

# Deep Root Memory Optimized Indexing Methodology for Image Search Engines

R. Karthikeyan<sup>1,\*</sup>, A. Celine Kavida<sup>2</sup> and P. Suresh<sup>3</sup>

<sup>1</sup>Department CSE, Vel Tech HighTech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai, 600062, Tamilnadu, India

<sup>2</sup>Department of Physics, Vel Tech Multitech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai, 600062, Tamil Nadu, India

<sup>3</sup>Department of ECE, Veltech Rangarajan Dr Sagunthala R & D Institute of Science and Technology, Chennai, Tamilnadu, 600062, India

\*Corresponding Author: R. Karthikeyan. Email: rkarthikeyan20vel@gmail.com

Received: 19 March 2021; Accepted: 05 May 2021

**Abstract:** Digitization has created an abundance of new information sources by altering how pictures are captured. Accessing large image databases from a web portal requires an opted indexing structure instead of reducing the contents of different kinds of databases for quick processing. This approach paves a path toward the increase of efficient image retrieval techniques and numerous research in image indexing involving large image datasets. Image retrieval usually encounters difficulties like a) merging the diverse representations of images and their Indexing, b) the low-level visual characters and semantic characters associated with an image are indirectly proportional, and c) noisy and less accurate extraction of image information (semantic and predicted attributes). This work clearly focuses and takes the base of reverse engineering and de-normalizing concept by evaluating how data can be stored effectively. Thus, retrieval becomes straightforward and rapid. This research also deals with deep root indexing with a multi-dimensional approach about how images can be indexed and provides improved results in terms of good performance in query processing and the reduction of maintenance and storage cost. We focus on the schema design on a non-clustered index solution, especially cover queries. This schema provides a filter predication to make an index with a particular content of rows and an index table called filtered indexing. Finally, we include non-key columns in addition to the key columns. Experiments on two image data sets 'with and without' filtered indexing show low query cost. We compare efficiency as regards accuracy in mean average precision to measure the accuracy of retrieval with the developed coherent semantic indexing. The results show that retrieval by using deep root indexing is simple and fast.

**Keywords:** Multi-dimensional indexing; deep root; hashing; image retrieval filtered indexing

## 1 Introduction

In memory optimization, major researches on the primary memory database are restarted in the 1980s. Databases were widely used for the processing of transaction based operative data contents. Today, databases will not reporting and analysis on consolidated historic data. This significance leads to complexity in data

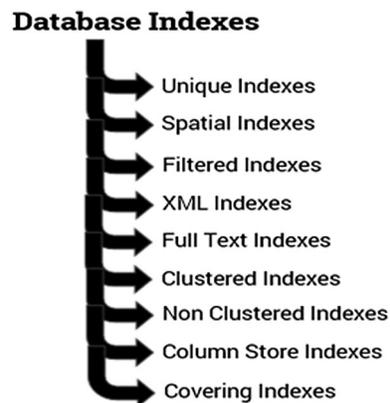


This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

queries also given the increasing demand for efficiency in query processing and memory optimization. This optimization refers to the method of manufacturing an optimal execution plan for a given query, where the optimal plan is with reference to a price function of the memory to be minimized. That plan seeks to supply minimum reaction time and maximum throughput. The good commercial success of database (DB) systems is partly due to sophisticated memory optimization technology. Memory optimization in a large DB has gained meaningful importance because it supports the scaling back of dimensions and the utilization of memory and time taken for any type of query to be processed. Most of these systems were focused on two important factors: 1) performance-critical applications and 2) data criticality in terms of accuracy. Recently, the trends have shifted and become interesting due to the important factors such as memory price drops and multi-core parallelism. Related to this kind of research is a new kind of DB schema compared with the traditional disk-based RDB system. The key focus areas of this research considered (i) organization of data and indexing,(ii) concurrency control, (iii) durability and recovery, and (iv) query processing and compilation.

Fig. 1 represents various database indexes used still now. The highlights in this study are as follows:

- All the index structures discussed so far in the technological trends are mono dimensional, that is, they assume that a single key for the search will retrieve the record of data. However, even if a key contains some combined fields, it will still be considered as uni-dimensional. Thus, we can consider the search key value as concatenated values from F1, F2, ..... and so on.
- The filter on the INDEXES will increase the efficiency of the query process and reduce the utilization of memory compared with a complete tabulated indexing.
- Indexing the text column with the “full text feature” further optimizes performance on the BLOB and XML columns.



**Figure 1:** Indexing Types

## 2 Related Works

References [1,2] suggested the combinational sparsing of semantic features stored as inverted index Vs re-ranking features. The resultant from both techniques provides an optimized solution but overlooks the approach of the indexing technique in storing the resultant sets and data retrieval [3]. suggested the dominant color composition (DCC) signature of the zoned DCD signature extraction scheme. Here, they emphasized the DCD signature which is generated by anticipating places of each sub territory with its own predominant tone on the horizontal and vertical hub after dividing a picture territory into sub areas. The DCC mark is utilized to prune away unessential pictures to a given query before assessing more costly comparability measures on the planned candidate data.

References [4,5] proposed a semantically enhanced information retrieval model for fine gaining the indexing and querying scope. Irrelevant words were also filtered out from the key. However, that method needs some enhancement toward incorporating new technique.

References [6,7] developed a local auxiliary color maximum vector pattern (LACMVP) which includes stimulating the color and surface data by taking a primary (red, green, and blue) and a secondary channel (value) from two diverse shading spaces (RGB and HSV). A vector design including the size, sign, and position are determined for the most extreme nearby contrast between the middle pixel and its neighbor from the secondary channel. In addition, LACMVP converts the image features into local vectors along with the maximum edges of the inter-chromatic texture pattern but overlooks the core feature of retrieval efficiency and how value manipulation is achieved.

References [8] constructed a semantic visual feature space by utilizing feature descriptors and attributes (semantic), and that space can reduce the semantic gap by considering semantic attributes and spectral hashed (SH) indexing into a solitary system. This case involves the lenience of coherent semantic indexing (CSI) employed binary codes to gain data retrieval speed with underlying accuracy maintenance. First, the author suggested an intuitive improvement strategy to examine the joint semantic and visual descriptor space. Second, the intermingling of their improvement calculation which ensures a decent arrangement after a specific number of emphases is taken. Third, a semantic visual joint space framework with spectral hashing (SH) is incorporated to locate an optimized answer to search large datasets. The research provided a proactive approach with a major drawback on overlooked concepts such as database indexing and retrieval technique on large images versus small images scenarios.

References [9] suggested a technique which entailed splitting of the internal node with overlapping. This method is an index scheme and a dynamic multi-dimensional one for CBIR. The technique enabled indexing schemes to search efficiently and achieved high dimensional features by decreasing the size of the inside hub data by killing the inner hub pointer. Additionally, this approach parts the inner hub similarly by utilizing the covering split to keep up a height balanced tree.

References [10,11] evaluated a new indexing method for web documents and usual document repository using a new technique called the multi-dimensional indexing approach (MIWD). The first approach is inverted indexing. The second one is compression and conventional indexing for spatial related data images. This approach provides an innovative scheme for indexing and introduced a version according to indexing spatial data for web contents especially for web images. In the R-Tree data structure, the spatial indexed data is encompassed by the minimal bounding rectangle (MBR). This form will continue up to the root node (bottom-up traversal). The root comprises MBR for the overall objects. However, the author provides a more complex model and no explanation was given in the area of image databases.

The major contribution of this study is as summarized below:

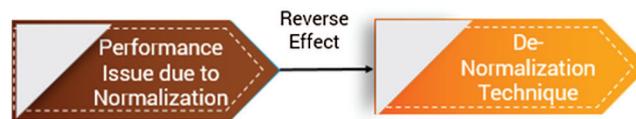
- A novel index scheme to maintain database with low cost.
- A simplified domain specific deep root indexing technique in which the indexing is designed according to the depth of data stored.
- Taking the advantage of a relational database index and the nature of its data structure will improve the data processing capacity to achieve greater accuracy and require less memory.

### **3 Deep Root Indexing**

#### **3.1 Base Knowledge**

Advanced feature extraction followed by filtering and numerous image processing techniques were involved in the current trends in image identification [12]. This development provides an optimized

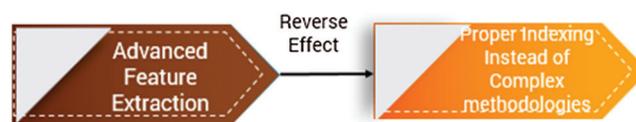
approach to finding relevant images. Comparing or considering two factors involves image retrieval versus performance in image retrieval. We should consider both factors. Thus, the solution provided could not satisfy the exact requirement. Fig. 2 states the reverse effects causes during normalization of features. In turn, a key enforcement toward image retrieval to fetch the image properly [13] becomes a prominent methodology in which the relevant images should be named properly and indexing must be correctly implemented.



**Figure 2:** Reverse effect to de-normalize

### 3.2 Proposed Knowledge

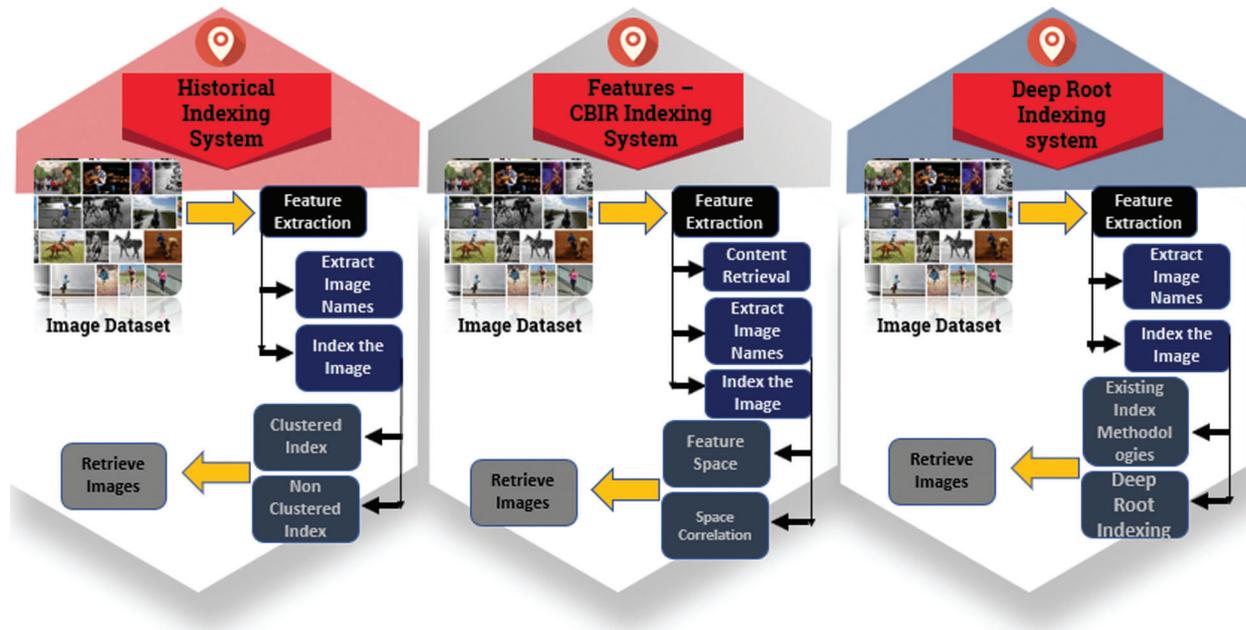
In Fig. 3 states the importance of proper indexing and shows advanced extraction techniques depends on proper indexing instead of complex indexing methodologies. The attribute highlight for each picture comprises of the reactions to classify all the characteristics from the binary classifier. Pictures are associated in the independent R-Tree flow and are dependent to the two sets of sorted features. By referring the correlations mutually at their own feature space, a coordinated improvement involves binding semantic low level visual features and mapping every image into an idle high level feature space. Encoding the visual related information in terms of both high- and low-level features [14] were also considered in this space to save the comparability continuity between the visual and quality spaces so as to forestall over fitting of boisterous and wrong semantic ascribes and smother certain uproarious or excessive highlights in the first space. Investigation of a few off-the-rack hashing procedures for smaller double coded highlights the new feature space and guarantees that comparative images lie on the nearest binary codes. For a given sample query, comparative neighbors have been recovered by searching the resultant image within a smaller Hamming distance [15]. The Fig. 4 gives comparison among indexing techniques from historical perspective to deep root indexing.



**Figure 3:** Indexing rather than using the complex method

Semantic hashing is one of the most opted techniques and is considered as a premise to insert high-level multi-dimensional components into a low-level Hamming space. Thus, the hashing strategy is viewed as speed and memory productive and has been employed to help in similarity search. For example, in locality-sensitive hashing (LSH) [16], each bit from the code is determined by irregular projected variations which are trailed by an arbitrary edge. Moreover, the H-distance among the nearby codes asymptotically imagined the Euclidean between the stored image data [17]. However, the dynamic projection of images is dependent on data and codes might just be inefficient. The reason is for that outcome is that numerous tables with lengthy binary codes are essential for verification. As a solution for this case, the binary re-constructive embedding (BRE) [18] method was considered in terms of hashing capacity dependent on unequivocally limiting the remarks between the Hamming space and measured space. These embedded codes normally struggle from dynamic scale invariance in the given query due to data normalization, a situation which is practically non-negligible for the nearest neighborhood searching

algorithm. To overcome this problem, the spectral hashing (SH) technique was proposed to average the credentials of nearest neighbor searching on the basis of spectral graph partitioning [19]. SH techniques manipulate the bits by the threshold of the eigenvectors of the Laplacian and closeness chart and conveys extensive upgrades as far as various bits are needed to discover comparative neighbors [20,21]. Although the technique is satisfactory, its major flaw is due to its performance constraints.



**Figure 4:** Historical indexing to deep root indexing

### 3.3 Implementation

Indexes play an important role in image retrieval and its associated decision support systems by reducing the cost of resulting complex queries. In choosing an optimized indexing scheme, the factors that should be considered are as follows:

- Analyzing poorly-performing queries;
- Proposing a set of opted indexing mechanisms which will potentially benefit the given query;
- Choosing of meaningful subset from the optimized indexes (from b).

Poor schema in indexing and a lack of effective indexes are the main sources of DB application bottlenecks. Designing efficient indexes is paramount to achieve a good DB and its performance over on an application. This research employs a base index design methodology that contains information on index architectures and best practices on designing effective indexes to meet the needs of an application. In addition to existing index requirements, we develop a deep root indexing technique which further provides the necessary components in terms of performance.

### 3.4 Deep Root Indexing Technique

Filtered indexing along with the included column provides an innovative approach in schema designs and storage. Fig. 5 shows alignment of columns. A filtered index has been applied to index a particular portion of the Tab. 1 as row wise, an approach which will improve the following features

1. The performance of query processing,
2. The maintenance by reducing the cost, and
3. The utilization of opted memory space.

**Conditions for Deep root indexing:**

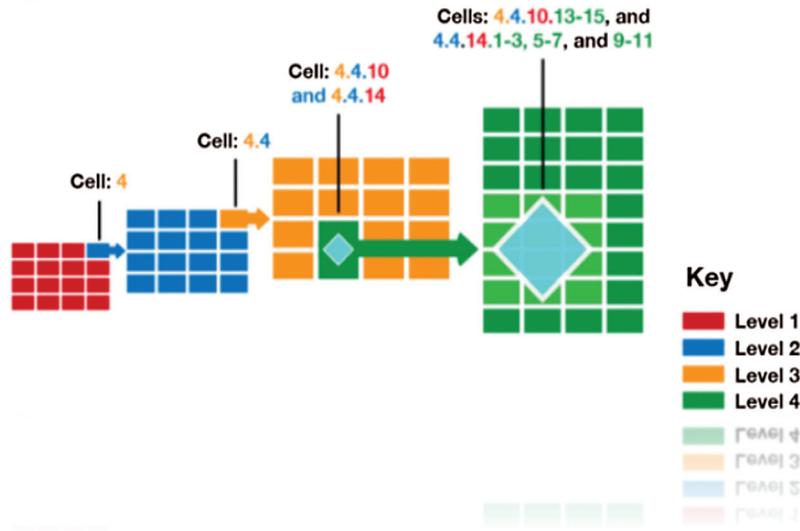
- Constructed as Non-Clustered only
- Views only if they are persisted.
- Not possible to create on an index with full text

**Table 1:** Comparison of clustered and filtered methods

Existing	Proposed
<p><b>Clustered</b> A clustered index provides the functionality which includes row-wise sorting of the data stored in the table. This index has been implemented as an index structure of the B-Tree and supports retrieval in fast based on the key values.</p>	<p><b>Filtered Indexing</b> An optimized non-clustered index solution that is especially well-suited for queries that have been selected from a well-suited subset of data. This index provides a filter predicate to index a portion of rows in the table.</p>
<p><b>Non clustered</b> A non-clustered index is associated with a cluster index or like a heap structure. Each row in the index contains a key value and a row locator. This locator points to the row of the clustered index or key value of heap. The rows are stored in any particular order unless a clustered index is created on the table. This method provides an additional performance along with the clustered indexes.</p>	<p>The performance is in the hands of a DBA who designs the well-designed filtered index. In turn, this approach improves query performance, reduces index maintenance costs, and minimizes index storage costs.</p>
<p><b>Unique</b> A unique index provides the condition of no duplicate values consequently in every row or a content view that is uniquely arranged.</p>	<p><b>Index with included columns</b> A non-clustered index solution which provides the capability to include non-key columns in addition to the key columns.</p>
<p><b>Hash</b> Data accessed through memory hash table. Hash indexes consume a fixed volume of memory which is based on the bucket count.</p>	

#### 4 Indexing Efficiency

Efficiency becomes the major challenge in large-scale image retrieval [22,23]. The cost and complexity are said to be very high. The main reason behind this situation is the result of the process which is in vector pattern with hundreds of multiple dimensions. Compared with the exploited SH method to overcome this challenge, this proposed indexing technique achieves better performance and accuracy. Different numbers of performance evaluators on the non-clustered platforms with INRIA data set are compared.



**Figure 5:** Index with included Column

The following are the SH codes which satisfy the criteria below:

To identify the Hamming distance

$$\|h_a - h_b\|_2 \tag{1}$$

Multiply with  $S_{ab}$  and the minimum will be

$$\text{Min} \sum_{ab} S_{ab} \|h_a - h_b\|_2. \tag{2}$$

Here,  $S_{ab}$  is the similarity between data  $X_a$  and  $X_b$  and is defined as

$$S_{ab} = \exp(-(x_a - x_b)^T M (x_a - x_b)), \tag{3}$$

$$\text{s.t. } h_a \in \{-1, 1\}^k, \tag{4}$$

$$\frac{1}{n} \sum_a h_a = \{0\}^k, \tag{5}$$

$$\frac{1}{n} \sum_a h_{aya^T} = 1. \tag{6}$$

$h_a$  is a binary code corresponding to  $X_a$ .

A state-of-art performance is produced by the observed codes of SH, thereby ensuring the flexibility of the semantic index to gain the necessary retrieval speed and reduce the semantic gap. By contrast, conventional SH valuates visual features and eliminates the attribute features which reduce the semantic gap. The binary set of contents can be computed by training set  $X_a$  points and the needed number of “K” bits. In our example, the extension in the manifold learning is a problem for which we attempt to find a method to evaluate binary codes for a query image, which is a very important process. The Nyström extension of a new image is calculated using the Nyström method. This process requires complete max similarity findings over the entire data set, and the cost for the process is also relatively high.

We compute the same with new samples of data on the basis of an assumption of data distribution [24], and this approach is a similar strategy like what we used for SH. The distribution of data has been assumed to have a multi-dimensional similarity, and one dimension is assumed to have uniform distribution on  $a$  and  $b$ .

The systematic end results of the eigenfunctions  $\phi_k(x)$  and eigenvalues  $\lambda_k$  of the mono dimensional values of the Laplacian functions are expressed as

$$\varphi_k(x) = \sin(\pi/2 + (\pi k/(b-a))x), \quad (7)$$

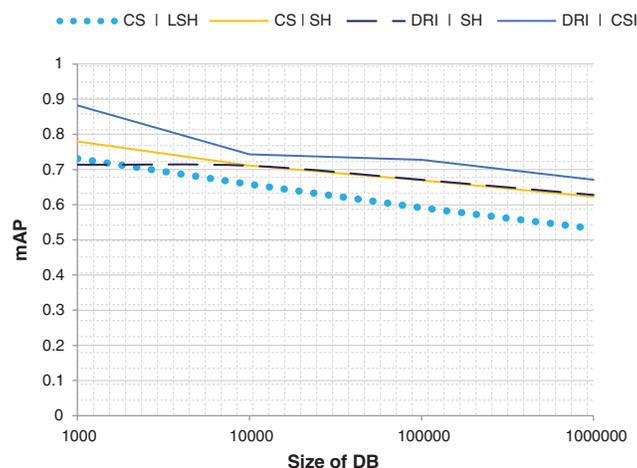
$$\lambda_k = 1 - e - \sigma^2/2 \sqrt{k\pi/b - a}^2. \quad (8)$$

The combination of eigenvalues is obtained and generated by a forming list of eigenvalues of multi-dimensional distribution. The obtained eigenvalues have been sorted to identify the list of k-smallest eigenvalues. We also obtain the hash functions corresponding to the given eigenfunctions. By evaluating the values of the hash function, we can derive the hash code of the given query image in our proposed unified space. The basic ingredients of this kind of research involves the optimal concept of de-normalization for which this technique provides an opted solution to revert back the changes due to excessive normalization which may produce a reverse effect in terms of performance. To subside the effects of performance, we achieved an optimized solution of de-normalizing the data.

By using the deep root indexing (DRI) method, efficient retrieval of the resultant images will be obtained, as the method uses the binary codes for faster comparison similar to the CSI algorithm. Moreover, the DRI algorithm uses a multi-dimensional model of data distribution, an approach which helps in the parallel comparison of data for the efficient retrieval of images. Our proposed DRI method relies on a filtered index with a key column in the existing indexing techniques, and we experimented with different kinds of indexing especially with the SH associated with CSI and LSH as given in Tab. 2. The obtained results have been shown in the graphically in Fig. 6. Although the size of the DB was increased, the proposed approach generates a 13.57% increased optimized result compared with SH and its embedded techniques.

**Table 2:** Semantic hashing table

DB Size	CS   LSH	CS   SH	DRI   SH	DRI   CSI
<b>mAP</b>				
$10^3$	0.731	0.78	0.714	0.882
$10^4$	0.658	0.711	0.711	0.743
$10^5$	0.591	0.669	0.671	0.728
$10^6$	0.532	0.623	0.628	0.671



**Figure 6:** Comparison of the efficiency between image retrieval methods along with the existing indexing and deep root indexing technique on an INRIA data set

Fig. 7 depicts the query and results has been obtained for the given query from high to low priority and for which optimized results are obtained

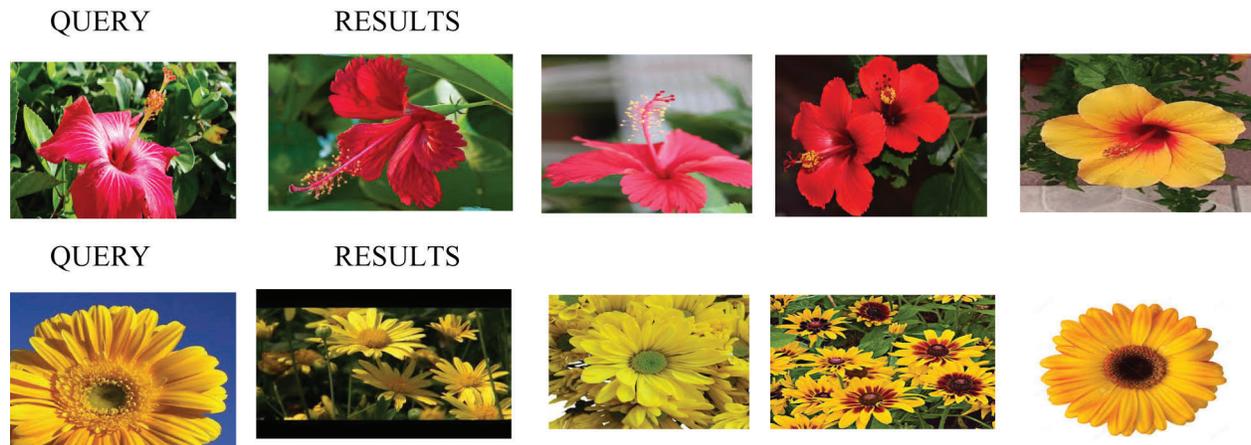


Figure 7: Results for the given query using deep root indexing compared with existing techniques

### 5 Performance Evaluation of Memory Optimization

**Implementation 1:**

WITH(INDEX(NC\_ImageDatabase\_UploadDate))  
 (5000 row(s) affected)

Table 'ImageDatabase'. Scan count 1, logical reads 20660, physical reads 8, read-ahead reads 45, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

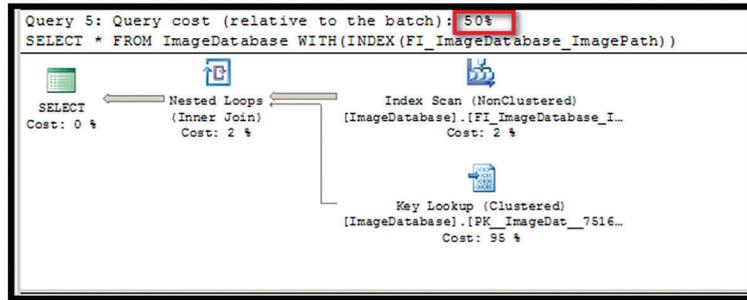
**Implementation 2:**

WITH(INDEX(FI\_ImageDatabase\_ImagePath))  
 (5000 row(s) affected)

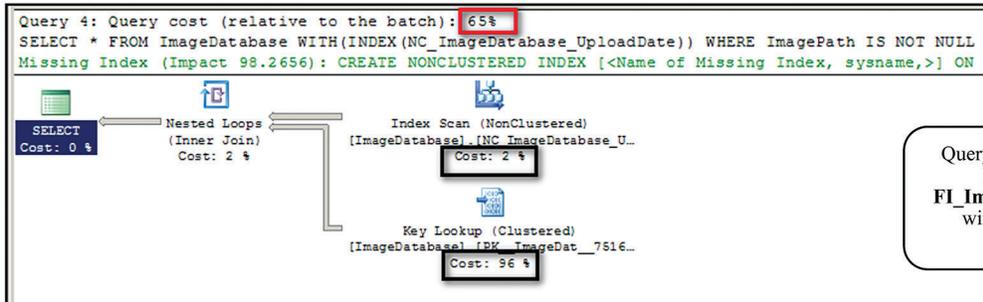
Table 'ImageDatabase'. Scan count 1, logical reads 10345, physical reads 0, read-ahead reads 23, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

Implementing the indexes with the proposed technique achieves the following output reported in Figs. 8 and 9.

Query cost for the execution of Non-Clustered index **FI\_ImageDatabase\_ImagePath** is 50%



index_id	name	type_desc	reserved_page_count	used_page_count	row_count	filter_definition
1	PK_ImageDat_7516F4EC0EA330E9	CLUSTERED	49	46	10000	NULL
2	NC_ImageDatabase_UploadDate	NONCLUSTERED	41	35	10000	NULL
3	FI_ImageDatabase_ImagePath	NONCLUSTERED	33	32	5000	((UploadDate) IS NOT NULL)



Query cost for the execution of Non-Clustered index **FI\_ImageDatabase\_ImagePath** with filtered Index is 35%

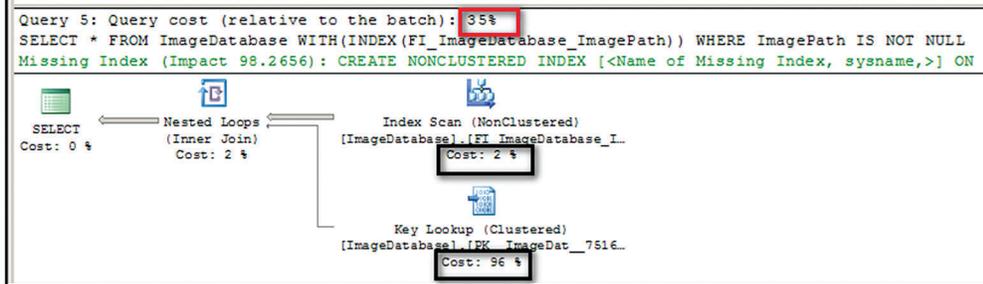


Figure 8: Comparison of the query costs between the filtered index with other indexes

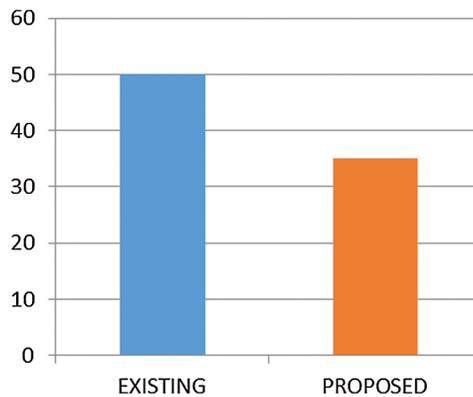


Figure 9: Optimized memory (35%) by DRI

## 6 Conclusion

The proposed methodology provides an optimized solution for image retrieval. This work also presents a model for the query processing through the deep root indexing method. Optimistic and efficient results in terms of memory and accuracy were obtained for the focal area for image retrieval and the data storage related to images (indexing). Thus, integrating this kind of technique in spatial analysis and geographical data can be considered as a future enhancement of this research.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interests:** The authors declare that there is no conflict of interest regarding the publication of the paper.

## References

- [1] T. Vileiniškis and R. Butkienė, “Applying semantic role labeling and spreading activation techniques for semantic information retrieval,” *Information Technology and Control*, vol. 49, no. 2, pp. 275–288, 2020.
- [2] K. T. Ahmed, H. Afzal, M. R. Mufti, A. Mehmood and G. S. Choi, “Deep image sensing & retrieval using suppression, scale spacing & division, interpolation and spatial color coordinates with bag of words for large and complex datasets,” *IEEE Access*, vol. 8, pp. 90351–90379, 2020.
- [3] B. Xu, H. Lin, Y. Lin, K. Xu, L. Wang *et al.*, “Incorporating semantic word representations into query expansion for micro blog information retrieval,” *Information Technology and Control*, vol. 48, no. 4, pp. 626–636, 2019.
- [4] K. Sundara Krishnan, B. Jaison and S. P. Raja, “Secured color image compression based on compressive sampling and lü system,” *Information Technology and Control*, vol. 49, no. 3, pp. 346–369, 2020.
- [5] A. Popescu, G. Ginsca and H. L. Borgne, “Scale-free content based image retrieval (or nearly so),” in *2017 IEEE Int. Conf. on Computer Vision Workshops (ICCVW)*, Venice, pp. 280–288, 2017.
- [6] S. Arslan, A. Saçan, E. Açar, I. H. Toroslu and A. Yazıcı, “Comparison of multidimensional data access methods for feature-based image retrieval,” in *2010 20th Int. Conf. on Pattern Recognition*, Istanbul, Turkey, pp. 3260–3263, 2010.
- [7] S. Jo and K. Um, “A signature representation and indexing scheme of color-spatial information for similar image retrieval,” in *Proc. of the First Int. Conf. on Web Information Systems Engineering*, Hong Kong, China, vol. 1, pp. 384–392, 2000.
- [8] N. Khalili, C. Prasad, M. Vedprakash, S. Chaudhary and S. Murala, “Local auxiliary-color maximum vector pattern: A new feature descriptor for image indexing and retrieval,” in *TENCON 2017 - 2017 IEEE Region 10 Conference*, Penang, pp. 1201–1206, 2017.
- [9] R. Hong, L. Li, J. Cai, D. Tao, M. Wang *et al.*, “Coherent semantic-visual indexing for large-scale image retrieval in the cloud,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4128–4138, 2017.
- [10] M. Kim, G. Kim, D. Kim, J. Park, J. Han *et al.*, “A dynamic high dimensional index scheme for contents based image retrieval,” in *2016 IEEE Int. Conf. on Consumer Electronics-Asia (ICCE-Asia)*, Seoul, Korea (South), pp. 1–2, 2016.
- [11] D. Sotoude and N. Yazdani, “MIWD: Multidimensional indexing for web documents,” in *The 16th CSI Int. Sym. on Artificial Intelligence and Signal Processing (AISP 2012)*, Shiraz, Iran, pp. 167–173, 2012.
- [12] Y. Tang, J. Xu, S. Zhou, W. Lee, D. Deng *et al.*, “A lightweight multidimensional index for complex queries over DHTs,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 12, pp. 2046–2054, 2011.
- [13] Y. Gong and S. Lazebnik, “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval,” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2011.
- [14] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” in *IEEE Sym. on Foundations of Computer Science*, NM, USA, pp. 459–468, 2006.
- [15] B. Kulis and T. Darrell, “Learning to hash with binary reconstructive embeddings,” in *22nd Int. Conf. on Neural Information Processing Systems (NIPS’09)*, NY, USA, pp. 1042–1050, 2009.

- [16] Y. Weiss, Y. Fergus and R. Torralba, "A multidimensional spectral hashing," in *Computer Vision – ECCV 2012. ECCV, Lecture Notes in Computer Science*, vol. 7576, pp. 340–353, 2012.
- [17] L. Zhang, H. P. Shum and L. Shao, "Discriminative semantic subspace analysis for relevance feedback," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1275–1287, 2016.
- [18] R. Zhang, L. Lin, R. Zhang, W. Zuob and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4766–4779, 2015.
- [19] J. Deng, A. C. Berg and L. Fei-Fei, "Hierarchical semantic indexing for large scale image retrieval," in *CVPR 2011*, Colorado Springs, USA, pp. 785–792, 2011.
- [20] C. Ding, J. Choi, D. Tao and L. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 518–531, 2016.
- [21] Y. Wang and G. Mori, "A discriminative latent model of object classes and attributes," in *Proc. of the European Conference on Computer Vision*, Springer, pp. 155–168, 2010.
- [22] L. Torresani, M. Szummer and A. Fitzgibbon, "Efficient object category recognition using classemes," in *Proc. of the European Conference on Computer Vision*, Springer, pp. 776–789, 2010.
- [23] M. Rizkinia, T. Baba, K. Shirai and M. Okuda, "Local spectral component decomposition for multi-channel image denoising," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3208–3218, 2016.
- [24] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer *et al.*, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 931–945, 2011.