International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

An Efficient Clustering Based CBIR System using Hadoop to Analyze Massive MRI Images Dataset for Early Disease Diagnosis

Harinder Singh, Kulvinder Singh Mann

Abstract-: ***In past decade, the use of medical imaging for disease diagnosis is increased rapidly. The medical images provide useful information about the anatomy of patients. The medical images are used not only to assist the doctors for diagnosis purpose, but also used in Research & Development for deeper insights and better understanding into cause and cure of numerous diseases. To retrieve medical images from large scale repositories in real time, there is urgent need of an efficient medical image retrieval system. For this purpose, an efficient clustering based content based image retrieval (CBIR) system using Hadoop is proposed to analyze massive magnetic resonance imaging (MRI) images dataset for early disease diagnosis. The proposed CBIR system uses Hadoop platform, local mesh peak valley edge patterns (LMePVEP) for feature extraction, MapReduce based parallel k-means algorithm for clustering and euclidean distance to measure similarity. The method proposed is tested and compared with state-of-the-art CBIR methods on massive MRI images dataset. The experimental results obtained show that the method proposed in our work outperforms traditional CBIR methods in terms of average retrieval time and mean average precision for massive MRI images dataset.***

Keywords: content based image retrieval (CBIR), Hadoop, magnetic resonance imaging (MRI) images dataset, local mesh peak valley edge patterns (LMePVEP), MapReduce based parallel k-means algorithm.

I. INTRODUCTION

The significance of medical images is increasing in the modern era for the disease diagnosis of patients. In recent times, the technological development of digital imaging equipments leads to hasty growth of medical images. The medical images have different categories either on the basis of body organ they are representing or depending on types of tissues of the same body organ. The medical image data exist in different medical modalities like X-rays, computed tomography (CT), positron emission tomography (PET), magnetic resonance imaging (MRI), and ultrasound (US). These medical modalities are highly capable to provide accurate disease diagnosis. For clinical diagnosis, it becomes highly important to manage medical image data for efficient retrieval of medical images. In the medical image processing field, the medical image retrieval has become one of the important research areas.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Harinder Singh*, Research Scholar, IKGPTU, Kapurthala, Punjab, India. Email- dhaliwalhsingh@gmail.com

Kulvinder Singh Mann, Professor, Department of IT, GNDEC, Ludhiana, Punjab, India.Email-mannkulvinder@yahoo.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

For the solution of this problem, many ideas came into existence but the most popular is content based image retrieval (CBIR). The CBIR retrieves most relevant images from massive images database using the visual content of image like shape, color and texture [1]. The CBIR system consists of three phases; first, selecting image features; second, selecting feature descriptor; third, selecting the similarity distance measure. The CBIR system is widely used in disease diagnosis and supports the clinical decision making process [2].

A. Motivation

CBIR is highly data intensive computing process. The retrieval efficiency for large-scale data is a challenging task [3]. For massive medical image dataset, computational efficiency and accuracy have become the key challenges [4]. To tackle these problems, an efficient CBIR system using Hadoop is proposed in [5] to analyze large scale MRI images dataset. The main problem of this proposed system is that it does not take into account the idea of clustering which can be very effective to improve the retrieval efficiency for massive images dataset. Clustering helps to reduce the size of the large dataset for searching which can be applied for fast retrieval of images. For this purpose, we need to design and implement CBIR system using Hadoop that uses the concept of clustering. This has motivated us to provide a clustering based CBIR system using Hadoop to analyze massive MRI images dataset.

B. Related Work

Various methods have been explored for CBIR systems in recent years. The local ternary pattern (LTP) is proposed in [6] which is insensitive to noise. The local mesh peak valley edge patterns (LMePVEP) is put forward in [7] for CT and MRI image retrieval. The local directional ternary pattern (LDTP) is put forward in [8]. The 3D local ternary co-occurrence patterns (3D-LTCOP) is presented in [9]. In this paper, 3D-LTCOP in combination with 3D-LTP is used for image retrieval purpose. Local Neighborhood Intensity Pattern (LNIP) is presented in [10] for pattern calculation. A new method based on orthogonal Fourier-Mellin moments (OFMMs) is proposed in [11] for effective indexing and retrieval of medical images. The analysis for MapReduce efficiency using parallel K-means algorithm for document clustering is proposed in [12]. Clustering of large data sets using MapReduce and Hadoop is provided in [13]. Hadoop architecture and its ecosystem for processing big data is described in [14]. A survey of big data challenges and technologies is presented in [15].

C. Main Contribution of this work

An efficient clustering based CBIR system using Hadoop is put forward to analyze massive MRI images dataset for early disease diagnosis. The proposed CBIR system uses Hadoop platform, LMePVEPs descriptor for extraction of texture features of images, MapReduce based parallel k-means algorithm for clustering the database and euclidean distance to measure similarity. The proposed method in our work is compared with state-of-the-art CBIR methods and the experimental results obtained reveal that method proposed in this work outperforms traditional CBIR methods in terms of average retrieval time and mean average precision. The work presented is structured as: Section 1 includes introduction, related work and main contribution of this work. The materials and methods used are highlighted in Section 2. Section 3 demonstrates proposed system framework that is carried out. The experimental results are shown in section 4 and finally, the conclusion and future scope is provided in section 5.

II. MATERIALS AND METHODS USED

A. Hadoop Platform

Hadoop Platform is the distributed computing platform for parallel processing of large dataset [1]. The working of Hadoop is based on master/slave architecture. In the Hadoop environment, master node is known as NameNode and slave nodes are known as DataNodes. The NameNode coordinates with DataNodes for storage and computation distribution and the user interacts with DataNodes only through NameNode. The following two modules form the base of Apache Hadoop:

- **Hadoop Distributed File System**: The Hadoop Distributed File System (HDFS) has data storage and management capability. The volumetric datasets are handled by HDFS. The dataset is divided by the HDFS of NameNode and the divided datasets are sent to the DataNodes before the execution of a data intensive application. Data can be replicated by HDFS among multiple nodes to provide reliability and fault tolerance of data.
- **MapReduce Programming Model:** The MapReduce is the programming model used by Hadoop for processing massive amount of dataset across a number of nodes. In this model, the MapReduce jobs are submitted to the NameNode by the user and the submitted jobs are transferred to the dataNodes by the NameNode in the cluster. The MapReduce is based on the idea of <key, value> pairs and is run on the functions known as Map and Reduce. In Map function, Mapper receives the input as <key, value> pair and the outputs are set of <key, value> pairs. All these output values are gathered and given to Reduce function. In Reduce function, Reducer receives <key, value> pairs from the Mapper and sort these set of values to obtain optimal 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.

B. Feature Extraction by LMePVEPs

The idea of LMePVEPs is put forward in [7] for CT and MRI image retrieval. For the center pixel $I(g_c)$, the first-order derivative in forward direction among P neighbors is computed as:

$$\begin{aligned} &\overrightarrow{I}^{j}_{P,R}(g_c,g_i) = I(g_{\alpha_1}) - I(g_i); i = 1,2,....,P \\ &\alpha_1 = 1 + \mathrm{mod}((i+P+j-1),P), \\ &\forall j = 1,2,.....,(P/2) \end{aligned} \tag{1}$$

here j represents distance of first-order derivation.

For center pixel $I(g_c)$, the first-order derivative in backward direction among P neighbors is computed as:

$$\overleftarrow{I}^{j}_{P,R}(g_{c,}g_i) = \begin{cases} I(g_{(P+i-j)}) - I(g_i), & if\ j \geq i \\ I(g_{(i-j)}) - I(g_i), & else \end{cases} \tag{2}$$

LMePVEP can be defined as:

$$LMePVEP^{j}_{P,R} = \begin{bmatrix} f_3(\overrightarrow{I}^{j}_{P,R}(g_c,g_1), \overleftarrow{I}^{j}_{P,R}(g_c,g_1)), \\ f_3(\overrightarrow{I}^{j}_{P,R}(g_c,g_2), \overleftarrow{I}^{j}_{P,R}(g_c,g_2)), \\ ...., f_3(\overrightarrow{I}^{j}_{P,R}(g_c,g_P), \overleftarrow{I}^{j}_{P,R}(g_c,g_P)) \end{bmatrix} \tag{3}$$

$$f_3(a,b) = \begin{cases} 1 & if\ a>0 \quad and \quad b>0 \\ 2 & if\ a<0 \quad and \quad b<0 \\ 0 & else \end{cases} \tag{4}$$

LMePVEP descriptor is the ternary pattern (0, 1, 2) and the two binary patterns i.e. local mesh valley edge pattern(LMeVEP) and local mesh peak edge pattern (LMePEP) are formed from these ternary patterns by using the idea of LTP [6].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 [7].

C. MapReduce based parallel k-means algorithm for clustering

The parallel k-means algorithm using MapReduce programming model for clustering volumetric dataset is shown in Fig 1.

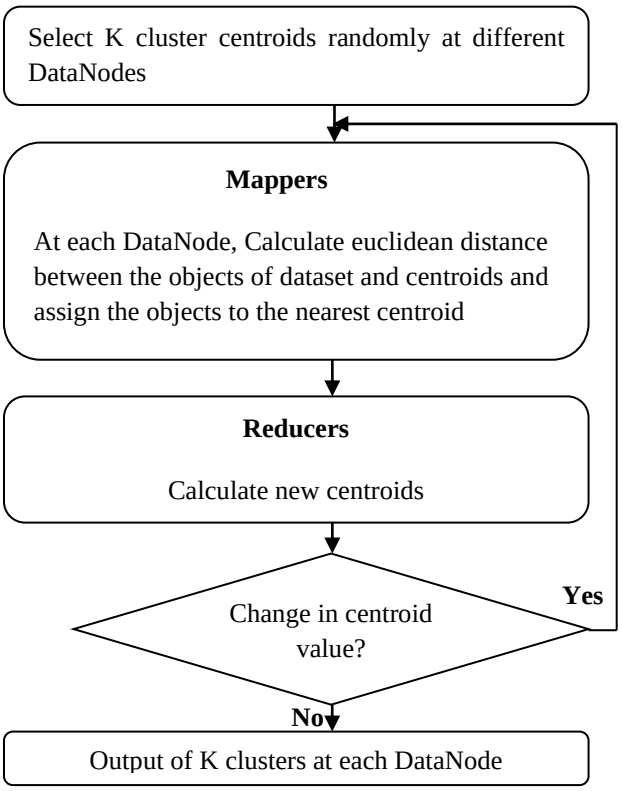

Fig 1: Stages of MapReduce based parallel K-means algorithm.

The details of algorithm are as below:
Step 1: **Selection of K cluster centroids:**
In this step, K cluster centroids are provided at each DataNode before the execution of algorithm and are loaded in the file called cluster_centroids. Parallel k-means use these values in the first iteration of Map function.
Step 2: **MapReduce based k-means parallel execution:**
The numeric feature vector database generated by feature extraction technique provides the input to the k-means algorithm. The K cluster centers are obtained from the database and are loaded in the file called centroid_data. HDFS copies centroid_data to each DataNode. The execution of MapReduce based parallel k-means has two parts: one, iteration and parallel part for calculating the distance between objects of dataset and centroids and assigning each and every object to nearest centroid using Map function: second, after each iteration, updation of new centroids after every object is assigned using Reduce function. The more details can be found in [12].

D. Similarity measure

In database, a feature vector represents each image as $f_{DB_j} = (f_{DB_{j1}}, f_{DB_{j2}}, ..., f_{DB_{jLg}})$; where $j = 1,2,.....,|DB|$. A feature vector represents query image Q as $f_Q = (f_{Q_1}, f_{Q_2}, ..., f_{Q_{Lg}})$ after the extraction of features. The objective of this function is for retrieving N images that most match with query image. To accomplish this task, distance is measured between images in the database |DB| and query image. The euclidean distance (ED) is used to calculate similarity measure as:

$$\text{ED (Q, DB)} = \sqrt{\sum_{i=1}^{Lg} (f_{DB_{ji}} - f_{Q_i})^2} \quad (5)$$

The normalized euclidean distance can be calculated as:

$$\text{Normalized-ED(Q, DB)} = \frac{\sqrt{\sum_{i=1}^{Lg} (f_{DB_{ji}} - f_{Q_i})^2}}{\sqrt{\sum_{i=1}^{Lg} (f_{DB_{ji}})^2} + \sqrt{\sum_{i=1}^{Lg} (f_{Q_i})^2}} \quad (6)$$

Here $f_{DB_{ji}}$ represents the i^{th} feature of j^{th} image in 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 represents the maximum match between the two images.

E. 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 are 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 for 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.

III. PROPOSED SYSTEM FRAMEWORK

The framework of proposed system is given in Fig. 2 and the method for the same is provided as below:

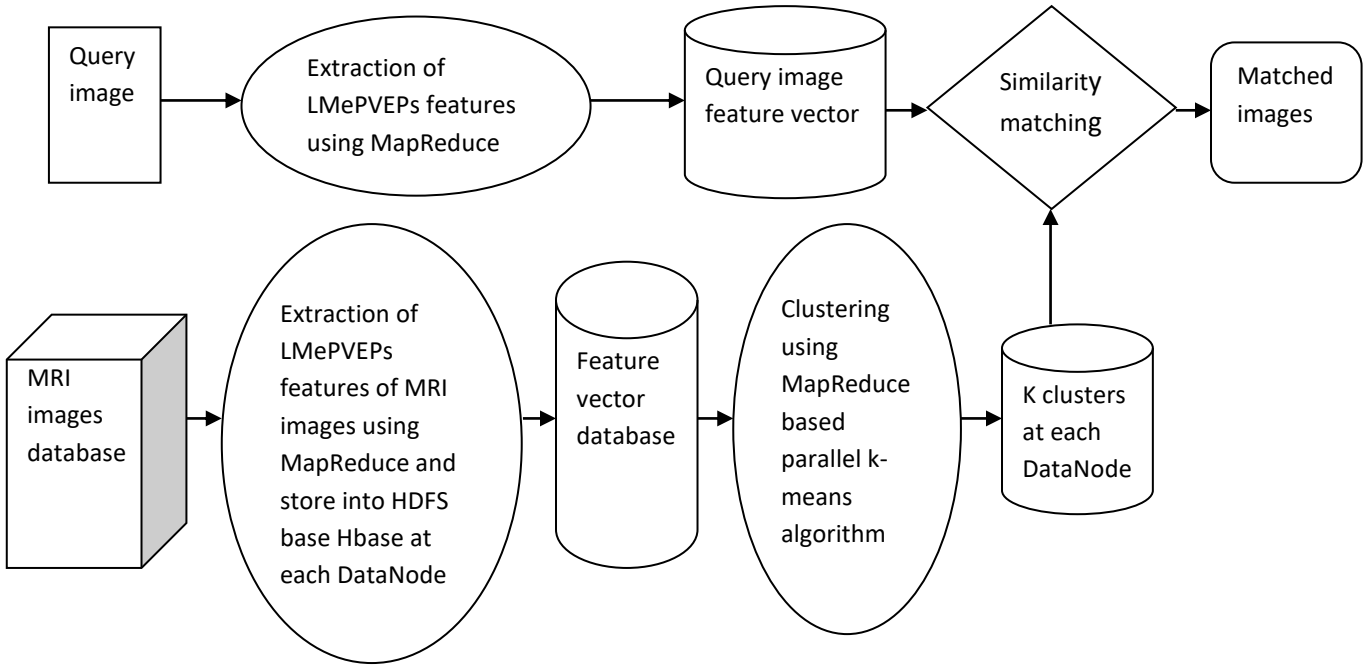

Fig.2: Proposed CBIR System

a) Load the MRI images into HDFS at different DataNodes. The NameNode splits MRI images dataset to different DataNodes.
b) In the Map Phase, the Map task takes an image from HDFS at each DataNode and extracts the LMePVEPs features of each image. In the Reduce Phase, LMePVEPs features extracted by Map task at each DataNode are stored in Hbase database of HDFS. HBase contains the image ID, pathname and LMPvEPs features of each image. The output of this phase is feature vector database.
c) Apply MapReduce based parallel k-means clustering algorithm (with K=24 cluster centroids) to the feature vector database. The output is 24 clusters at each DataNode.
d) To add new MRI images to clusters, the NameNode uniformly distribute images to different DataNodes. At each DataNode, the MapReduce model is applied for the extraction of LMePVEPs features for each image and calculates euclidean distance between image feature vector and each cluster feature vector and assign image to a cluster that has minimum euclidean distance.
e) With the submission of query image, the NameNode sends the query image to each DataNode and store into HDFS.
f) In the Map Phase, extract the LMePVEPs features of query image stored in HDFS at each DataNode and store these extracted features into a temporary file.
g) At each DataNode, calculate euclidean distance between query image feature vector and each cluster feature vector using Map function. The Map phase output is <key, value> pairs of <similarity, cluster ID> and input these to the Reducer. During the Reduce phase, perform similarity sorting for these <similarity, cluster ID> pairs and select the cluster that has minimum euclidean distance for query image.
h) Search the selected cluster for query image at each DataNode using MapReduce model. During the Map Phase, perform the similarity matching between feature vector of each image in the selected cluster and query image feature vector at each DataNode. The outcome of each Map task is the <key, value> pairs of < similarity, image ID>. During this similarity matching process, select those <similarity, Image ID> pairs that have similarity value in the predefined range and input these <key, value> pairs into the Reducers. During the Reduce phase, obtain all selected < similarity, imageID> pairs from each DataNode and perform similarity sorting of these pairs and first N pairs are written in HDFS.
i) Output image ID of written N <similarity, image ID> pairs in HDFS, and the user obtains the final results of image retrieval.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Evaluation criteria

To show the performance of method proposed, it is tested using massive MRI images database. The performance is evaluated on the basis of 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 on the basis of average retrieval time and mean average precision. We can define Storage Time, Average Retrieval Time and Precision as follows:

i. **Storage Time:** The Storage efficiency is measured in storage time. Storage time is defined as the time taken for the extraction of images features and stores them into the database for a given size of the dataset.

ii. **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:

Average Retrival Time (AvgRTime)

$$= \frac{1}{M}\sum_{m=1}^{M} t_{m,n} \quad (7)$$

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

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

$$\text{Precision}(P_r) = \frac{\text{Number of relevant images from the retrieved images}}{\text{Total number of retrieved images(N)}} \quad (8)$$

Average of Precision is calculated for each image category d 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:

$$Average\ \text{Re}trieval\ \text{Pr}ecision\ (P_{avg}(C)) = \frac{1}{L}\sum_{r=1}^{L} P_r \quad (9)$$

Where $P_{avg}(C)$ shows Average Retrieval Precision of category (C) and L represents total number of query images for that category.

We can compute Mean Average Precision (MAP) for our experiment as:

$$Mean\ Average\ \text{Pr}ecision\ (MAP) = \frac{1}{K}\sum_{C=1}^{K} P_{avg}\ (C) \quad (10)$$

Here K represents number of categories in the whole dataset.

B. Dataset

To analyze the efficiency of method proposed, the experiment is performed on massive MRI images dataset. 500 GB sized MRI images dataset is collected from Rajindra Medical 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 demonstrates the storage time of the proposed method to upload different sized MRI images dataset. Fig. 3 shows the storage time for different sizes of MRI images dataset. Fig. 3 clearly demonstrates that the storage time grows slowly with massive increase in size of the dataset, which clearly shows the storage efficiency of method proposed.

Table 2: Storage time to upload different sizes of MRI images dataset.

Size of MRI images dataset(in GB)	Storage Time (in seconds)
10	647.13
50	1519.78
100	2273.07
150	2697.53
200	2962.37
250	3217.11
300	3344.67
350	3459.09

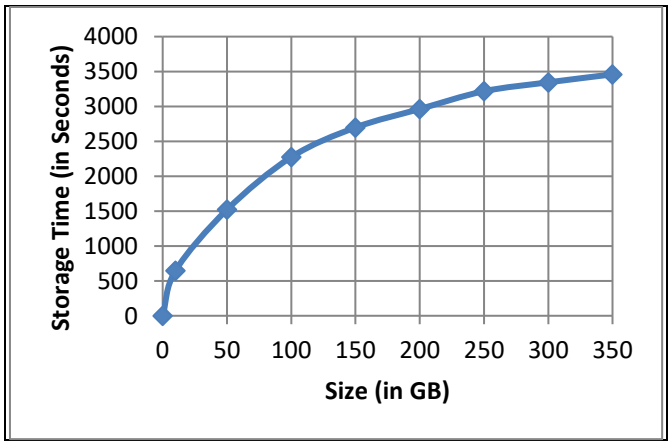

Fig. 3: Graph showing storage time of the proposed method for different sizes of MRI images dataset.

Table 3: Average retrieval time for different sizes of MRI images dataset.

Size of MRI images dataset(in GB)	Average Retrieval Time (in seconds)
10	17.64
50	121.17
100	213.46
150	301.35
200	381.49
250	453.77
300	519.61
350	567.08

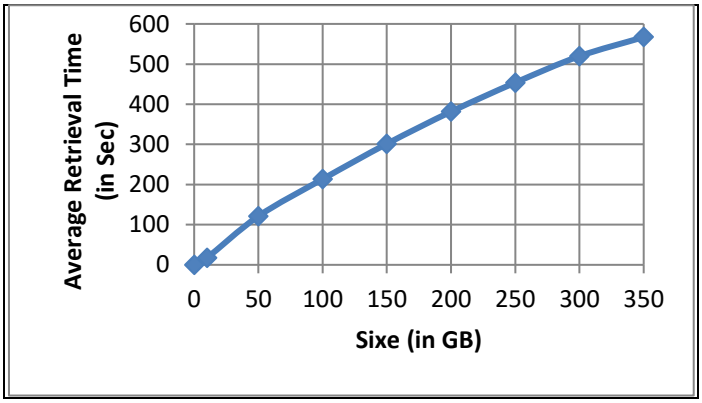

Fig. 4: Graph showing average retrieval time of the method proposed for different sizes of MRI images dataset.

Table 3 shows average retrieval time of the method proposed for the processing of different sizes of MRI images dataset. Fig 4 graphically represent the average retrieval time for different sizes of the dataset. Fig. 4 clearly demonstrates that the average retrieval time grows slowly with the massive increase in size of the dataset, which shows the retrieval efficiency of the proposed method.

Table 4 shows the average retrieval precision (ARP) (%) and Table 5 demonstrates the mean average precision (MAP) (%) of the proposed method for first N retrieved images from the MRI images dataset. Fig. 5 diagrammatically represents the mean average precision (MAP) (%) for top N retrieved images from the dataset.

Table 4: Average Retrieval Precision (%) of proposed method for massive MRI images dataset.

Different categories of MRI images dataset	Average Retrieval Precision (%)						
	N=20	N=40	N=60	N=80	N=100	N=120	N=140
Brain	64.72	54.52	47.40	40.30	35.18	30.87	27.08
Cervical	70.45	57.07	50.25	44.37	38.98	34.70	30.52
Dorsal	74.02	63.13	53.81	46.05	39.35	34.47	30.75
Lumbar	68.96	60.07	52.65	46.45	41.92	38.35	35.48
Pelvis	72.15	60.03	52.45	47.35	42.55	37.96	34.03
Thigh	86.76	75.15	64.07	58.37	53.39	48.61	44.95
Leg	83.83	70.62	58.50	50.47	43.41	38.25	34.01
Foot	88.03	80.15	72.98	66.72	61.45	57.18	53.35
Ankle	86.25	75.10	66.64	58.44	51.30	45.98	42.00
Abdomen	62.85	55.77	49.62	44.88	40.42	36.50	33.05
Neck	74.81	65.59	57.46	50.22	44.26	38.37	33.52
Arm	75.45	68.37	62.65	57.07	53.03	49.85	46.98
Forearm	86.56	77.39	69.36	62.24	56.21	50.47	45.65
Hand	92.08	83.21	76.13	69.09	63.05	57.23	51.47
Elbow	78.92	65.59	58.38	52.24	47.18	42.46	39.17
Wrist	85.59	77.39	70.25	64.27	59.16	54.31	50.52
Shoulder	79.27	67.18	59.03	52.96	48.05	43.80	40.16
Breast	75.27	62.22	54.14	46.78	40.92	35.63	31.48
Hip	69.30	57.39	49.52	44.04	39.06	34.98	31.91
Knee	85.61	72.41	64.30	57.26	51.14	45.75	40.86

Face	75.21	66.03	58.05	50.90	45.54	40.50	35.62
Chest	71.71	61.42	53.39	47.34	42.25	37.42	33.59
Heart	66.68	54.52	46.33	41.16	36.14	31.43	28.00
Orbit	69.56	59.52	52.05	45.95	41.30	37.04	33.68

Table 5: Mean Average Precision (%) of proposed method for massive MRI images dataset.

	Mean Average Precision (%)						
	N=20	N=40	N=60	N=80	N=100	N=120	N=140
Whole MRI Images Database	76.83	66.24	58.30	51.87	46.46	41.75	37.82

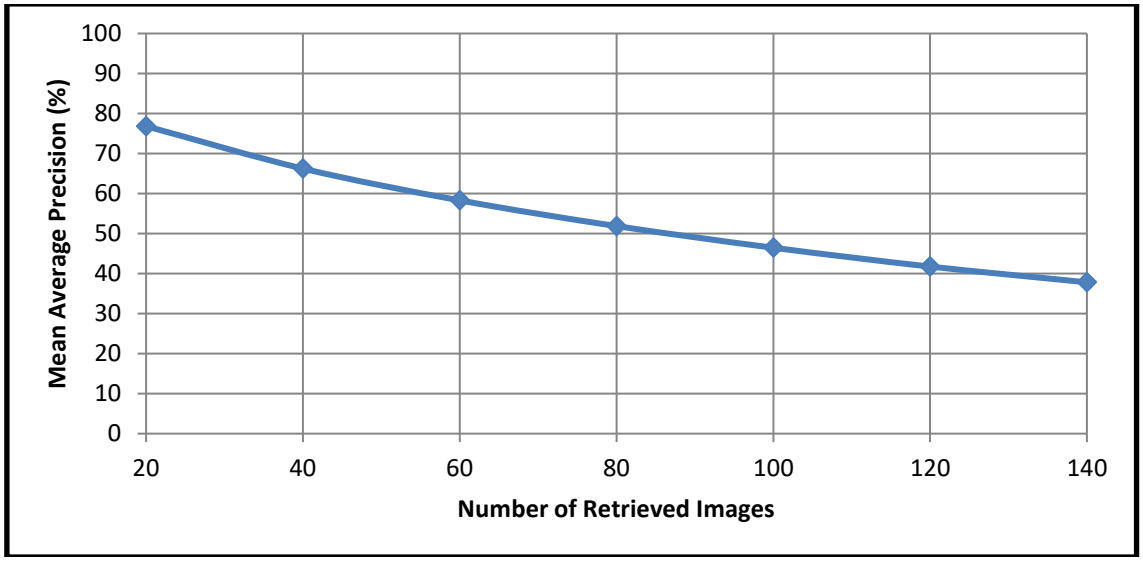

Fig. 5: Graph showing Mean Average Precision (%) versus number of retrieved Images from massive MRI images dataset for the proposed method.

To show the efficiency of proposed method, it is compared with state-of-the-art CBIR methods on the basis of average retrieval time and mean average precision. Table 6 provides the average retrieval time of method proposed in our work and existing method for different sizes of sample MRI images dataset. Fig. 6 shows comparison of method proposed with existing method for varying sizes of MRI images dataset in terms of average retrieval time. Fig. 6 clearly demonstrates that the method proposed in our work outperforms state-of-the-art method in terms of average retrieval time.

Table 7 reveals the mean average precision (%) of method proposed and state-of-the-art CBIR method for top N retrieved images from the sample MRI images dataset. Fig. 7 shows the comparison of method proposed with existing CBIR method in terms of mean average precision (%). Fig. 7 clearly demonstrates that method proposed in our work outperforms state-of-the-art CBIR method in terms of mean average precision (%).

Table 6: Average retrieval time of method proposed and other state-of-the-art CBIR method.

Size of MRI images dataset(in GB)	Average Retrieval Time (in seconds)	
	Method Proposed in [5]	Proposed Method
10	42.04	17.64
50	193.37	121.17
100	302.78	213.46
150	397.51	301.35
200	489.19	381.49
250	565.67	453.77
300	625.31	519.61
350	658.48	567.08

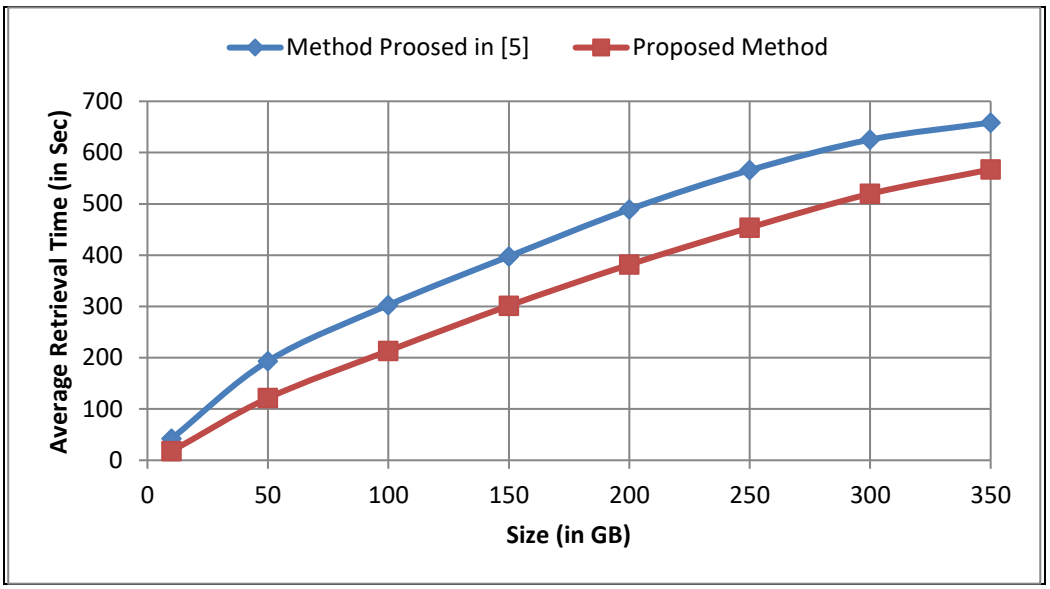

Fig 6: Graph showing comparison of method proposed with state-of-the-art CBIR method on the basis of average retrieval time.

Table 7: Mean Average Precision (%) of the method proposed and state-of the-art CBIR method for top N retrieved images from massive MRI images dataset.

Method	Mean Average Precision (%)						
	N=20	N=40	N=60	N=80	N=100	N=120	N=140
Method Proposed in [5]	60.46	53.31	47.26	42.19	38.17	34.60	31.77
Proposed Method	76.83	66.24	58.30	51.87	46.46	41.75	37.82

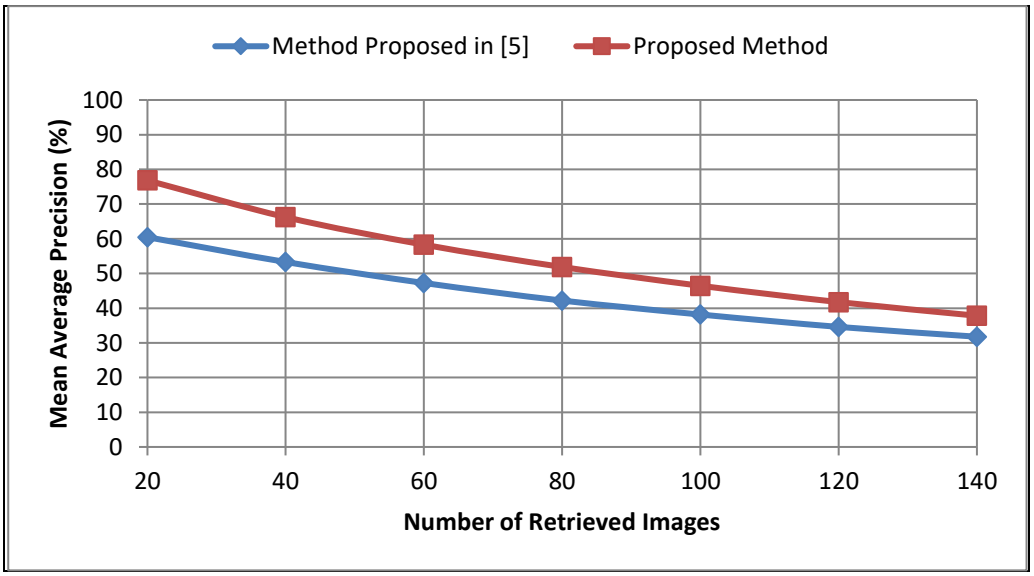

Fig. 7: Graph showing comparison of method proposed with state-of-the-art CBIR method in terms of mean average precision (%).

V. CONCLUSION AND FUTURE SCOPE

An efficient clustering based CBIR system using Hadoop is put forward to analyze massive MRI images dataset for early disease diagnosis. The experimental results obtained reveal that the method proposed in our work outperforms existing state-of-the-art CBIR methods in terms of average retrieval time and mean average precision for massive MRI images dataset. In the future scope, the developed system can be integrated with various heterogeneous information systems to provide interoperability among them which can be used for better disease diagnosis.

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. 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.
2. 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.
3. 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.
4. 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.
5 Singh, H., & Mann, K.S. (2019). Design and Development of an efficient CBIR system using Hadoop to analyze large scale MRI images dataset for early disease diagnosis. *International Journal of Applied Engineering Research*, 14(11), 2636-2646.
6. 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.
7. 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
8. 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.
9. 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.
10. 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.
11. Aggarwal, A., Sharma, S., Singh, K., Singh, H., & Kumar, S. (2019). A new approach for effective retrieval and indexing of medical images. *Biomedical Signal Processing and Control*, *50*, 10-34.
12. Sardar, T. H., & Ansari, Z. (2018). An analysis of MapReduce efficiency in document clustering using parallel K-means algorithm. *Future Computing and Informatics Journal*, *3*(2), 200-209.
13. Zhou, P., Lei, J., & Ye, W. (2011). Large-scale data sets clustering based on MapReduce and Hadoop. *Journal of Computational Information Systems*, *7*(16), 5956-5963.
14. Nagarjuna, D. N., & Yogesh, N. (2015). A Survey on Hadoop Architecture & its Ecosystem to Process Big Data-Real World Hadoop Use Cases.
15. 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.