International Journal of Engineering Research ISSN:2319-6890)(online),2347-5013(print)
Volume No.5, Issue No.6, pp : 484-488 1 June 2016

Privacy Preserving Content Based Image Retrieval with Load Balanced Framework

Bhagyashree V. Khapli[1], Manjushri A. Mahajan[2]
[1,2]Department of Computer Engineering, G. H. Raisoni College of Engineering, Wagholi, Pune, Maharashtra, India
bhagyashree.khapli2010@gmail.com

Abstract: Privacy-preserved using hash indexed queries and removal of some bits from it but generation of hash on both client and server side increases the computation overhead with delaying process. This paper has designed Load balanced framework by adding third party server who will compute hash for both client and Server without revealing search information.

Keywords — Content-Based Image Retrieval; Robust Sparse Hashing; Privacy-Preserving Content Based Image Retrieval.

I. Introduction

Content-Based Image Retrieval (CBIR) is the application of computer vision techniques to the image retrieval problem, it means, the problem of searching for digital images in large databases. "Content-based" indicates that search analyzes the contents of the image rather than the metadata such as keywords, tags, or descriptions are associated with the image and "content" might refer to colors, shapes, textures, or any other information that can be derived from image itself. Google goggle and Flip kart image search engines are the examples of the Content based Image Retrieval. In Image search engines, user gives one image as a query and gets similar images with the input query as an output. The System checks the content similarity of query image and images in the database. Images with higher similarity above the threshold value are returned as an output. Image search engine is one of the applications in which Content Based Information Retrieval [6] is used. At Most CBIR is applied on the Multimedia database. The main issue in the CBIR was the high dimensionality. Because of which CBIR becomes Time consuming system.

A. Need of the Privacy Preservation in CBIR

Privacy can be explained as No one should know whatever I am doing. In CBIR, some privacy leak issues are found. Generally CBIR is multi-party system, which contains minimum 2 parties, User and Information retrieval system. Both the parties consider unknown to each other. If none of them want to reveal sensitive information then here comes the requirement of Privacy preservation in CBIR. CBIR uses query hashing and bit removal techniques to maintain the privacy but generation of hash and searching on both client and server side increases the computation overhead. We have designed a solution to address this problem. New system includes the third party server who will compute and maintain hash for both client and server without revealing any search information to third party server and improved result accuracy, a Robust Sparse Hashing technique is used.

II. Brief Survey

Paper 1- This paper [1] suggests two layers of protection. First layer protection is; robust hash values are used as queries to prevent revealing original contents. Second layer protection is; the client can choose to omit certain bits in a hash value to further increase the ambiguity for the server. Two robust hash algorithms are used, one is based on random projection that is Locality Sensitive Hashing (LSH); the other is based on the Discrete Wavelet Transform (DWT). It gives Recall up to 0.9 as a performance result.
Advantages:-
1. Privacy enhancement improves the retrieval performance.
2. Designed for Large Scale Databases.
Disadvantages:-
1. Client and Server follow same architecture.
2. Unnecessary computational overload.

Paper 2- In this paper [2], to find nearest neighbor matches to high dimensional data, it propose two most efficient algorithms: the randomized k-d forest and the priority search k-means tree. To scale to very large data sets that would otherwise not fit single machine memory, proposes a distributed nearest neighbor matching framework. All this research has been released as an open source library named as fast library for approximate nearest neighbors (FLANN). It gives precision up to 99 percent as a result.
Advantages:-
1. Scalable for large High Dimensional data.
2. High precision.

Paper 3- The paper [3] presents Discrete Cosine Transform (DCT) hashing technique for creating index structures for face descriptors. A hash index is created, and further queried to find the images most similar to the query image. DCT hashing algorithm has better retrieval accuracy and more efficient compared to other popular state of- the-art hash algorithms. It gives 88 percent retrieval accuracy as a result.
Advantages:-
1. Fast and computationally inexpensive.
2. Outperforms than LSH, E2LSH and KLSH for nearest neighbor recall.

ISSN:2319-6890)(online),2347-5013(print)
1 June 2016

Disadvantage:-
1. Challenging issue is the cost of computing the hash.

Paper 4- The paper [4] presents a new Nearest Neighbor (NN) framework: Robust Sparse Hashing (RSH). This paper approach is inspired by the concept of dictionary learning for sparse coding. For accurate and fast NN retrieval, basic ideas is to sparse code the data by using learned dictionary, and then generate hash codes out of these sparse codes. Results tell that RSH gives different accuracy with different dataset i.e. 92 percent - MNIST dataset, 100 percent - SIFT dataset.
Advantages:-
1. Fast Hash generation.
2. Best accuracy performance on SIFT and MNIST.

Paper 5- In this paper [5] Projected Residual Vector Quantization (PRVQ) algorithm is proposed. The effectiveness of PRVQ algorithm is validated on two kinds of high-dimensional vectors: GIST and vector of locally
aggregated descriptors (VLAD). PRVQ outperforms existing techniques, for example product quantization (PQ), transform coding (TC), and Residual Vector Quantization (RVQ). It gives 30 ms per vector as a result of search time/ speed up parameter.
Advantage:-
1. High Accuracy.
Disadvantage:-
1. No Unified framework.

III. Methodology

A. *Existing System*

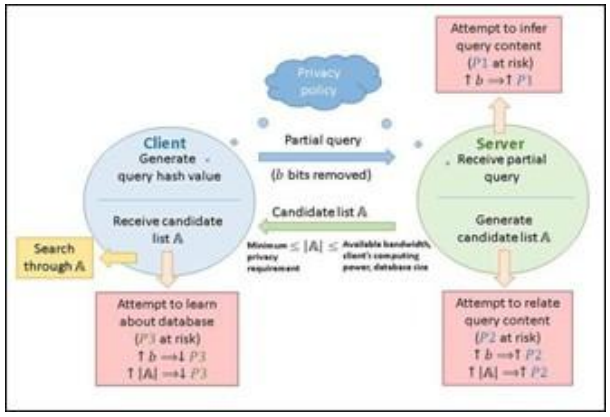

Fig. 1: Existing approach in PCBIR [1]

In existing system an approach is proposed for PCBIR with facility of adjusting the level of privacy. As in fig.1 user queries the database for similar content retrieval. For e.g. user provides an image to server for searching. Prior to this for Privacy Preservation Secure index of the database is generated using hash algorithm. When user querying the object, secure index is generated by removing some bits from the secure index, then query along with position of removed bits is sent to server. Server finds the n no. of nearest neighbor of the query. In return, Hash i.e. Secure Index of the n nearest neighbor is sent to the user. On client side results are searched by using original query and hash list received from server.

B. *Proposed System*

New System is proposed to reduce the computation cost from both sides and with improved result of accuracy. In privacy preserving method, generation of hash on client and server is done commonly on separate server to reduce the burden from client and server. For privacy preservation of the client, multiple image queries are used. For improved accuracy and scalability, reduced size of hash is used. For less communication overhead, increased speed of hash generation, Robust Sparse Hashing (RSH) technique [4] is used. The key point is to sparse code the data using a learned dictionary, and then to generate hash codes from these sparse codes for accurate and fast Nearest Neighbor retrieval.

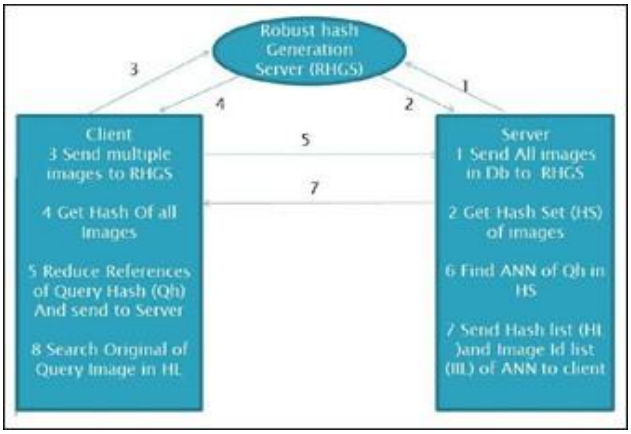

Fig. 2: Architecture of Proposed system

Sparse Hashing- Robust Sparse Hashing (RSH) [4] is a new NN retrieval framework. Sparse code the data using a learned dictionary and then to generate hash codes out of these sparse codes for accurate and fast NN retrieval.

Dictionary learning- Generally dictionary is the basis set of containing image vectors. Dictionary formed by using training data to adapt solutions for the different possibilities of the problem such as image de noising etc.

Sparse coding- Sparse coding is defined as learning complete set of vectors to represent input vectors i.e. sparse coding is the representation of image into sparse format by using some learned dictionary. Sparse coding goals to construct succinct representations of input data sparse coding techniques have been widely used in some applications like image processing, audio processing, visual recognition, clustering and machine learning.

ISSN:2319-6890)(online),2347-5013(print)
1 June 2016

IV. Algorithms

A. *Algorithm for Proposed system*

As shown in fig.2, the proposed system will perform in following way.

1. Start
2. The server containing the information to search will send images from database to the hash generation server.
3. Robust Sparse Hash server (RHGS) will generate the hash table of the received images.
4. Call Hash generation Algorithm.
5. Client wants retrieval of desired contents related of query image, for privacy purpose it will send the image with multiple images to the third party hash generation server i.e. (RHGS).
6. RHGS will generate hash set of the queried images and return it to the client.
7. Client reduces the references of query hash (Qh) i.e. choose the hash query of the search query image and send it to the search engine server.
8. Search engine server will find Approximate Nearest Neighbor (ANN) of received query hash (Qh) in hash set (HS).
9. Server will send the hash list (HS) and corresponding image ID list of found ANN to the client.
10. At client side, client will search the original and similar of query image in hash list.
11. End.

B. Algorithm for Hash Generation

1. RHGS Server extracts the features from received images.
2. On the basis of extracted features a dictionary will be learned.
3. Each data vector is sparse coded using this learned dictionary.
4. The indices in the dictionary for each active base in the sparse code then used to construct a tuple based hash code for each data vector.
5. Tuple based hash code is used for hashing and sends the hash sets of images back to the server.

V. Mathematical Model

Let, S is the System having Input, Processes and Output. It can be represented as,

S = { QI, X, A, P, O}

Where,

QI is a set of all Query images as a inputs
O is a set of all outputs given by the System,
P is a set of all processes in the System.

X= {x1, x2.. xm} Set of images at server side database

Where,

x: Number of Images x1, x2...xm

A= (a1, a2... ak)

Where,

a: Filtered images a1, a2... ak (sparse code)

P = {P1, P2, P3, P4}

P1 Generate the Secure Index or Hash using Robust Hash Algorithm of the image. Robust sparse hashing algorithm using dictionary learning and sparse coding is used. Input of this process is X and output is O1 and O2

Dictionary Learning-

$$\frac{min}{a,\emptyset}\sum_{i=1}^{m}\left(\left\|x^{(i)}-\sum_{j=1}^{k}a_j^{(j)}\emptyset_j\right\|^2-\lambda\sum_{j=1}^{k}|a_j^{(i)}|\right)$$

.....(1)

Output is stored at O1.

P2 - Sparse representation using previously learned dictionary and generate hash code

Output is stored at O2.

$$\frac{min}{a}\left\|x-\sum_{j=1}^{k}a_j\emptyset_j\right\|^2+\lambda\sum_{j=1}^{k}|a_j|$$

.....(2)

P3 - This process removes some bits from the O2 and also stores the positions of removed bits. Number of bits (N) to be removed decided on the basis of level (L) of the privacy preservation.

Output is stored at O3.

$L \alpha N$ .....(3)

P4 Search nearest neighbor of the O3 from A.

Hamming distance between images is evaluated by summation of hamming distance between all respective sub hashes. Output is stored at O4.

$D(H1,H2) = \sum n\ i = 0\ d(h1i, h2i)$ .....(4)

Where,

h1i and h2i are the i^{th} sub hash of H1and H2 resp.

Content identification at client site using O3 and O1.

O = {O1, O2, O3, O4}

O1 Hash generated of Image using P1 i.e. Dictionary of bases 1.

O2 Representation of sparse code [a1, a2...ak] of image x.

O3 Updated Hash from P3 and locations of the removed bits.

O4 Nearest neighbor list of O2.

VI. Results and Tables

Table 1. Changes in precision and recall according to number of omitted bits

Nk (No. of omitted bits)	**Precision of PCBIR using DWT**	**Recall of PCBIR using DWT**	**Precision of PCBIR using sparse code**	**Recall of PCBIR using sparse code**
2	0.81	0.76	0.83	0.78
4	0.79	0.77	0.81	0.79
6	0.75	0.79	0.77	0.81
8	0.72	0.81	0.73	0.83

ISSN:2319-6890)(online),2347-5013(print)
1 June 2016

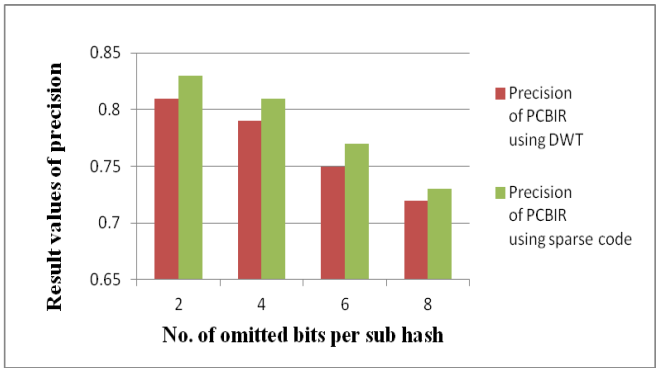

Fig. 3: Changes in Precision according to number of omitted bits

Dataset of 1000 images from public domain image collection will be used for experiments. SIFT feature of image will be used to represent the image. Discrete wavelet transforms (DWT) based hash generation [1] and Dictionary learning based sparse code of image will be used for hash generation of the image. In proposed system hash generation process will be done at robust hash generation server therefore low processing power machines (Smart phones) can be used as client machines. At the time of indexing at CBIR server, hash generation will be done at robust hash generation server. Number of bits (Nk) to be omitted from query hash depends on the requirement of privacy preservation. As Nk are varied precision, recall and number of results are varied. Precision, Recall and Number of results will be compared as shown in table 1. Values in table are taken arbitrarily, actual values will prove, which Hash generation technique is more effective.

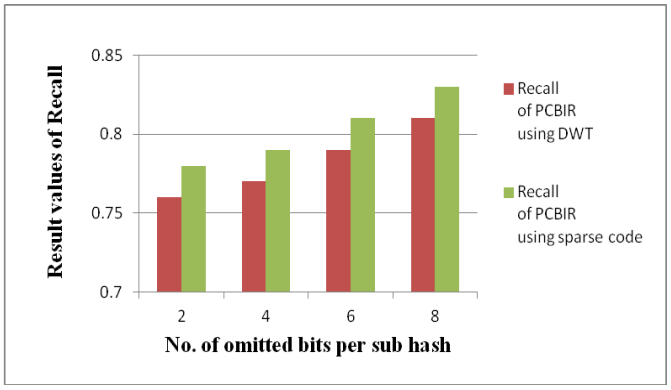

Fig. 4: Changes in Recall according to number of omitted bits

Table 2. Average Number of Candidates Per Query

Nk (No. of omitted bits)	Result Database percent of PCBIR using DWT	Result Database percent of PCBIR using Sparse codes
2	0.25	0.30
4	0.50	0.60
6	1.50	1.25
8	4.60	4.25

As number of omitted bits is increased number of results returned is increased exponentially. Accuracy of the each retrieval will also calculated, accuracy is ratio of number of relevant images retrieved to the total images retrieved.

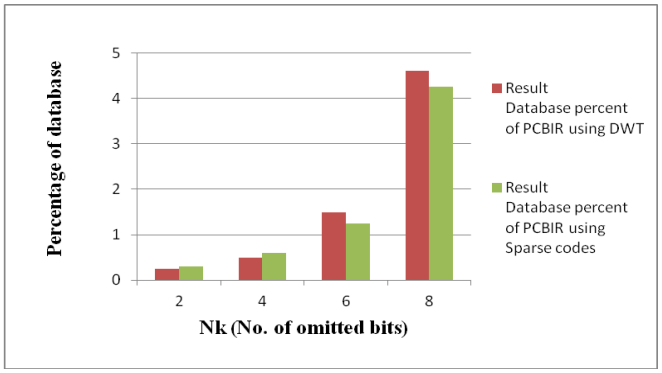

Fig. 5: Average number of Candidate per query

VII. Application of CBIR System

Content Identification System is used by YouTube to avoid duplicate videos, Image search engines like Google goggle for Image Searching, E-Commerce websites like Flip kart [8], Medical diagnosis, Trademark creation and Identification.

VIII. Conclusion

This paper addressed the limitation in existing system and overcome the problem of computation overhead and performance by alleviating load of hash generation process to the third party server with preservation of privacy. This paper used a robust sparse hashing algorithm for quick and robust hash generation. Our approach makes content based image retrieval process accurate without revealing any information of interest.

References

i. *Li Weng, Member, IEEE, Laurent Amsaleg, April Morton, and Stphane Marchand- Maillet, A Privacy Preserving Framework for Large-Scale Content-Based Information Retrieval, IEEE Transactions on Information Forensics And Security, vol. 10, no. 1, January 2015.*
ii. *Marius Muja, Member, IEEE and David G. Lowe, Member, IEEE, Scalable Nearest Neighbor Algorithms for High Dimensional Data, IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 36, no. 11, November 2014.*
iii. *Mehran Kafai, Member, IEEE, Kave Eshghi, Bir Bhanu,Fellow, IEEE, Discrete Cosine Transform Locality-Sensitive Hashes for Face Retrieval, IEEE Transactions on multimedia, vol. 16, no. 4, June 2014.*
iv. *Anoop Cherian, Suvrit Sra, Vassilios Morellas, Nikolaos Papanikolopoulos, Efficient Nearest Neighbors via Robust Sparse Hashing, IEEE Transactions on vol. 23 , Issue: 8, 2014*
v. *Benchang Wei, Tao Guan, and Junqing Yu Huazhong University of Science Technology, Projected residual vector quantization for approximate nearest neighbor (ANN) search, Published by the IEEE Computer Society, 2014.*

ISSN:2319-6890)(online),2347-5013(print)
1 June 2016

vi. *M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, Content-based multimedia information retrieval: State of the art and challenges, ACM Trans. MultimediaComput., Commun., Appl., vol. 2, no. 1, pp. 119, Feb. 2006.*
vii. *Boufounos, P., Rane, S., Secure Binary Embeddings for Privacy Preserving Nearest Neighbors, IEEE InternationalWorkshop on Information Forensics and Security (WIFS), November 2011.*
viii. *Rane, S.; Boufounos, P.T.; Privacy-Preserving Nearest Neighbor Methods: Comparing Signals without Revealing Them, IEEE Signal Processing Magazine, February 2013.*
ix. *Z. Erkin et al., Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing, EURASIP J. Inf.Secur., vol. 2007, p. 20, Dec. 2007.*
x. *R. L. Lagendijk, Z. Erkin, and M. Barni, Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation, IEEE Signal Process. Mag., vol. 30, no. 1, pp. 82105, Jan.2013.*
xi. *Peter, A. et al. Privacy-Preserving Architecture for Forensic Image Recognition IEEE Conference 2012*
xii. *S. Rane and P. T. Boufounos, Privacy-preserving nearest neighbor methods: Comparing signals without revealing them, IEEE Signal Process. Mag., vol. 30, no. 2, pp. 1828, Mar. 2013.*
xiii. *Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, Privacy-preserving face recognition, in Proc. 9th Int. Symp. Privacy Enhancing Technol. (PETS), 2009.*