Al-Mustansiriyah Journal of Science
ISSN: 1814-635X (print), ISSN:2521-3520 (online)
Volume 28, Issue 3, 2017
DOI: http://doi.org/10.23851/mjs.v28i3.77

Research Article

Encrypted Image Retrieval System Based on Features Analysis

Methaq T. Gaata, Fadya F. Hantoosh
Department of Computer Science, College of Science, Mustansiriyah University, IRAQ
*Correspondent email: tmethaq@yahoo.com

ArticleInfo

Received
13 May 2017

Accepted
02 Jul. 2017

Abstract

Content-based search provides an important tool for users to consume the ever-growing digital media repositories. However, since communication between digital products takes place in a public network, the necessity of security for digital images becomes vital. Hence, the design of secure content-based image retrieval system is becoming an increasingly demanding task as never before. In this paper, the secure CBIR with additional improvement for the image retrieval has been presented. The proposed system consists of six phases briefly described as follows: first, feature extraction phase, which produces the low-level quantitative description of the image (color and texture) that used in calculation of similarity score and image indexing. Second, indexing for search process phase, hash table and bloom filter were employed for classification. Third, feature encryption phase, where content protection is performed using a method developed by us (including Chaotic Logistic Map). Fourth, image encryption phase, the chaos and stream cipher systems were applied as an image encryption system in order to achieve image security. Fifth, the retrieval phase, which provides a group of images replying the query based on the similarity score between images, calculated using the extracted features from each image. Finally, Relevance feedback phase, a technique that attempts to capture the user's needs through iterative feedback. Although the system proved its efficiency in search performance (with 88% of average precision), security strength, and computational complexity, it does not mean the optimal system is designed, since some weakness points still can be found that are suggested to be improved as a future work.

Keywords: Content Based Image Retrieval, Secure Content Based Image Retrieval, Image Mining, Similar Image Retrieval, Bloom Filter.

الخلاصة

توفر طرق البحث بالاعتماد على المحتوى وسيلة مهمه للمستخدمين لاستهلاك مستودعات الوسائط الرقمية المتنامية بشكل مستمر. بالاضافة الى ذلك، أن الاتصالات بين المنتجات الرقمية تتم ضمن شبكة عامة (غير امنة) لهذا اصبح من الضروري الاهتمام بأمنية الصور الرقمية. لذلك، فإن تصميم نظام آمن لاسترجاع الصور بالاعتماد على المحتوى اصبح مطلب مهم اكثر من أي وقت مضى. في هذه المقالة، تم تقديم نظام آمن لاسترجاع الصور بالاعتماد على المحتوى مع تطويراضافي. يتألف النظام المقترح من ستة مراحل والتي وصفت باختصار كما يأتي: أولا، مرحلة استخلاص الخصائص والتي تنتج وصف كمي للمستوى الواطئ من الصورة ضمن نطاق اللون والنسيج ويتم استخدام الخصائص الناتجة في حساب درجة التشابه وفي فهرسة الصورة. ثانياً، مرحلة فهرسة عملية البحث وتتم من خلال استخدام جدول التجزئة و مرشح بلوم للتصنيف. ثالثا، مرحلة تشفير الخصائص، حيث يتم تنفيذ حماية قيم الخصائص باستخدام طريقة مطورة من قبلنا (اعتمادا على النظرية اللوجستية الفوضوية). رابعاً، مرحلة تشفير الصورة من خلال تطبيق انظمة التشفير الانسيابي والفوضوي لضمان امنية الصورة. خامساً، مرحلة الاسترجاع ويتم فيها استرجاع مجموعة من الصور كاستجابة على الاستعلام اعتماداً على درجة التشابة المحسوبة بين الخصائص المستخلصة من كل صورة خلال مرحلة استخلاص الخصائص. وأخيرا، مرحلة التغذية العكسية ذات الصلة وهي تقنية التي تحاول الحصول على احتياجات المستخدم من خلال ردود الفعل المتكررة. بالرغم من أن النظام المقترح أثبت كفاءته من ناحية الاداء خلال عملية البحث (بنسبة 88٪ من متوسط الدقة)، وقوة امنية النظام، والتعقيد الحسابي، فهذا لا يعني أنه النظام الأمثل ، حيث لا يزال هناك بعض نقاط الضعف التي يمكن تحسينها خلال العمل المستقبلي.

Introduction

Content-based image retrieval (CBIR) is a way that uses visual image contents such as color, texture, shape, and spatial layout to search for images inside large image databases rather than alphanumeric-based indices. A number of

CBIR systems were proposed, for example, Blobworld, C-bird, ImageMiner, ImageRover [1]. The growing interest in CBIR technique is due to its good scalability, real time property, and needless of human labor. There are many (potential) applications using CBIR techniques. For example, art collection, surveillance system, geographical and remote sensing, medical diagnosis, military, engineering, and so on [2]. As a motivating trend of cloud computing, CBIR has concerned a lot of focus, but the consideration was mostly on enhancing the retrieval accuracy instead of handling security problems such as copyrights and user privacy. With the wide growing in computer networks usage the security attacks have been increased, the security issue becomes serious for CBIR systems [3].

In last years, some of secure image retrieval systems exist in literature. Kozak [4] presented an evaluation of the existing secure indexing schemes; the evaluation criteria were about (usability, security, and efficiency) and suggested two new methods namely EM-Index and Dynamic Secure Hash-based Index. The balance of the first proposed technique is more on the efficiency side while the other (DSH Index) shifts the balance more to the privacy side. One idea he kept insisting on and considered it as a must have for a practical similarity cloud, was to create a technique supports efficient index updates, by update he meant insertion and deletion, and that what his two new techniques had. At the end, his main goal was to provide a complex, scalable similarity cloud solutions with provable security where the balance between efficiency and level of privacy can be shifted according to requirements of the specific domain and application. Abdulsada *et al.* [5] introduced a system that works on grey images. They did not use any feature extraction method, instead the feature vector was simply a series of pixels value (and that was after reducing image resolution). Mostly the work was about the indexing scheme, where they employed Locality Sensitive Hashing (LSH); which importantly improves the system efficiency by returning the similar images in a graded order with a least distance estimation. To security objectives, the index was rotated into a confident index to avoid the detected any useful information included in the contents of the index. Bhagat *et al.* [6] developed an image retrieval system based facial data. The main idea is based on combination of two orthogonal approaches are attribute coding and attribute embedded indexing. The attribute coding utilizes high level features such as human attributes with low level features which create semantic code words in offline phase. Attribute embedded indexing reflects human attributes in binary signature of particular query image and offers effective retrieval in online phase. While for image encryption, Secured Quick Crypt (SQC) encryption algorithm was applied to protect images when they are transferred via network. Ferreira *et al.* [7] suggested a new secure system that supports the security of image, search and retrieval of images in the encrypted form, the central aspect of their system was the reduction of network traffic and client's computational overhead. The idea started by observing that in images, the color information can be extracted from texture areas, and that made them use two different techniques for feature encryption, they utilized probabilistic and deterministic encryption for texture and color features respectively. That combination according to what they said allows privacy preserving CBIR depending on color information in order to be implemented on the outsourced servers directly, while protecting the contents of images (especially their texture information) from the operators of these servers and from multiple, possibly malicious, users issuing queries.

To enhance security, the proposed system applies an encryption technique that provides the ability to encrypt and keep secure images with its contents, and later retrieve relevant images without need to decrypt all encrypted image in database. In this paper, image retrieval system over encrypted image database has been proposed. For that, the bloom filter and Hash-Table used to create our searchable index, which significantly improves the system performance by restoring the similar images in a graded order with a minimum distance calculation. Also, numerous experimental trials

to prove the efficiency of proposed system have been done.

The rest of the paper is structured as follows: the retrieval system for encrypted images is proposed in Section II. This section included: the proposed system, the image encryption technique has been developed, extract features from images, create image-indexing scheme, encrypt feature vectors, choosing a method to compare and retrieve relevant images, and finally using relevance feedback to enhance performance. The results and evaluation of the proposed system are presented in section IV it also gives examples, results and finally comparison between relevant systems. Finally, conclusions will be drawn in section IV and some possible directions for the future study are discussed.

Proposed System Design and Implementation

Secure Content Based Image Retrieval (SCBIR) technique is a challenging issue that has not been widely studied in the literature. Therefore, the focus in this work is placed on designing a new retrieval system that support image privacy preserving. A block diagram and a sequence diagram are presented in Figure 1 then each step is explained separately.

Feature Extraction

This process is consider as form of dimensionality reduction, which means that reduces the dimensions of the existing object (image) and takes essential details only. In the proposed system, three functions would be implemented:

Image resize: the size of an image must be changed from whatever it was to 96 × 96. This function is implemented to reduce the time needed for the next two steps. Size reduction is mostly related to GLCM step, although the best results can collected are by keeping the image resolution unchanged, but it will make the process time slower. However, choosing that certain size is an option depends on previous experiments done on GLCM by other researchers; results of those experiments suggest three sizes (32, 64, and 96). Obviously, 96 × 96 will have more details, and therefore better results.

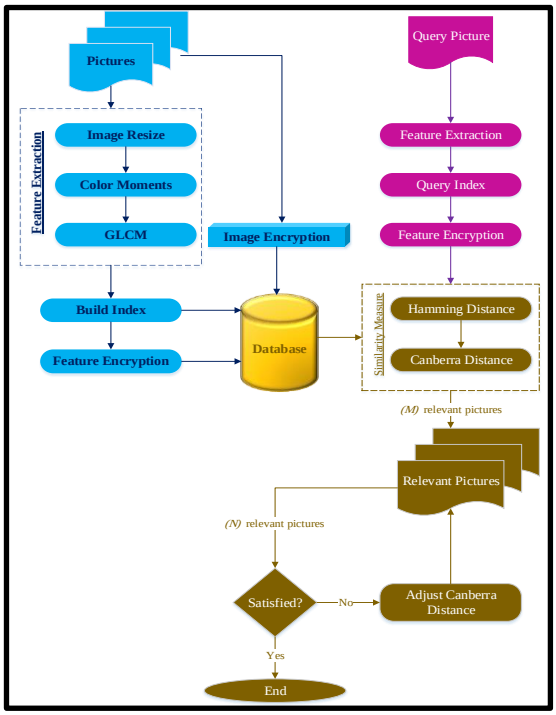


Figure 1: The block diagram for the proposed system.

Color moment: those features are used in many retrieval systems. Mainly, when the image includes independent objects, the mean, variance and skewness have been used due to its ability in descriptive color distributions in images. Mathematically, the first three moments are defined as [8]:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} f_{ij} \tag{1}$$

$$\sigma_i = \sqrt[2]{\frac{1}{N}\sum_{j=1}^{N}\left(f_{ij}-\mu_i\right)^2} \tag{2}$$

$$\theta_i = \sqrt[3]{\frac{1}{N}\sum_{j=1}^{N}\left(f_{ij}-\mu_i\right)^3} \tag{3}$$

Where μ, σ, and θ are mean value for image, standard deviation and skewness respectively, the f_{ij} is represented the value of the i-th color component of the image at pixel j, and N is represented the number of all pixels in the

image. These color moments are calculated using HSV color space.

GLCM: is a statistical tool used to compute the second order features of image (statistical texture). It can be described as a square matrix with size equal to the number of grey scale levels that the whole image has been reduced. Grey/ Color Level Co-occurrence Matrix have its own steps:

Step One: Convert the colored image into grey, which is done by summing the three channels (RGB) of each pixel and dividing it by three.

Step Two: Grey/Color level reduction Greyscale images consist of 8-bits per pixel, which means 256 different intensities (shades of grey); these intensities must be changed into much lower levels, like 4, 8, 16, or more. This is important for time reduction and similarity, the more the shades are the less similarity and more slowly will be. Choosing the number of bits for the grey level is one of the three main elements of GLCM (pixel distance, and angle) [9]. Grey level with 16 shades is the chosen value in the proposed system; this value was proven to give better results than those before it or after it. Level reduction process is clarified in algorithm 1. While Color Level Co-occurrence Matrix (CLCM) is an exact copy of GLCM except for the color part, where the three channels RGB used instead of grey and the rest is the same (same level, same distance, and almost same directions). As for distance we chose it to be one, and the direction was chosen after many tests to be 135° for Red channel, 0° for Green, 135° for blue, and 0° for grey.

Step Three: Build the GLCM and feature extraction. In this step, *Equation* (4) is utilized to build the matrix. Algorithm 2 shows how to generate GLCM. Haralick features are extracted from the previously developed matrix, using equations from [10].

$$CM(i,j) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\begin{cases}1 \text{ if } p(x,y) = i \text{ and } I(x+\Delta x, y+\Delta y) = j \\ 0 \text{ otherwise.}\end{cases} \quad (4)$$

Where CM is the Co-occurrence Matrix, *G* is the number of grey levels in the image, *P(x,y)* is the image pixel value at location (*x, y)*, finally the offset (Δ*x*, Δ*y*), specifies the distance from the pixel-of-interest to its neighbor and also used for angle specification.

Algorithm 1: Grey level Reduction

```
INPUT: image, G
OUTPUT: image with G grey levels
1: i ← 0
2: For  x ← 0 to 256
3:    Arrayi ← x
4:    Increase x by (256 / G)  //G is the chosen No. intensities (16 in our case).
5:    i ← i + 1
6: End
7: Arrayi ← 256
8: For  i ← 0 to width of image
9:       For j ← 0 to height of image
10:            Get pixel_value
11:            For each x ϵ Array
12:                  IF (pixel_value  >=  Arrayx AND pixel_value  <  Arrayx + 1)
13:                  THEN  pixel_value ← Arrayx
14:                  End if
15:            End for x
16:       End for j
17: End for i
18: Return image
```

Algorithm 2: Build Grey Level Co-Occurrence Matrix

```
INPUT: Image, G, row, col, width, height
OUTPUT: GLCM matrix
1:        For x ← 0 to G
2:             For y ← 0 to G
3:                  Count ← 0     // Calculate the number of occurrences of each pixel
4:                  For  i ← row to width
5:                       For j ← col to  height
6:                          Get current_Pixel_value
7:                       IF  current_Pixel_value  =  x And next_Pixel_value = y
8:                       THEN increase Count by 1
9:                       End for j
10:                 End for i
11:                 Matrix [x,y] ← Count;
12:            End for y
13:       End for x
14:       Return Matrix
```

Where the G is represented the number of grey levels, row/col is the starting point of the image top/left (the value might be zero or one), width/ height is the image width and height. The last four parameters depends on the direction/ angle we choose to calculate GLCM, and finally, next_Pixel means the pixel next/ above/ right diagonal or left diagonal to the current pixel, note that next_pixel value also depends on the angle value.

Build the Searchable Index

After generating image feature vectors, the system execute the indexing algorithm to build a searchable index. A bloom Filter is a probabilistic tool used to describe huge data and assign membership queries. A simple example is clarified in Figure 2 to explain how it works, where data 1 and data 2 are of any type of information (words or numbers), h1, h2, and h3 are the hush functions used to generate the hash code, and finally, the array in the middle is the output of the whole bloom filter [11].

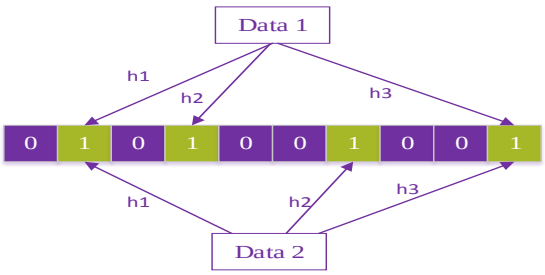


Figure 2: Bloom Filter.

Feature Encryption

Presuming the server is honest but curious, the extracted features must be encrypted for more privacy. Features encryption is a critical property in the proposed system. However, to make sure that our features are not going to be damaged (by "*damaged*": it completely changes the values, which will take us to wrong places when using similarity) a simple encryption (or modification) method is utilized (Algorithm 3) using the logistic map, logical and mathematical operations on the key before adding its effect on the feature vector.

Algorithm 3: Feature Encryption

INPUT: Feature vector, Key
OUTPUT: Encrypted feature vector
1: Encode Key to its ASCII form.
2: $X \leftarrow 0$
3: K ← convert key bytes to integer
4: For i ← 0 to K length
5: $X = (K_i \text{ XOR } K_{(K \text{ length } -1 - i)}) / X + 1$
6: End for i
7: Y ← Convert ASCII Key to Hexadecimal
8: $X_{01} \leftarrow (Y_0 * 2^0 + \ldots + Y_{15} * 2^{15}) / 2^{16}$
9: $X_{02} \leftarrow (Y_{16} * 2^0 + \ldots + Y_{31} * 2^{15}) / 2^{16}$
10: $X_0 \leftarrow (X_{01} + X_{02}) \bmod 1$
11: $X_1 \leftarrow 4 * X_0 * (1 - X_0)$ //CLM *Equation*
12: For i ← 0 to Feature vector Length
13: Encrypted feature vector$_i$ ← Feature vector$_i$ / $(X * X_1 + 0.1)$
14: End for i
15: Return Encrypted feature vector

Image Encryption

A step that guarantees picture confidentiality, a particular algorithm was utilized [12] to encrypt the images that won't take too much time and yet encrypts well.

Similarity Measure

A similarity measure is used to expresses the relationship between two n-dimensional feature vectors. When a query process is performed the similarity index between the query image and each image in the database is computed. The images that have less distance compared to the query image are considered the best results. The proposed system consists of two types of numeric data (binary generated by bloom filter and real numbers from Haralick features). Therefore, two kinds of distance measures are utilized [13]:

1. Hamming Distance: The Hamming distance computed between two vectors to determine the correlation degree between them. Obviously, the value of Hamming distance can calculated according to *Equation* (5), and if it is zero, then the vectors is identical [13].

$$D_{Ham}(x,y) = \sum_{i=0}^{n} |x_i - y_i| \quad (5)$$

Where *n* is the length of string *x*, and string *y*.

2. Canberra Distance: The Canberra distance is derived from the Manhattan distance. It is calculated as [13]:

$$D_{Can}(x,y) = \sum_{i=0}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (6)$$

Where *n* is the number of features, *x* and *y* are the feature vectors of an image and a query image respectively.

Relevance Feedback

Relevance feedback is a supervised learning method used to increase the efficiency of image retrieval systems. The core idea is based on used positive and negative examples in order to enhance the performance of proposed. In this step, the proposed system will refine the obtained results using the feedback and current a new list of images.

Experimental Results

To test system performance, the Corel's databases have been used. The dataset consists of 1000 images grouped in 10 classes; each class contains 100 color images [14]. These classes are categorized as Africa, Beach, Buildings, Buses, Dinosaurs, Flowers, Elephants, Horses, Mountains, and Food. Figure 3 shows sample images from the dataset.

Our experiment has been performed on 2.00 GHz Pentium Processor, Core i7, Windows 7 Operating System, with a RAM of 8 GB, using C sharp from Visual Studio 2012 and SQL Server to implement this system.

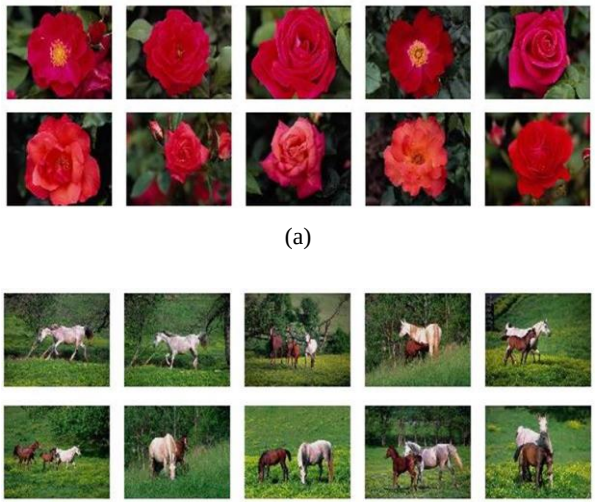
Figure 3: Sample images from Corel dataset

In order to evaluate the proposed system, the precision graphs have been used. Precision is the likelihood that a retrieved image is relevant, defined as follows:

$$\text{Precision} = \frac{\text{No. relevant images retrieved}}{\text{No. images retrieved}} \quad (7)$$

The some examples from the proposed system show in Figure 4 (the top left picture is the query image and the most relevant result, while the bottom right picture is the least relevant one).

As for the consumed time in uploading and retrieving image processes, it has been calculated using a built-in function on C sharp that gives approximate time elapsed for each process performed, as shown in Table 1.



(a)

(b)

Figure 4: Examples from testing the Proposed System, (a) case 1, b) case 2.

Table 1: Time Measurement for the Proposed System.

Upload 1000 Pictures		**Retrieve 10 Pictures**	
Without Encryption ~2.5 Minutes	With Encryption ~5 Minutes	Without Decryption ~0.5 Seconds	With Decryption ~2 Seconds

Finally, the proposed system has been compared with other relevant systems that use the same dataset; it has been found that our system takes the second place of highest accuracy as shown in Table 2.

Table 2: Comparison between the Proposed System and Relevant Systems.

Ref. No.	Utilized Features	Utilized Index	Accuracy
2009 [15]	Bag of Visual Words (BoVW) generated from Color Histogram	Inverted Index Min Hash	~ 0.75 ~ 0.75
2011 [3]	Color Structure Descriptor (CSD)	_	~ 0.65
	Color Layout Descriptor (CLD)		~ 0.59
	Edge Histogram Descriptor (EHD)		~ 0.39
	Homogeneous Texture Descriptor (HTD)		~ 0.58
2013 [16]	Color Histogram	Secure Searchable Index	0.90
2015 [6]	BoVW	Inverted Index (Posting List)	~ 0.69
Proposed System	Color Moment, GLCM	Bloom Filter	0.88

Conclusions

In this paper, the image retrieval system for encrypted images has been introduced. Grey Level Co-Occurrence Matrix with Haralick features, combined with color moments are employed to create the feature vector. We utilized Bloom filter and hash-table for image dataset classification, this scheme is known for its efficiency and speed. Feature vectors are encrypted, but the indexing table is not, because it does not have anything to be revealed, since its contents are binaries created by our hash function. The proposed system does not support query image encryption (while many researchers do encrypt it, or watermark it) because the user does not need to upload it or expose it to the server. The server does not even need the query image to complete the search since the similarity measure is between the features of the images, which are already encrypted. Our system supports relevance feedback, which makes the results even better.

For future works suggestions, we shall make our system support invariant property, since our system does not support scale or rotation invariant, adding this feature would reduce one of the weakness points. Utilize different indexing scheme, we could suggest using kernelized locality sensitive hashing, or convolutional neural networks. Finally yet importantly, we would like to make the system simpler and less user intervention. This is done by letting all the work to be occupied on the server side, which means we need first to find an image encryption that changes only parts from image, to make the features extraction process possible in encrypted images, once we achieve this the rest is easy.

References

[1] R. C. Veltkamp, H. Burkhardt, H Kriegel, “State-of-the-Art in Content-Based Image and Video Retrieval,,” First Edition, Springer Science and Business Media Dordrecht, ISBN 978-90-481-5863-8, 2001.

[2] F. Wang, “A Survey of Content Based Image Retrieval Techniques,,” 2014.

[3] J. Zhang, Y. Xiang, W. Zhou, L. Ye, Y. Mu, “Secure Image Retrieval Based on Visual Content and Watermarking Protocol,,” The Computer Journal, 2011.

[4] S. Kozak, “Efficiency and Security in Similarity Cloud Services,,” Journal Proceedings of the VLDB Endowment, Vol. 6 Issue 12, Pages 1450-1455, August 2013.

[5] I. Abdulsada, A. N. M. Ali, Z. A. Abduljabbar, H. Sh. Hashim, “Secure Image Retrieval over Untrusted Cloud

Servers,," International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-3, Issue-1, October 2013.

[6] M. N. Bhagat, B. B. Gite, "Image Retrieval using Sparse Codewords with Cryptography for Enhanced Security,," IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 16, PP. 22-26, 2014.

[7] B. Ferreira, J. Rodrigues, J. Leitao, H. Domingos, "Privacy-Preserving Content-Based Image Retrieval in the Cloud,," arXiv:1411.4862, 2015.

[8] S. Mangijao Singh, K. Hemachandran, "Content-Based Image Retrieval using Color Moment and Gabor Texture Feature,," International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, September 2012.

[9] D. Gadkari, "Image Quality Analysis Using GLCM,," Master thesis, University of Central Florida, Orlando, Florida, 2004.

[10] R. M. Haralick, J. Shanmugam, and I. Dinstein, "Texture feature for image classification,," IEEE Transactions on Systems, Man, and Cybernetics, 1973.

[11] G. Manjula, P. Brindha, "A Survey on Architectural Design of Bloom Filter for Signature Detection," International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 3, 2013.

[12] M. T. Gataa, F. F. Hantoosh, "An Efficient Image Encryption Technique using Chaotic Logistic Map and RC4 Stream Cipher,," International Journal of Modern Trends in Engineering and Research (IJMTER), Volume 03, Issue 09, September– 2016.

[13] P. Cichosz, "Data Mining Algorithms: Explained Using R", First Edition, John Wiley and Sons, pp 316-318, 2015.

[14] http://wang.ist.psu.edu/docs/related/

[15] W. Lu, A. Swaminathan, A. L. Varna, M. Wu, "Secure Image Retrieval Through Feature Protection,," IEEE international conference on acoustics, speech, and signal processing, Pages 1533 – 1536, 2009.

[16] Z. Xia, Y. Zhu, X. Sun, J. Wang, "A Similarity Search Scheme over Encrypted Cloud Images based on Secure Transformation,," International Journal of Future Generation Communication and Networking (IJFGCN), Vol.6, pp.71-80, 2013.