})^2}}{\sqrt{\sum_{i=1}^{Lg} (f_{DB_{ji}})^2} + \sqrt{\sum_{i=1}^{Lg} (f_{Q_i})^2}} \quad (9)$$

where $f_{DB_{ji}}$ represents the i^{th} feature of j^{th} image in the database $|DB|$ and f_{Q_i} represents i^{th} query image feature. The value of Normalized Euclidean Distance ranges between [0, 1]. Minimum Normalized Euclidean Distance shows the maximum match between two images.

2.4. Grid Set-up

For the proposed system, grid has been setup using multi node cluster in which one node act as master (NameNode) and other three nodes act as slaves (DataNodes). Physical connections need to be established among all nodes. The hardware configuration of NameNode and DataNodes in the proposed system is shown in Table1:

Table1: Hardware configuration of NameNode and DataNodes in the proposed system.

Node	Processor	RAM	Hard Disk
NameNode	Core i7-8700K, 3.7GHz	32GB	2TB
DataNode	Core i3-8100, 3.6GHz	16GB	1TB
DataNode	Core i3-8100, 3.6GHz	16GB	1TB
DataNode	Core i3-8100, 3.6GHz	16GB	1TB

The software requirements for proposed system include Hadoop, open source OS Linux and java 1.7.

3. PROPOSED SYSTEM FRAMEWORK

The block diagram of the proposed system has been shown in Fig. 1 and the method for the same is given as below:

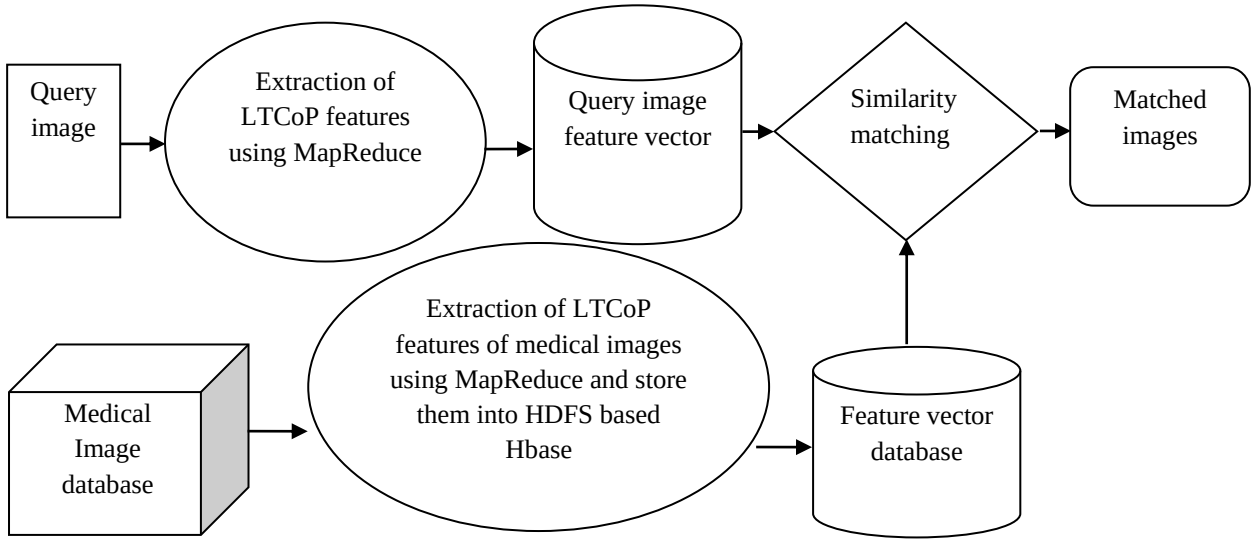

Fig.1: Block diagram of the proposed CBIR system

I.) Method for Storage of Medical Images

Image storage forms the base for the retrieval of images. In this proposed system, theLTCoP descriptor andMapReduce is used to upload the images into HDFS based HBase. The steps for image storage are as follows-:

a) During the Map Phase, the Map task is used to take an image from HDFS and extract texture features of image using LTCoP descriptor.
b) During the Reduce Phase, LTCoP features extracted by Map task are stored in Hbase. HBase contains the image ID, pathname and LTCoP features of the image. The Storage process is shown in fig. 2.

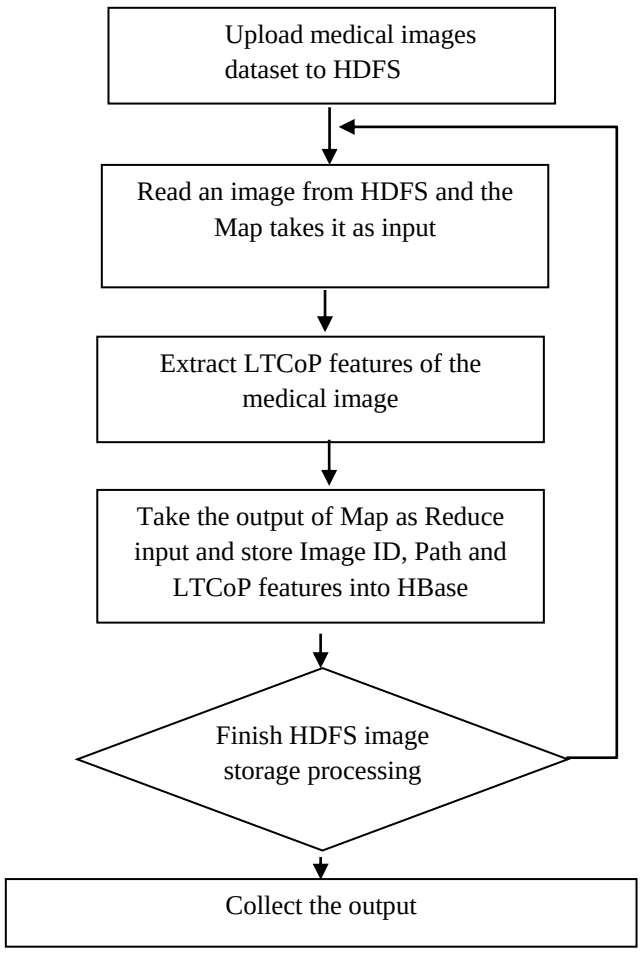

Fig.2: Storage Process for Medical Images

II.) Method for medical Image Retrieval

In this phase, as the dataset of Hbase is very big, so it becomes very time consuming to search the entire HBase. To reduce the image retrieval time, the MapReduce model is applied to perform the parallel computing. The steps for retrieval of images based on MapReduce are as follows:

a) With the submission of query images, store the query images into HDFS.
b) During the Map Phase, extract the LTCoP feature of the query images stored in HDFS and store the extracted LTCoP features into HDFS.
c) During the Map Phase, perform the similarity matching between images features in Hbase and query image features. The outcome of each Map task is <key, value> pair of < similarity, image ID>. In this similarity matching process, select the <key, value> pairs of <similarity, ImageID>, that have the similarity in the predefined range of similarity value and input <key, value> pairs into the Reducers.
d) During the Reduce phase, obtain all selected <key, value> pairs of < similarity, imageID> and perform similarity sorting of theses <key, value> pairs and first N <key, values> are written in HDFS.
e) Output image ID of written N <key, values>in HDFS, and the user obtains the final results of image retrieval.

The specific process for medical image retrieval is shown in fig. 3.

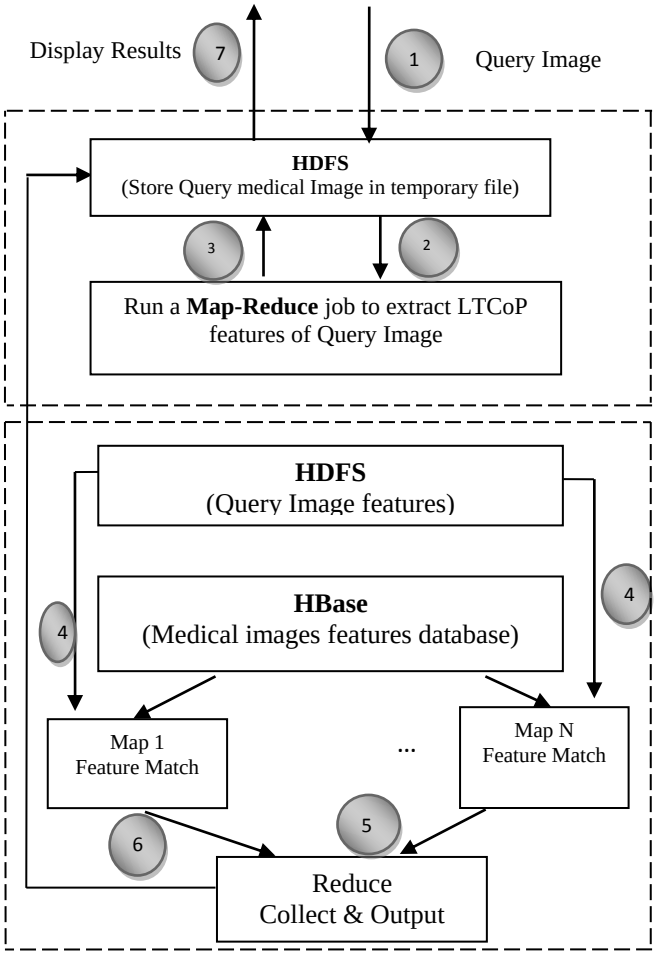

Fig.3: Medical images retrieval process

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Evaluation criteria

To show the efficiency of proposed method, it is tested on large scale MRI images database. The performance is evaluated on the basis of CPU Usage Rate, Storage Time, Average Retrieval Time and Precision. The superiority of the method proposed in our work is verified by comparing it with state of the art CBIR methods in terms of computation efficiency i.e. Storage Time and Average Retrieval Time. We

can define CPU Usage Rate, Storage Time, Average Retrieval Time and Precision as follows:

i.) **CPU Usage Rate:** CPU Usage Rate is the percentage of utilization of computing resources for a given task. It is used to measure the system performance under different circumstances.

ii.) **Storage Time:** The Storage efficiency is measured in Storage Time which is the time taken for the extraction of image features and upload them into the database for a given size of dataset.

iii.) **Average Retrieval Time:** The retrieval efficiency is measured in Average Retrieval Time which can be defined as average time taken to retrieve images from the database. Average Retrieval Time (AvgRTime) can be formulated as:

$$\text{Average Retrival Time (AvgRTime)} = \frac{1}{M}\sum_{m=1}^{M} t_{m,n} \qquad (10)$$

Where $t_{m,n}$ represents the time cost to retrieve n relevant images for m^{th} query image.

iv.) **Precision:** Precision is defined as the ratio of number of relevant images from retrieved images to total number of retrieved images from the database. If the total number of retrieved images are N, then Precision can be calculated as:

$$\text{Precision}(P_r) = \frac{\text{Number of relevant images from the retrieved images}}{\text{Total number of retrieved images}(N)} \qquad (11)$$

Average of Precision for each image category is calculated and finally mean of average precisions is used for the performance evaluation of CBIR system. The Average Retrieval Precision for each image category is calculated as:

$$\text{Average Retrieval Precision}\,(P_{avg}(C)) = \frac{1}{L}\sum_{r=1}^{L} P_r \qquad (12)$$

Where $P_{avg}(C)$ shows Average Retrieval Precision for category (C) and L represents the total number of query images in that category. We can compute the Mean Average Precision (MAP) for our experiment as:

$$\text{Mean Average Precision (MAP)} = \frac{1}{K}\sum_{C=1}^{K} P_{avg}(C) \qquad (13)$$

Here, K is the number of categories present in the database.

4.2. Dataset

To analyze the performance of the proposed method, the experiment is performed on massive MRI images dataset. 500GB sized MRI Images Dataset has been collected from Rajindra Medical College and Hospital, Patiala, Punjab, India as a sample dataset. 350GB sized dataset has been used as training dataset and 150GB sized dataset has been used as testing dataset. The dataset consists of 24 different categories of MRI images of different aged people. Each image of MRI dataset has the size of 1105x646.

Table 2, Table 3 and Table 4 demonstrates the CPU usage rate of different DataNodes for processing 130GB, 230GB and 350GB sized MRI images dataset respectively. Fig. 4, Fig.5 and Fig. 6 shows the CPU usage rate versus point in time for processing 130GB, 230GB and 350GB sized dataset respectively. From Fig.4, Fig.5 and Fig.6, it is evident that CPU usage rate for processing different sized dataset is almost same at t6 moment, which clearly shows the efficiency of the proposed method to analyze large scale MRI images dataset.

Table 2: CPU Usage Rate of the proposed method for processing 130 GB MRI images dataset.

Point in time	CPU Usage Rate for processing 130 GB MRI images dataset		
	at DataNode1	at DataNode2	at DataNode3
t1	95	65	95
t2	95	65	90
t3	97	30	55
t4	52	30	55
t5	35	30	55
t6	30	30	35

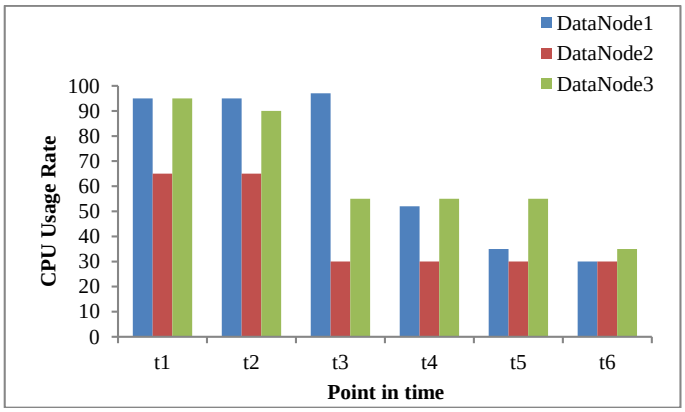

Fig 4: Graph showing CPU Usage Rate versus point in time for the proposed method to process 130GB MRI images dataset at different DataNodes.

Table 3: CPU Usage Rate of the proposed method for processing 230 GB MRI images dataset.

Point in time	CPU Usage Rate for processing 230 GB MRI images dataset		
	at DataNode1	at DataNode2	at DataNode3
t1	92	92	90
t2	95	92	90
t3	98	80	45
t4	45	60	35
t5	30	60	35
t6	30	35	35

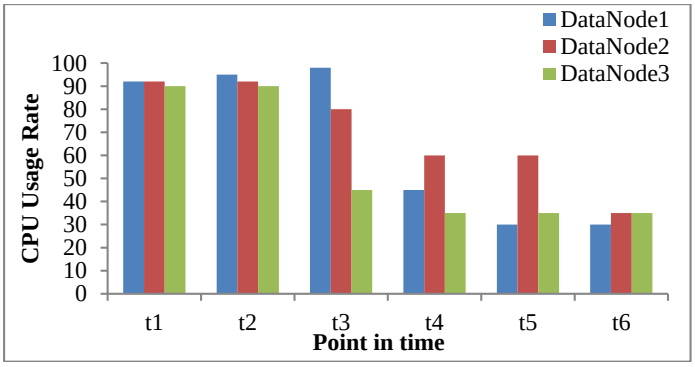

Fig 5: Graph showing CPU Usage Rate versus point in time for the proposed method to process 230GB MRI images dataset at different DataNodes.

Table 4: CPU Usage Rate of the proposed method for processing 350 GB MRI images dataset

Point in time	CPU Usage Rate for processing 350 GB MRI images dataset		
	at DataNode1	at DataNode2	at DataNode3
t1	92	93	90
t2	95	93	90
t3	97	87	90
t4	45	60	45
t5	35	60	45
t6	35	35	40

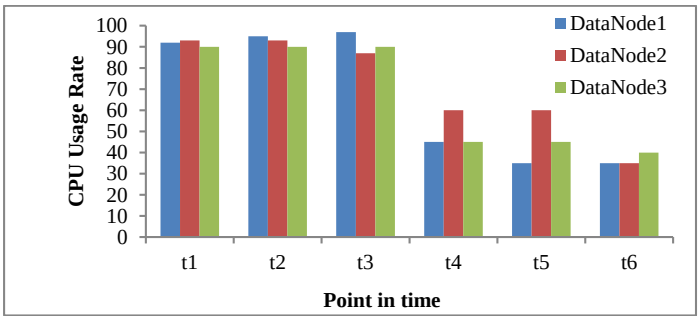

Fig 6: Graph showing CPU Usage Rate versus point in time for the proposed method to process 350GB MRI images dataset at different DataNodes.

Table 5 demonstrates the storage time of the proposed method to upload different sized MRI images dataset. Fig. 7 shows the storage time for different nodes with different sizes of MRI images dataset. From Fig. 7, it is clear that the storage time grows slowly with the massive increase in size of the dataset, which clearly shows the storage efficiency of the proposed method.

Table 5: Storage Time to upload different sizes of MRI images dataset.

Size of MRI images dataset (in GB)	Storage Time (in seconds)
10	107.67
50	478.13
100	753.53
150	1003.06
200	1211.19
250	1321.81
300	1427.23
350	1519.29

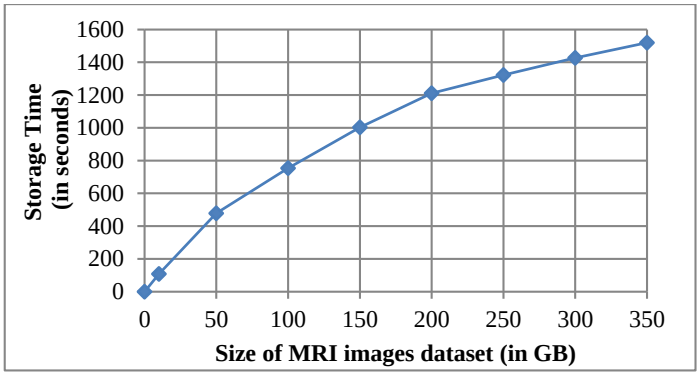

Fig. 7: Graph showing Storage Time versus Size of MRI images dataset for the proposed method.

Table 6 shows the Average Retrieval Time of the proposed method for the processing of different sizes of MRI images dataset. With different sizes of MRI images dataset for different nodes, the Average Retrieval Time is shown in Fig. 8. From Fig. 8, it is clear that the Average Retrieval Time grows slowly with the massive increase in size of the dataset, which clearly shows the retrieval efficiency of the proposed system.

Table6: Average Retrieval Time for different Sizes of MRI images dataset

Size of MRI images dataset(in GB)	Average Retrieval Time (in seconds)
10	42.04
50	193.37
100	302.78
150	397.51
200	489.19
250	565.67
300	625.31
350	658.48

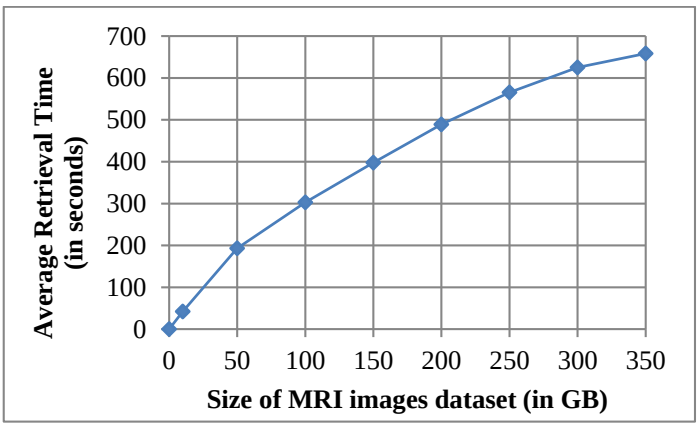

Fig. 8: Graph showing Average Retrieval Time versus Size of MRI images dataset for the proposed method.

Table 7 shows the Average Retrieval Precision (ARP) (%) and Table 8 demonstrates the Mean Average Precision (MAP) (%) of the proposed method for first N retrieved images of the MRI images dataset. Fig. 9 diagrammatically represents the

Mean Average Precision (MAP) (%) versus first N retrieved images of the dataset.

The storage and retrieval efficiency of proposed CBIR method is compared with state-of-the-art CBIR methods such as LDTP, CoALTP, 3DLTCoP+3DLTP and LNIP. We applied state-of-the-art methods on sample MRI images dataset using intel Core i7,8700k, 3.7GHz with 32 GB RAM and 2TB hard drive. The storage and retrieval efficiency of the above methods is evaluated on sample MRI images dataset. Table 9 shows the comparison of proposed method with above methods on the basis of storage time for different sizes of sample MRI images dataset. Fig. 10 shows the storage time comparison of the method proposed in our work and state-of-the-art methods for different sizes of sample dataset and clearly demonstrates that the storage time of the traditional methods increase sharply while that of the our proposed method grows slowly with the massive increase in size of the dataset. From Fig. 10, it is clear that the method proposed in our work outperforms state-of-the-art methods in terms of storage time for large scale MRI images dataset.

Table 7: Average Retrieval Precision (%) of proposed method for sample MRI images dataset.

Different categories of MRI dataset	Average Retrieval Precision (ARP) (%)						
	N=20	N=40	N=60	N=80	N=100	N=120	N=140
Images of Brain MRI	52.51	45.47	39.35	34.24	30.08	26.80	23.96
Images of Cervical MRI	56.22	49.18	43.11	38.08	34.02	30.26	27.45
Images of Dorsal MRI	57.93	50.17	44.12	39.45	35.29	31.59	28.62
Images of Lumbar MRI	53.10	46.00	39.88	34.75	30.85	27.65	25.01
Images of Pelvis MRI	61.52	54.40	48.36	43.31	39.28	35.51	32.68
Images of Thigh MRI	66.60	59.59	53.56	48.50	44.47	40.77	37.81
Images of Leg MRI	61.72	54.45	48.44	43.38	39.37	35.49	32.56
Images of Foot MRI	67.13	60.03	53.90	48.75	44.85	41.62	39.03
Images of Ankle MRI	65.35	58.27	52.15	47.08	43.01	39.28	36.41
Images of Abdomen MRI	53.00	45.88	39.70	34.62	30.73	27.52	24.80
Images of Neck MRI	54.61	47.46	41.38	36.20	32.17	28.30	25.46
Images of Arm MRI	63.27	56.22	50.15	45.11	41.00	37.16	34.39
Images of Forearm MRI	66.47	59.34	53.28	48.22	44.12	40.40	37.59
Images of Hand MRI	70.23	63.15	57.08	52.04	48.01	44.23	40.42
Images of Elbow MRI	64.85	57.54	51.29	46.18	42.12	38.75	35.92
Images of Wrist MRI	67.55	60.30	54.23	49.19	45.09	41.26	38.49
Images of Shoulder MRI	65.22	58.11	51.98	46.87	42.97	39.75	37.11
Images of Breast MRI	55.08	47.97	41.84	36.73	32.83	29.61	26.97
Images of Hip MRI	58.18	51.14	45.07	40.04	35.98	32.22	29.41
Images of knee MRI	63.46	56.33	50.27	45.20	41.11	37.39	34.56
Images of Face MRI	55.01	47.90	41.75	36.65	32.76	29.55	26.88
Images of Chest MRI	59.62	52.37	46.30	41.26	37.16	33.33	30.56
Images of Heart MRI	55.51	48.36	42.28	37.10	33.05	29.40	26.55
Images of Orbit MRI	57.06	49.96	44.82	39.75	35.86	32.60	29.90

Table 8: Mean Average Precision (%) of proposed method for sample MRI images dataset.

	Mean Average Precision (MAP) (%)						
	N=20	N=40	N=60	N=80	N=100	N=120	N=140
Whole MRI Images Database	60.46	53.31	47.26	42.19	38.17	34.60	31.77

Fig. 9: Graph showing Mean Average Precision (%) versus number of retrieved images from sample MRI images dataset for proposed method.

Table 9: Storage Time of the proposed method and state-of-the-art methods for different sizes of sample MRI images dataset.

Size of MRI images dataset(in GB)	Storage Time(in seconds)				
	LDTP	CoALTP	3DLTCoP+3DLTP	LNIP	Proposed Method
10	113.82	151.03	219.53	232.33	107.67
50	569.27	779.67	1107.50	1180.11	478.13
100	1183.01	1589.19	2274.67	2377.73	753.53
150	1797.38	2455.95	3469.21	3683.34	1003.06
200	2569.31	3405.73	4794.43	4935.69	1211.19
250	3574.77	4520.00	5862.78	6167.55	1321.81
300	4737.61	5815.23	7307.08	7640.70	1427.23
350	6509.20	7674.33	9162.99	9533.47	1519.29

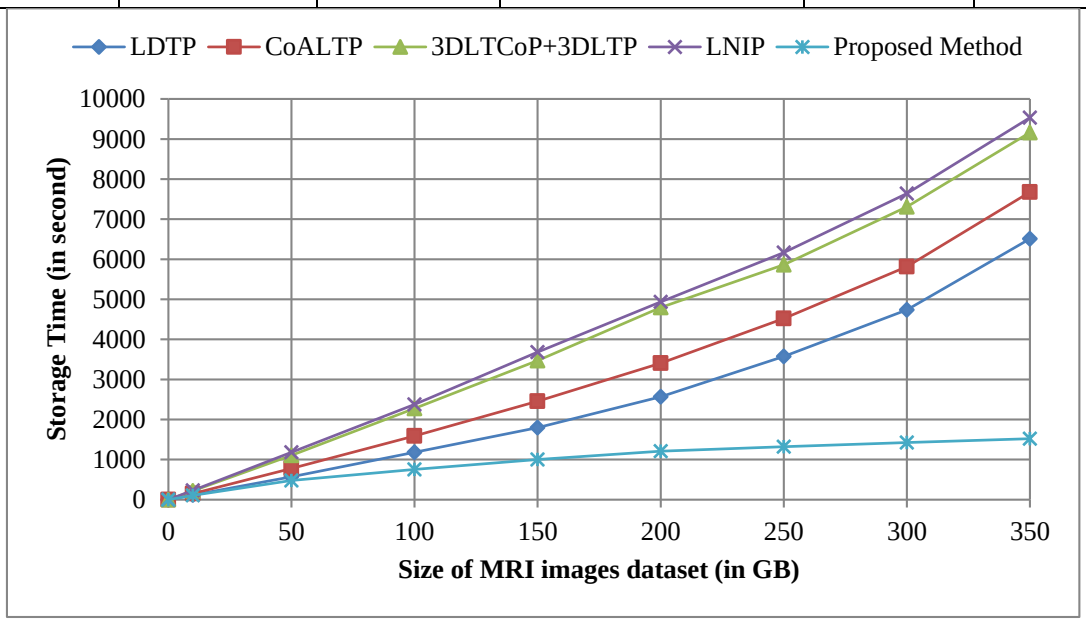

Fig.10: Graph showing comparison of the proposed method and state-of-the-art methods in term of storage time.

Table 10 shows the Average retrieval time of the proposed method and state-of-the-art methods for different sizes of MRI images dataset. Fig. 11 shows the Average Retrieval Time comparison of the proposed method with state-of-the-art methods for different sizes of dataset and demonstrates that the average retrieval time of state-of-the-art methods increases sharply while that of the proposed method grows slowly with massive increase in size of the dataset. Fig. 11 clearly demonstrates that proposed method outperforms state-of-the-art methods in term of average retrieval time for large scale MRI images dataset.

Table 10: Average Retrieval Time of the proposed method and state- of-the-art methods.

Size of MRI images dataset(in GB)	Average Retrieval Time(in seconds)				
	LNIP	LDTP	CoALTP	3DLTCoP+3DLTP	Proposed Method
10	51.09	83.76	112.48	208.55	42.04
50	252.63	410.43	556.13	1017.48	193.37
100	517.35	827.02	1027.69	2065.91	302.78
150	772.87	1294.73	1583.20	3177.58	397.51
200	1042.27	1722.19	2171.52	4309.77	489.19
250	1387.67	2283.99	2874.19	5667.33	565.67
300	1801.59	2970.23	3915.35	7201.87	625.31
350	2637.28	3889.89	5227.11	9034.17	658.48

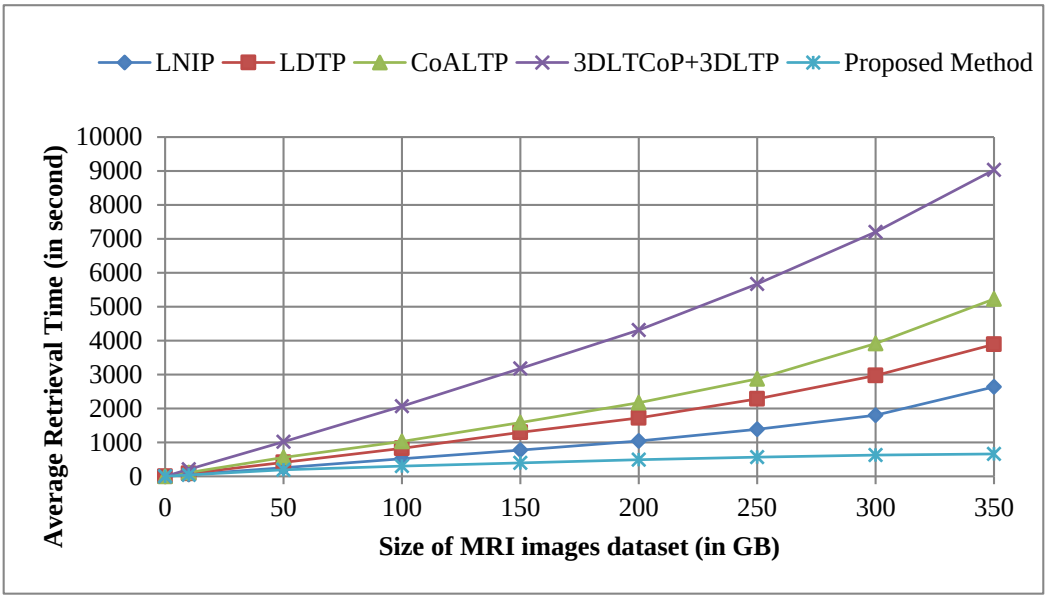

Fig. 11: Graph showing comparison of the proposed method and state-of-the-art methods in term of average retrieval time.

5. CONCLUSION AND FUTURE SCOPE

The traditional CBIR systems have the problem of low efficiency to analyze large scale medical images database. Due to this, an efficient CBIR system using Hadoop is put forward to analyze large scale MRI images dataset for early disease diagnosis. The results obtained show that the proposed method has improved storage and retrieval efficiency as compared to state-of-the-art CBIR methods for large scale MRI images dataset. In the future scope, clustering can be incorporated to the proposed CBIR system to further improve its efficiency.

Acknowledgement

Authors are highly thankful to the RIC department of IKG Punjab Technical University, Kapurthala, Punjab, India and the Radiology Department of Government Rajindra Medical College and Hospital, Patiala, Punjab, India for providing the valuable support to conduct this research work.

REFERENCES

[1] Müller, H., Michoux, 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.

[2] Yao, Q. A., Zheng, H., Xu, Z. Y., Wu, Q., Li, Z. W., & Lifen, Y. (2014). Massive medical images retrieval system based on Hadoop. *Journal of Multimedia*, *9*(2), 216.

[3] Li, Z., Zhang, X., Müller, H., & Zhang, S. (2018). Large-

scale retrieval for medical image analytics: A comprehensive review. *Medical image analysis*, *43*, 66-84.

[4] Zhang, S., & Metaxas, D. (2016). Large-Scale medical image analytics: Recent methodologies, applications and Future directions. *Medical image analysis*, 33, 98-101.

[5] Loganathan, A., Sinha, A., Muthuramakrishnan, V., & Natarajan, S. (2014). A systematic approach to Big Data. *International Journal of Information & Computation Technology*, *4*(09), 869-878.

[6] Zhang, X., Liu, W., Dundar, M., Badve, S., & Zhang, S. (2015). Towards large-scale histopathological image analysis: Hashing-based image retrieval. *IEEE Transactions on Medical Imaging*, *34*(2), 496-506.

[7] Ahmadian, A., Mostafa, A., Abolhassani, M. D., & Salimpour, Y. (2006, January). A texture classification method for diffused liver diseases using Gabor wavelets. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*(pp. 1567-1570). IEEE.

[8] Loupias, E., Sebe, N., Bres, S., & Jolion, J. M. (2000, September). Wavelet-based salient points for image retrieval. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)* (Vol. 2, pp. 518-521). IEEE.

[9] Kokare, M., Biswas, P. K., & Chatterji, B. N. (2006). Rotation-invariant texture image retrieval using rotated complex wavelet filters. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *36*(6), 1273-1282.

[10] Celik, T., & Tjahjadi, T. (2009). Multiscale texture classification using dual-tree complex wavelet transform. *Pattern Recognition Letters*, *30*(3), 331-339.

[11] Kokare, M., Biswas, P. K., & Chatterji, B. N. (2007). Texture image retrieval using rotated wavelet filters. *Pattern recognition letters*, *28*(10), 1240-1249.

[12] Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7), 971-987.

[13] Guo, Z., Zhang, L., & Zhang, D. (2010). A completed modeling of local binary pattern operator for texture classification. *IEEE transactions on image processing*, *19*(6), 1657-1663.

[14] Liao, S., Law, M. W., & Chung, A. C. (2009). Dominant local binary patterns for texture classification. *IEEE transactions on image processing*, *18*(5), 1107-1118.

[15] Liao, S., Zhu, X., Lei, Z., Zhang, L., & Li, S. Z. (2007, August). Learning multi-scale block local binary patterns for face recognition. In *International Conference on Biometrics* (pp. 828-837). Springer, Berlin, Heidelberg.

[16] Zhang, B., Gao, Y., Zhao, S., & Liu, J. (2010). Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE transactions on image processing*, *19*(2), 533-544.

[17] Papakostas, G. A., Koulouriotis, D. E., Karakasis, E. G., & Tourassis, V. D. (2013). Moment-based local binary patterns: a novel descriptor for invariant pattern recognition applications. *Neurocomputing*, *99*, 358-371.

[18] Murala, S., Maheshwari, R. P., & Balasubramanian, R. (2012). Directional local extrema patterns: a new descriptor for content based image retrieval. *International journal of multimedia information retrieval*, *1*(3), 191-203.

[19] Subrahmanyam, M., Maheshwari, R. P., & Balasubramanian, R. (2012). Local maximum edge binary patterns: a new descriptor for image retrieval and object tracking. *Signal Processing*, *92*(6), 1467-1479.

[20] Reddy, P. V. B., & Reddy, A. R. M. (2014). Content based image indexing and retrieval using directional local extrema and magnitude patterns. *AEU-International Journal of Electronics and Communications*, *68*(7), 637-643.

[21] Ren, J., Jiang, X., & Yuan, J. (2013). Noise-resistant local binary pattern with an embedded error-correction mechanism. *IEEE Transactions on Image Processing*, *22*(10), 4049-4060.

[22] Zhao, Y., Jia, W., Hu, R. X., & Min, H. (2013). Completed robust local binary pattern for texture classification. *Neurocomputing*, *106*, 68-76.

[23] Tan, X., & Triggs, W. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, *19*(6), 1635-1650.

[24] Wu, X., Sun, J., Fan, G., & Wang, Z. (2015). Improved local ternary patterns for automatic target recognition in infrared imagery. *Sensors*, *15*(3), 6399-6418.

[25] Murala, S., Maheshwari, R. P., & Balasubramanian, R. (2012). Local tetra patterns: a new feature descriptor for content-based image retrieval. *IEEE transactions on image processing*, *21*(5), 2874-2886.

[26] Dubey, S. R., Singh, S. K., & Singh, R. K. (2015). Local diagonal extrema pattern: a new and efficient feature descriptor for CT image retrieval. *IEEE Signal Processing Letters*, *22*(9), 1215-1219.

[27] Subrahmanyam, M., Maheshwari, R. P., & Balasubramanian, R. (2012). Local maximum edge binary patterns: a new descriptor for image retrieval and object tracking. *Signal Processing*, *92*(6), 1467-1479.

[28] Murala, S., & Wu, Q. J. (2013). Local ternary co-occurrence patterns: a new feature descriptor for MRI and CT image retrieval. *Neurocomputing*, *119*, 399-412.

[29] Naghashi, V. (2018). Co-occurrence of adjacent sparse local ternary patterns: A feature descriptor for texture and face image retrieval. *Optik*, *157*, 877-889.

[30] Murala, S., & Wu, Q. J. (2015). Spherical symmetric 3D

local ternary patterns for natural, texture and biomedical image indexing and retrieval. *Neurocomputing*, *149*, 1502-1514.

[31] Murala, S., & Wu, Q. J. (2014). Local mesh patterns versus local binary patterns: biomedical image indexing and retrieval. *IEEE Journal of Biomedical and Health Informatics*, *18*(3), 929-938.

[32] Murala, S., & Wu, Q. M. (2013). Peak valley edge patterns: a new descriptor for biomedical image indexing and retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 444-449).

[33] Murala, S., & Wu, Q. J. (2014). MRI and CT image indexing and retrieval using local mesh peak valley edge patterns. *Signal processing: image communication*, *29*(3), 400-409.

[34] Verma, M., & Raman, B. (2015). Center symmetric local binary co-occurrence pattern for texture, face and bio-medical image retrieval. *Journal of Visual Communication and Image Representation*, *32*, 224-236.

[35] Rivera, A. R., Castillo, J. R., & Chae, O. O. (2013). Local directional number pattern for face analysis: Face and expression recognition. *IEEE transactions on image processing*, *22*(5), 1740-1752.

[36] Chakraborty, S., Singh, S. K., & Chakraborty, P. (2018). Local gradient hexa pattern: A descriptor for face recognition and retrieval. *ieee transactions on circuits and systems for video technology*, *28*(1), 171-180.

[37] Verma, M., & Raman, B. (2018). Local neighborhood difference pattern: A new feature descriptor for natural and texture image retrieval. *Multimedia Tools and Applications*, *77*(10), 11843-11866.

[38] Chahi, A., Ruichek, Y., & Touahni, R. (2018). Local directional ternary pattern: A new texture descriptor for texture classification. *Computer Vision and Image Understanding*, *169*, 14-27.

[39] Fadaei, S., Amirfattahi, R., & Ahmadzadeh, M. R. (2017). Local derivative radial patterns: A new texture descriptor for content-based image retrieval. *Signal Processing*, *137*, 274-286.

[40] Ruichek, Y. (2018). Local concave-and-convex micro-structure patterns for texture classification. *Pattern Recognition*, *76*, 303-322.

[41] El Khadiri, I., Kas, M., El Merabet, Y., Ruichek, Y., & Touahni, R. (2018). Repulsive-and-attractive local binary gradient contours: New and efficient feature descriptors for texture classification. *Information Sciences*, *467*, 634-653.

[42] Agarwal, M., Singhal, A., & Lall, B. (2018). 3D local ternary co-occurrence patterns for natural, texture, face and bio medical image retrieval. *Neurocomputing*, *313*, 333-345.

[43] Banerjee, P., Bhunia, A. K., Bhattacharyya, A., Roy, P. P., & Murala, S. (2018). Local Neighborhood Intensity Pattern–A new texture feature descriptor for image retrieval. *Expert Systems with Applications*, *113*, 100-115.

[44] Kuo, M. H., Sahama, T., Kushniruk, A. W., Borycki, E. M., & Grunwell, D. K. (2014). Health big data analytics: current perspectives, challenges and potential solutions. *International Journal of Big Data Intelligence*, *1*(1-2), 114-126.

[45] Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information sciences*, *275*, 314-347.

[46] Fan, J., Han, F., & Liu, H. (2014). Challenges of big data analysis. *National science review*, *1*(2), 293-314.

[47] Fang, R., Pouyanfar, S., Yang, Y., Chen, S. C., & Iyengar, S. S. (2016). Computational health informatics in the big data age: a survey. *ACM Computing Surveys (CSUR)*, *49*(1), 12.

[48] Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information systems*, *47*, 98-115.

[49] Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Ali, M., Kamaleldin, W., ... & Gani, A. (2014). Big data: survey, technologies, opportunities, and challenges. *The Scientific World Journal*, *2014*.

[50] Landset, S., Khoshgoftaar, T. M., Richter, A. N., & Hasanin, T. (2015). A survey of open source tools for machine learning with big data in the Hadoop ecosystem. *Journal of Big Data*, *2*(1), 24.

[51] Tsai, C. W., Lai, C. F., Chao, H. C., & Vasilakos, A. V. (2015). Big data analytics: a survey. *Journal of Big data*, *2*(1), 21.

[52] Pouyanfar, S., Yang, Y., Chen, S. C., Shyu, M. L., & Iyengar, S. S. (2018). Multimedia big data analytics: A survey. *ACM Computing Surveys (CSUR)*, *51*(1), 10.