International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

A Fast, Secure, Efficient Image Retrieval Framework with user Feedback Support based on Color Features

Nitish Bhattacharjee
Executive M. Tech, Computer Science and Engineering
IIT (ISM), Dhanbad
Kolkata, India

***Abstract*—While designing an image retrieval framework based on content based techniques, a critical aspect is that of transfer of visual data. It opens up the Pandora's box of data privacy issues as well as retrieval performance bottleneck due to added network data transfer latency. The approach suggested here, elaborates enhanced privacy protection scheme. Firstly, by conducting search based on robust hashed values of features extracted from images to prevent revealing original content; secondly, by omitting random bits (both of length and position) from the search client's query hash to increase ambiguity for the image database server. It also lessens the network latency by limiting the server-client data transfer to variable sized candidate image sets. The search algorithm is made effective by using a combination of both local and global color features. So that even the local spatial information is not lost. To lessen computational complexity during search fusion of fuzzy color histogram with block color moment has been utilized to decrease the color feature dimension. Here a basic Relevance Feedback module is incorporated to capture the users' feedback on retrieval results and in turn improve, return better results to users.**

Keywords—Image Retrieval; Feature Extraction; Color Features; Fuzzy histogram; Relevance Feedback; Image Hashing; Color moment, Data Privacy.

I. INTRODUCTION

Multimedia technology and digital image databases are trending nowadays resulting in rapid growth in size of database, quality of images and variety of image obtaining sources. Hence for usage, there is an inherent demand for efficient image retrieval. There are two hurdles though, 1. the risk of privacy leakage and 2. computational complexity. Image retrieval should be secure and fast, i.e. relatively unaffected due to the network latency. These two aspects should be considered very carefully while designing any approach for Image retrieval. Here I am considering such retrieval based on the content of the images only, i.e. Content Based Image Retrieval or, in short CBIR. Three properties - color, shape and texture are said to be content of an image. Thereby, CBIR is a strategy of recovering similar images w.r.t. the content of a supplied image. In system described in this paper. I have considered an environment, where the image database owner (remote storage), database user (search client/ query user) are different parties, not necessarily trusting each other. Hence, the privacy issues. The followings are the key players in this environment: a private database, a private query, a private CBIR technique. The common approach to solve the privacy problem is designing a retrieval algorithm on an encrypted search domain after storing images in encrypted format. [1], [2] As such an approach relies on complicated cryptographic computation, they are costly. My approach inclines toward SRR [3]. Hence, can be used with large databases, has privacy cover and adjustable control for both privacy and computation cost. It is essentially an SRR with robust hashing as a key component.

The proposed CBIR technique uses robust hashing for privacy, using color feature from images. To begin with, image features after extraction are normalized and hashed into a binary vector. Users are allowed arbitrary bits' omission of random length & positions. Thereby, query user has option to choose privacy Vs search speed trade-off. Once, the query is sent, image database calculates the possible candidate matches and returns them. The designed clients then trim down the final search result based of content similarity matching of a fusion of fuzzy feature [4], decreasing computational time. I also incorporated a Relevance Feedback module to capture the users' feedback on retrieval results and then re-sort, update and return them as final results to the users.

II. RELATED WORKS

Content based image retrieval is a much studied topic. Its importance is felt when one considers the impact it has of various fields like digital image processing, medical imaging, diagnostic radiology, defense monitoring etc. Most of the articles I reviewed are based on color and texture features. Analysis of some of them are discussed below:

A. On Color Features

Sharma, Rawat & Singh [5], 2011, discussed the importance of color histogram for image database indexing and retrieval. In this process, all image pixels are counted and the track of color distribution is kept by the association of each quantized color value with a specific bin. They advise to check similarity of images through comparing obtained histogram outputs by intersecting them. This image descriptor is both simple to describe and easy to compute.

The work performed by Mangijao & Hemachandran et al. [6], 2012, suggests improving the discriminative power of color histogram indexing techniques, by dividing image horizontally into three equal non-overlapping regions. Then extract first three color moment from each of these three regions, to store a 27 floating point numbers in the index of the image.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Stricker & Orengo et al. [7], 1995, long back provided the algorithm to calculate color moments, and proved that image's color distribution can then be interpreted as a probability value which characterizes its color moments.

B. On Outsourced Image Privacy Aspects

The Earlier approaches for the support of outsourced storage, search, and retrieval of images can be broadly divided into two classes: based on Searchable Symmetric Encryption (SSE) and based on Public-Key Partially-Homomorphic Encryption (PKHE).

Z. Xia et.al [14], 2015, SSE-based solutions. Clients encrypts data and create encrypted index, before outsourcing it. Both encrypted index and data are outsourced. This allows searching in an efficient and secure way. The limitations are the need to index and encrypts it locally, entailing additional computational power; transferring additional data to cloud (encrypted index) etc.

Zheng et al. [15], 2015, Other approach is PKHE, schemes such as ElGammal [16] allowing additive and multiplicative freedom in encrypted domains. Clients does pixel by pixel processing of images with a PKHE schemed encryption and cloud indexes encrypted images. Issue with this is greater time and space complexities and limited scalability.

Li Weng, Laurent Amsaleg, April Morton and Stephane Merchand-Maillet [3], 2015, proposed a privacy protecting framework for large scale CBIR using robust hashing instead of encryption. My approach is built upon this very idea.

C. On Fuzzy Features

K. Konstantinidis, A. Gasteratos, I. Andreadis [17], 2005, Talked about replacing the classical color histogram creation with histogram linking. Reducing computationally expensive 3D histograms to one single-dimension histogram. Though it was based on the L*a*b* color space.

Mengzhe Li, Xiuhua Jiang [4], 2016, Talks about a highly effective image retrieval algorithm based on fusion of global fuzzy color feature algorithm and local color algorithm in low feature dimension.

III. PROPOSED SYSTEM & WORKING PRINCIPLE

Here, a scalable CBIR system has been considered. There are two primary entities: 1. Image data owner (search server) and 2. Search Client, or Query User.

A. System Model

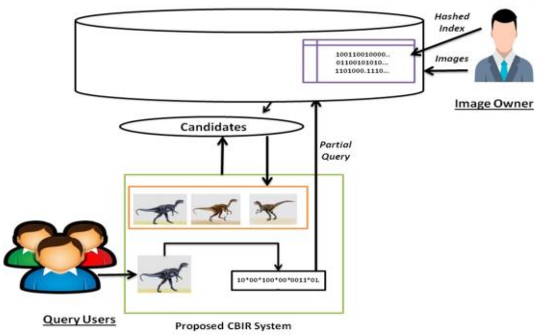


Fig. 1 – System Architecture

If elaborated in a step by step fashion, this discussed paradigm has six parts:

- Submit a partial query to the server (remove details to create ambiguity).
- An extended query list is created on the supplied partial query (calculating all possible combinations for the missing binary bits).
- The server performs a search with the extended query list, and sends back all matching items (it is called the 'Candidate list').
- The client matching against received results using original query and the fuzzy features.
- The client provides relevant feedback if he/she is not happy with the search performance.
- Take account of the feedback while a similarity match check is performed again with modified parameters (here, I used a simple statistical measure i.e. mean feature vector of original matches to further perform refined search for new matches)

In this approach, the server could narrow down search scope using partial query. Whereas it becomes difficult for the server to infer the original query. The framework makes sure **|Candidate Set|** is large but also client should be able to find the final matches. Client is presented with the option to choose how much ambiguity to introduce through partial query. Hence, the size and the diversity of Candidate set can be controlled.

B. Attack Model

Thinking from the query client's perspective, server 1) should not know original query content, or 2) query category. Fulfilling the first ask is tougher. On the other hand, image server should be assured that client doesn't know too much information about its content, or hierarchy of indexing.

There are two steps where image server may derive something about the query: A. While receiving the query hash (denote client's privacy here as Pc1), B. While returning candidate set (denote client's privacy here as Pc2). Server privacy is represented as Pc3. If length of candidate set is |A|, then measures and inter-relations between privacy and |A| are as shown below:

Min. privacy requirements ≤ |A| ≤ power of client, |database|				
$\|A\| \propto P_{c2}$	$\|A\| \propto 1/P_{c3}$	$P_{c1} \propto P_{c2}$,	$P_{c1} \propto 1/P_{c3}$	$P_{c2} \propto 1/P_{c3}$

- For a good system all of $P_{c1,}$ P_{c2} and P_{c3} should be sufficiently large. In the designed system, there is option for user to choose how many bits to omit from the original query hash. For each case, bits are omitted across various sub-hashes before concatenating it to create the final partial query hash. Options are 5, 7, 9, 11 and 13.
- Also, it has been considered here following Weng et. Al. [3], that images of similar nature has similar hash values if generated with same features. CBIR generally not only targets exact matches during search, but nearest neighbors too. Hence, while generating candidate list comparisons have been performed for different hamming radius 'r'. As per hashing theory if this r ≥ 1, then the process of search is called 'multi

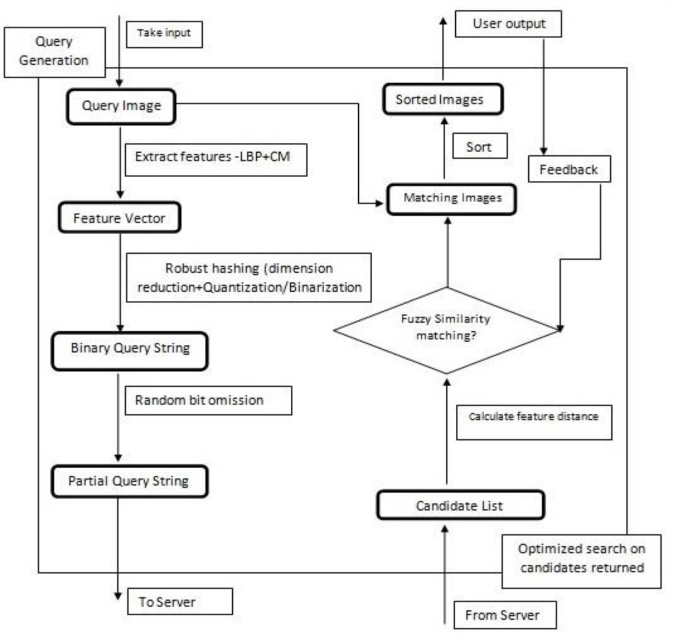
probing'. Which is what is being performed in the system discussed using r = 5 or 6 mostly.

A specific attack using majority voting has been considered here, where a curious server can and will try to predict the query category judging majority presence in candidate list. Details of the attack are listed in a later section.

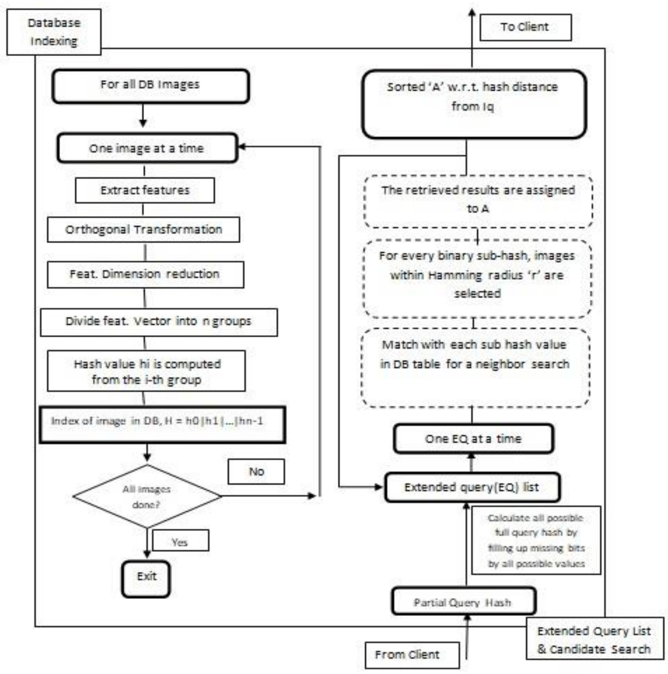
C. Workflow

For easy understandings, workflow of the discussed framework is shown from the entity standpoint, as separate flow charts:

From query user's end:



From image owner's end:



D. Fuzzy Feature Creation

Within the pool of candidates from server (image owner), I am performing an optimized search. Color histogram (in HSV) and color moment (in RGB) have limitations. They ignore local spatial information, reducing precision of retrieval. Plus, HSV color histogram feature has a high dimension of 256, increasing complexity of similarity calculation. Hence, here I reused the improved algorithm introduced by Li, Jiangthat, to reduce the dimension of color feature and to combine comprehensive color information Steps for the fuzzy fused feature creation are,

- Divide query & candidates into blocks of 40 × 40.
- Transform all 1600 spaces from RGB color model to HSV (where S, V belongs to [0,255]).
- Obtain average values of those spaces for all those images.
- Apply fuzzy filtering of 10 bins through three HSV channel deriving 10-bin color histogram (black, grey, white, red, orange, yellow, green, cyan, blue, magenta)
- Strengthen acquired 10-bin histogram with further fuzzy filtering:
  - Each color (except black, grey, white) divided into three levels – deep, medium, light on basis of S, V Channel
  - Total bins = ((10-3) *3) + 3) = 24
- Get a 24-D fuzzy color histogram (FCH).
- Create closely related block color moment (BCM) with method of average division.
  - divide image to 3 × 3 sub-blocks
  - for each of such sub-blocks,
    - calculate first three order color moments
    - arrange color moments by the order
  - get an extended 81-D color moment
- The problem of local color information loss, gets resolved.
- Combine FCH with BCM to integrate HSV & RGB color model and generate comprehensive feature of dimension 105 (24+81).

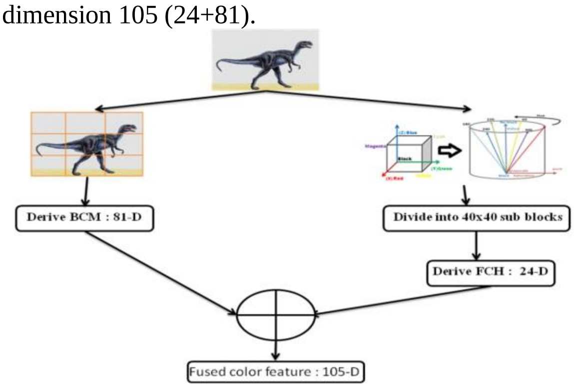


Fig.2 – Fuzzy feature creation

Note : Due to curse of high dimensionality this feature related operation in computationally expensive, hence is it only performed over the candidate set and not all whole image database.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

IV. IMPLEMENTATION & ALGORITHMS

Each algorithm details a particular sub-functionality provided in the paradigm,

A. *Extract Features For Hashing*

Input: Database of Images.
Output: A file storage filled with extracted feature vectors.

Begin:
For all images in the database provided -
Calculate color histograms for red channel as redHist.
Calculate color histograms for red channel as greenHist.
Calculate color histograms for red channel as blueHist.
Calculate feature vector f = [redHist greenHist blueHist]
End For
Save the feature vectors in a file storage.
End.

B. *Create Partial Query Hash*

Input: Query Image Feature
Output: Partial Query hash.

Begin:
Reduce dimensionality of the feature vector.
Divide the residual feature vector into n groups.
For each such group
For each feature bit
If value of the bit >= group's mean value
Then Set value of the bit = 1
Else
Set value of the bit = 0
End If
End For
End For
Append, binarized feat. vector groups together to generate binarized query vector.
Omit random multiple positions value and replace them with '*' to get partial query hash.
End.

C. *Create Image DB Index*

Input: Database of Images.
Output: Indexed Image Server Database.

Begin:
For all images in the database provided
Reduce dimensionality of the feature vector.
Divide the residual feature vector into n groups.
For each group
Compute hash value hi from the i-th group
End For
Index of image in DB, H = h0|h1|...|hn-1
End For.
End.

D. *Image Server Candidate Search*

Input: Partial Query Hash.
Output: Candidate Set A.

Begin:
Create Extended query(EQ) list, calculating all possible full query hash by filling up missing bits by all combinations of possible values (for n missing bits 2n values in EQ)
For one value in EQ list at a time
Match with each sub hash value in DB for nearest matches
For all sub-hashes
neighbors within 'r' Hamming distance are picked
Retrieved objects for all sub-hashes are put to list A
End For
End For
Sort A by the hash distance from the value of query-hash
Return A to Query Client.
End.

E. *Query Client Selective Searcharch*

Input: Query Image (Iq), Candidate Set of Images (A)
Output: Top 20 closest matches for Iq.

Begin:
For all images (A+Iq)
Extract 105D fused fuzzy features
End For
Save the feature vectors in a file storage.
Let feature vector of Iq be fq.
For all images Iv in A, let feature vector of Iv be fv
Calculate Euclidean distance between (fv, fq)
End For
Sort all Iv - s in A, according ascending order of Euclidean distance.
Return first 20 Iv - s to user.
End.

F. *Relevance Feedback Processing*

Input: Actual matching images (Mq) as suggested by user, Candidate Set of Images (A)
Output: Top 20 closest matches updated for Iq (original query).

Begin:
Count number of images in Mq, say C.
For all images in Mq,
take sum of all the feature vectors to create Fq.
End For
Calculate average Aq as Fq/C.
Consider Aq for an assumptive image Iav. (Iav has central tendency of all matches)
Call Query Client Selective Search (Iav, Candidate Set of Images (A))
Return output received from this call.
End.

V. RESULTS

My primary goal of design was to create a functioning image retrieval scheme for -

- Similarity retrieval.
- Establish bias if any, between # of bits omitted from query and candidate set size.
- Protect some privacy of image data. I have only focused on content confidentiality and not about non-detectability or unlinkability.
- Provide search client option to submit feedback for better retrieval accuracy.

The below elaborated results are generated following experiments using Matlab R2018a on a machine having Intel (R) Core i3-5005U CPU @ 2.00 GHz, 4 GB RAM, 64 bit, Microsoft Windows 10 OS. The paradigm has been tested on the Corel-1K image database [21], freely available on the Internet. It contains images of 10 categories, each with 100 images.

Samples of each category –

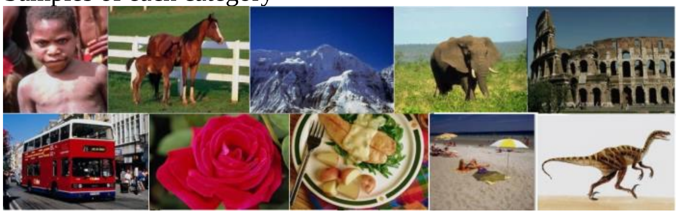


Fig.3 – Tested Image Categories

A. *Sample Result*

First Search Respone

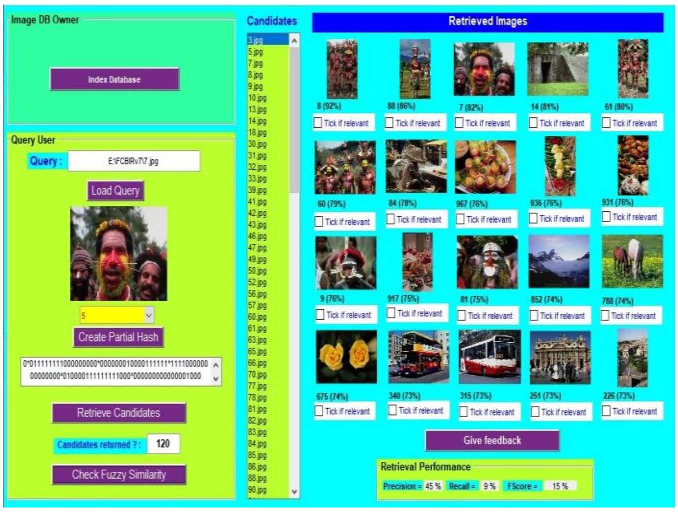


Fig.4 – African People image search

Response after user's feedback

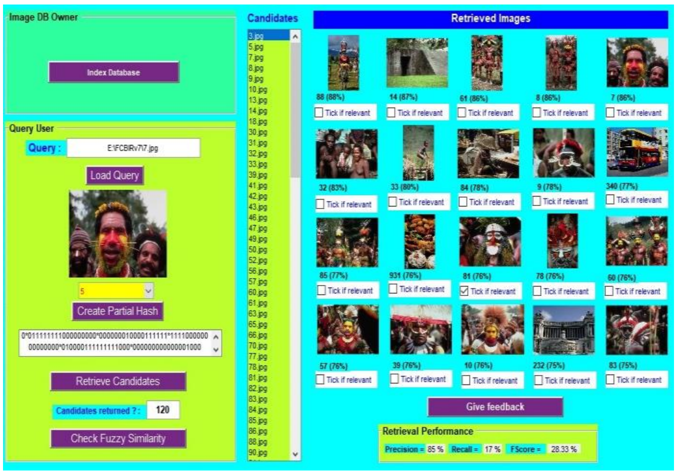


Fig. 5 – African People image search after feedback

B. *Retrieval Performance*

To perform a quantitative analysis of retrieval, I used the following metrics:

Precision (Pr) - # of relevant images retrieved (A) divided by # of searched images (B) from the image DB.

Recall (Rc) - # of relevant images retrieved (A) divided by relevant images (C) present in the image DB.

F-score/F-measure (Fm) - A combined metric providing overall accuracy, as shown below.

So, mathematically,

$Pr = A / B$, $Rc = A / C$, $Fm = ((2*Pr*Rc) / (Pr+Rc))$.

The results are shown in a comparative fashion [9], [10], [11], [12], [13], [20], [21].

Table 1: Precision comparison with existing CBIR schemes (refer Table-5 here)

Class	Elalami	Poursistani	Irtaza	Walia	Shrivastava	This method
African People	70.30	70.20	65	51	74.8	49.17
Beach	56.10	44.40	60	90	58.20	63.33
Monuments	57.10	70.80	62	58.00	62.10	55.83
Bus	87.60	76.30	85.00	78.00	80.20	75
Dinosaur	98.70	100.00	93.00	100.00	100.00	100
Elephant	67.50	63.80	65.00	84.00	75.10	54.17
Rose/flower	91.40	92.40	94.00	100.00	92.30	53.33
Horse	83.40	94.70	77.00	100.00	89.60	98.33
Mountain	53.60	56.20	73.00	84.00	56.10	15
Food	74.10	74.50	81.00	38.00	80.30	60.83
Average	**73.90**	**74.30**	**75.50**	**78.30**	**76.90**	**62.50**

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Table 2: Recall comparison with existing CBIR schemes (refer Table-5 here)

Class	Elalami	Poursistani	Irtaza	Walia	Shrivastava	This Method
Africans People	15.30	14.04	13.00	10.20	15.00	9.83
Beach	19.80	8.88	12.00	18.00	12.00	12.67
Monuments	18.20	14.16	12.40	11.60	12.00	11.17
Bus	11.60	15.26	17.00	15.60	16.00	15
Dinosaur	9.80	20.00	18.60	20.00	20.00	20
Elephant	15.60	12.76	13.00	16.80	15.00	10.83
Rose/flower	11.80	18.48	18.80	20.00	19.00	10.67
Horse	13.90	18.94	15.40	20.00	18.00	19.67
Mountain	22.80	11.24	14.60	16.80	11.00	3
Food	13.80	14.90	16.20	7.60	16.00	12.17

Table 3: F-Scorel comparison with existing CBIR schemes (refer Table-5 here)

Class	Elalami	Poursistani	Irtaza	Walia	Shrivastava	This Method
Africans	25.1	23.4	21.7	17	24.9	16.39
Beach	29.3	14.8	20	30	19.9	21.11
Monuments	27.60	23.6	20.67	19.33	20.11	18.61
Bus	20.49	25.43	28.33	26.00	26.67	25
Dinosaur	17.83	33.33	31.00	33.33	33.33	33.33
Elephant	25.34	21.27	21.67	28.00	25.00	18.05
Rose/flower	20.90	30.80	31.33	33.33	31.50	17.78
Horse	23.83	31.57	25.67	33.33	29.94	32.78
Mountain	31.99	18.73	24.33	28.00	18.39	5
Food	23.27	24.83	27.00	12.67	26.67	20.28
Average	**25.13**	**23.40**	**21.67**	**17.00**	**25.64**	**20.83**

Following points are clear from these three tables:

- These performances are average in comparison to the existing schemes.
- But, considering the fact of added ambiguity for privacy, then the trade-off seems fine.
- The dinosaur images have provided the most satisfactory.
- The mountains have the worst results.
- There is the difference of structural contents among them.
- This gives some idea about the future scope of this work.

As per time complexity is concerned, the average time taken by the major operations in this framework is listed in table-4. Figure

Table 4: Average time requirements for main CBIR operations

Database	Time in seconds			
	Feature Extraction	Query hash generation (avg.)	Candidate list generation (avg.)	Client search (avg.)
Corel-1K	17.73	0.035	8.63	0.39

Table-4 data seems a bit biased towards 'higher bit omission' scenarios (viz. 11,13 and 15-bit omission). They significantly differ from those of less bit omission scenarios specially for the candidate list generation. Hence, a bar chart comparison of time, against varying bit omission length seems more suitable,

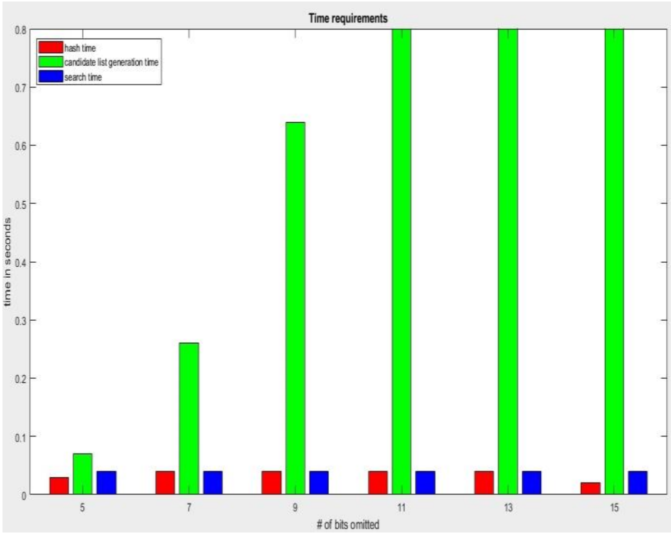


Fig. 6 – Time Requirements

C. Privacy Performance

To focus on the search performance with regards to the varying degree of ambiguity in search query, privacy performance analysis has been done.

Table 5: Candidate set length with varying degree of ambiguity in query (bits)

Category	5	7	9	11	13	15	Max.
Africans	120	122	130	152	143	137	152
Beach	22	27	29	29	33	31	33
Monument	30	34	47	58	65	36	65
Bus	56	102	74	144	127	96	144
Dinosaur	195	195	225	231	233	210	233
Elephant	40	47	44	49	58	50	58
Rose/Flower	120	86	90	104	110	108	120
Horse	58	58	66	59	73	76	76
Mountain	22	26	25	29	33	30	33
Food	169	122	144	144	203	170	203

The Same data, when plotted in graph also verifies the fact that there is no apparent bias for different classes of images in between candidate set length and the number of bits omitted.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

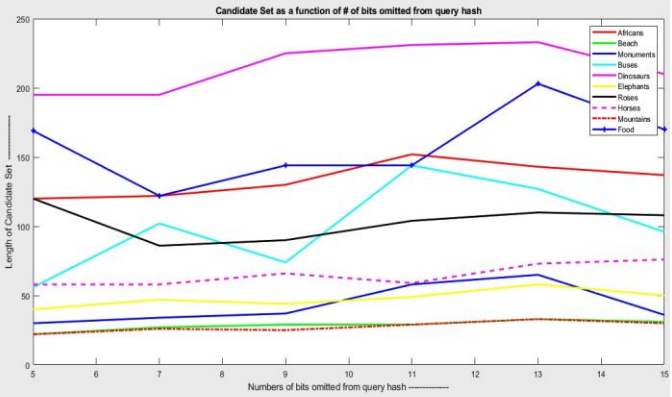


Fig. 7 – Candidate set size Vs # of omitted bits

To do proper estimation of system settings, handing of curious server and client server communication costs, I have listed the details below: -

- System settings: -
  - No. of distinct items (N) = 1000 (Corel-1K database):
  - No. of near duplicates per item (x) = 99.
  - Sub-hash size (l bits) = 32.
  - Groups of sub-hashes (n) = 3.
  - Meta data size (d bits) = 96.
  - No. of omitted bits (b) € [5,7,9,11,13,15].
- Handing curious server: -
  - Wants to guess the query.
  - Has to generate 2b possible values in extended query list.
  - Use large 2b, candidate list generation cost too high.
  - Server would not do such costly operation.
  - Pc2 preserved, but Pc3 decreases with the numbers of omitted bits.
- Client-server communication cost: -
  - These are calculated using the following equation [3], -
    Cost =(Ni*(d+l*n)) for i = 1,2. Here N1, N2 are # of candidates returned for public query (with no omitted bits) and private query with multi-probing respectively.

Table 6: Cost of client server communication

Category	N1	Cost (bits)	N2	Cost (bits)
Africans	143	27456	152	29184
Beach	33	6336	33	6336
Monuments	65	12480	65	12480
Bus	127	24384	144	27648
Dinosaur	233	44736	233	44736
Elephant	58	11136	58	11136
Rose	110	21120	120	23040
Horse	73	14016	76	14592
Mountain	33	6336	33	6336
Food	203	38976	203	38976

From Table – 6, one can say the cost incurred for public and private database are mostly close enough. Or, we can say this privacy requirement doesn't cost much.

D. Majority Voting Attack

To measure resilience against majority voting attack, I am guessing the query category from candidate list result for some uses cases. I am intentionally choosing some cases where images have greater structural contents and other cases where they have lesser structural contents.

Table 7: Candidate list length and majority voting attack

		# of bits omitted						Attack
Category		5	7	9	11	13	15	Success (%)
Dinosaur	|A|	195	195	185	179	183	210	100
	Majority	Dinosaur	Dinosaur	Dinosaur	Dinosaur	Dinosaur	Dinosaur	
Rose	|A|	120	86	43	60	67	108	33.33
	Majority	Rose	Rose	Rose	Rose	Africans	Africans	
Mountain	|A|	22	26	25	29	33	30	0
	Majority	Beach	Beach	Beach	Beach	Beach	Beach	
Food	|A|	169	122	144	144	203	170	16.67
	Majority	Rose	Rose	Rose	Rose	Rose	Food	

See Table-7, that greater structural content fairs better in case of majority voting attack. If practical scenarios are considered, this should mostly be the case with modern high resolution, detailed image capture and processing apparatus.

E. Feedback performance

I have given an option for the user in the implemented model, to specify which returned images are proper to his/her query by clicking of a check box next to each returned images. I am gathering these user selections as relevant feedback (through human interactions) to try improving the search performance. The algorithm used to improve retrieval performance after feedback submission is already discussed in appropriate section. It is nothing revolutionary, just a simplified approach. Following the suggestion of statistical analysis of feedback in CBIR mentioned in a paper [19].

I have used a metric called ROC (Rate of Convergence) [19] along with precision and recall here. To check if the proposed feedback at all improves retrieval performance. ROC is the defined, as the requisite numbers of iterations of feedbacks following which precision of a CBIR system remains constant or the other system parameters do not change considerably. It measures whether the most accurate results possible can be produced fast enough, another practical demand for modern CBIR systems.

Below are the results when only least ambiguous (5-bit omission) query is considered:

Category	First Retrieval		After Feedback		ROC
	Precision	Recall	Precision	Recall	
Africans	50	10	95	19	3
Beach	70	14	70	14	NA
Monument	65	13	70	14	2
Bus	75	15	75	15	NA
Dinosaur	100	20	-	-	-
Elephant	55	11	55	11	NA
Rose	60	12	90	18	3
Horse	100	20	-	-	-
Mountain	10	2	10	2	NA
Food	60	12	60	12	NA

From the above table, it can be said that the relevant feedback algorithm is only effective in some specific cases. The performance of this algorithm is also upper bounded by the original matches present in candidate list. As in the case of mountain, the candidate list only contained two perfect matches. Hence feedback could not improve the performance any further. For Dinosaurs and Horses, the performance was already optimum. Hence, feedback was not utilized.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my advisor Dr. Arup Kumar Pal, Assistant Professor, Department of I would like to express my sincere gratitude to my advisor Dr. Arup Kumar Pal, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad for his continuous support and help in all time. I am also thankful for his motivation, enthusiasm, and immense knowledge. I could not have imagined having a better advisor and mentor.

REFERENCES

[1] 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.

[2] 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. 82–105, Jan. 2013.

[3] Li Weng, Member, IEEE, Laurent Amsaleg, April Morton, and Stéphane 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.

[4] Mengzhe Li, Xiuhua Jiang, "An Improved Algorithm Based on Color Feature Extraction for Image Retrieval", 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2016.

[5] Sharma, Rawat, Singh, "Efficient CBIR Using Color Histogram Processing", An International Journal(SIPIJ), Vol.2, No.1, March 2011.

[6] Mangijao, Hemachandran, "Image Retrieval based on the combination of Color Histogram and Color Moment", International Journal of Computer Applications, Vol 58, No. 3, 2012.

[7] M. Stricker, M. Orengo, "Similarity of color images", SPIE Conference on Storage and Retrieval for Image and Video Databases, vol 2420, pg. 381-392, 1995.

[8] Jain, Muthuganapathy, Ramani, "Content-Based Image Retrieval Using Shape and Depth from an Engineering Database", ISVC, 2007

[9] ElAlami ME. A novel image retrieval model based on the most relevant features. Knowledge-Based Systems, 2011. 24(1):23–32.

[10] Poursistani P, Nezamabadi-pour H, Moghadam RA, Saeed M. Image indexing and retrieval in JPEG compressed domain based on vector quantization. Mathematical and Computer Modelling, 2013. 57(5):1005–1017.

[11] Irtaza A, Jaffar MA, Aleisa E, Choi TS. Embedding neural networks for semantic association in content based image retrieval. Multimedia tools and applications, 2014. 72(2):1911–1931.

[12] Walia E, Pal A. Fusion framework for effective color image retrieval. Journal of Visual Communication and Image Representation, 2014. 25(6):1335–1348.

[13] Shrivastava N, Tyagi V. An efficient technique for retrieval of color images in large databases. Computers & Electrical Engineering, 2015. 46:314–327.

[14] Xia, Zhu, Sun, Qin, Ren, "Towards Privacy-preserving Content-based Image Retrieval in Cloud Computing", IEEE Transactions On Computer Computing, September 2015.

[15] Qin C, Zhang X, "Effective reversible data hiding in encrypted image with privacy protection for image content", J Vis Commun Image Represent, vol. 31, pg. 154-164, 2015.

[16] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” in Adv. Cryptol. Springer, 1985.

[17] Konstantinidis, Gasteratos, Andreadis, "Image retrieval based on fuzzy color histogram processing", Optics Communications, vol. 248, pg. 357-386, 2005.

[18] Singh, Sohoni, Kumar, "A Review of Different Content Based Image Retrieval Techniques", Semantic Scholar, 2014.

[19] Karthik, Jawahar, "Analysis of Relevance Feedback in Content Based Image Retrieval", 9th International Conference on Control, Automation, Robotics and Vision, 2006.

[20] Naushad Varish, Arup Kumar Pal, “Content Based Image Retrieval using Statistical Features of Color Histogram",3rd ICSCN,2015.

[21] Naushad Varish, Sumit Kumar, Arup Kumar Pal, “A Novel Similarity Measure for Content Based Image Retrieval in Discrete Cosine Transform Domain”, Fundamenta Informaticae,2016, pp. 1-27.G. Eason