International Journal of Engineering & Technology, 7 (4.39) (2018) 1042-1046

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Cloud Space Optimization by Preventing Similar Image Insertion Using Hashing Based on Feature Point

[1]Radhika Nambiar, [2]Prithvi A, [3]Priyanka A, [4]Suma M Shirahatti, [5]Prof. Radhakrishna Dodmane

[1,2,3,4,5] *Department of Computer Science and Engineering, NMAM Institute of Technology, Nitte-574110, Karkala, Udupi,India*
Corresponding author E mail: radhakrishna@nitte.edu.in

Abstract

People tend to click a lot of pictures in their phones or they download images from the internet. When these images are uploaded to cloud, there is high possibility of similar images being already present which were uploaded earlier. This leads to the wastage of space and related issues. For efficient utilization of cloud space, the proposed system provides a framework to determine duplicate images in the cloud. The duplicate images are determined using hash value as a key parameter. The hash values of the image are generated using perceptual image hashing algorithms. Since these algorithms are geometric transformation variant, the proposed system is developed based on feature extraction and image hashing. This will ensure more accuracy. The system is compared with existing algorithm based on accuracy and the results are shown.

Keywords: *A-Hash, D-hash, P-hash, Feature extraction, DCT, MSER, Correlation coefficient factor etc.*

1. Introduction

In modern era, it has become a practice to store everything on the cloud. Due to this, there may be a possibility of duplicate information occupying the cloud. This leads to the wastage of space and related issues. When a user stores textual data, the space occupied on the cloud will not be alarming. But when the data to be stored are image or video, it definitely becomes a burden. This also reduces the performance of the cloud due to the increase in the time to search for a given image/video. So we need to have an efficient system which can detect similar images and prevent them from being uploaded. Images in cloud occupy a lot of memory because of which storage space problems may occur frequently. It is possible to get extra space by paying an amount to the cloud service provider monthly or yearly. However, determining similar images and keeping only one copy of those images helps in freeing up some space. There are a lot of research works that have been carried out in the direction of detecting images for similarity. These are based on whether the corresponding pixels or features of two images are equal. When there is large number of images already present in the cloud, it is quite difficult to compare a recently uploaded image with every other image using the above method. This method also increases the time needed to process due to its linearity. The best practice is to generate a [4] hash value for each image which is like a fingerprint for that image. This provides facility to search duplicate or near duplicate images in constant time. These hashes can be generated using cryptographic hashing algorithm or perceptual
hashing algorithm. Cryptographic hashing [6] checks image integrity for security purpose. But the hash of the image generated using this method is sensitive to its content [6]. In perceptual image hashing [2, 6] the hash value of similar images are slightly different. Hence perceptual image hashes are preferred to calculate the similarity of images. Another method which can be adopted in identifying duplicate images is based on feature extraction. Feature extraction [7] is a type of dimensionality reduction, that efficiently represent interesting (pixel of interest) part of an image as a compact feature vector. An interest point detector [8] or feature point detector is an algorithm that chooses feature points from an image based on chosen criterion. Feature point detectors [2] are sensitive to content changing manipulation. A feature descriptor is a vector of values which describes the image patch around an interest point. Together, the interest point and its descriptor are called the local feature point of that image. In this paper, the proposed system provides a framework based on combination of feature extraction and hashing. The algorithm is compared with the existing algorithms based on accuracy. The rest of the paper is organized as follows. The pros and cons of the existing techniques are specified in the literature review. The next section elaborates on the design of the existing systems followed by the proposed system design. The result and comparison analysis are included in the later sections, followed by the conclusion and future enhancement of the proposed system. Final section includes the references, which motivated to do the proposed work.

2. Literature review

2.1 Interest point detectors:

The Maximally Stable Extremal Region (MSER) is an affine-invariant alternative to the SIFT. This detector extracts stable regions from the image by considering the change in area with respect to the change in intensity of a connected component defined by thresholding the image at a given gray level.

2.2 Perceptual image hash

The hash value of an image should remain invariant even if it undergoes lossy (content preserving) transformation. [3] A-Hash uses the average pixel strength of the image to calculate a hash value. D-Hash algorithm uses the difference between adjacent pixels to generate a hash value. P-Hash algorithm compares the mean value with DCT coefficient matrix to generate hash value. In [4] P-Hash was found to be more efficient than other hashing functions. But it is sensitive to image rotation as specified in [5]. Perceptual image-hash been used in areas like plagiarism detection of similar images [4]. The proposed system [4] compares the various rotated versions of an image to images present in the database to identify plagiarized images.
To minimize the flaws which have been spotted in the literature, the proposed system adopts the combination of feature extraction using SURF and image hashing. So this makes it scale invariant and rotation invariant and also helps in identifying similar images efficiently.

3. Methodology

A hash value can be generated using two methods. They are Cryptographic Hashing and Perceptual Hashing. Cryptographic hash functions have the following properties:
•It will produce same hash value for identical images [9, 11, 12].
•For any given message, cryptographic hashing will compute the hash value quickly [10].
The disadvantage of this method is small change in the image will result in entirely different hash value [9, 11, 12]. The new hash value is very different from the old hash value. This is called the avalanche effect. This is not an efficient method to determine similar images because of this effect. That results in difficulty in identifying duplicate images.
Unlike the cryptographic hash function, the perceptual hash generates the same hash value for similar images. Slight variation in images will result in same hash value [9]. This can be used to check the similarity between two images. There are various algorithms to generate a hash value for an image, which can then be used to check for similar images from a set of images:

3.1 Average Hashing (A-HASH).

Average hash is the simplest perceptual hash algorithm. Where hash value will be generated by taking the average of all the low frequency of an image. In an image, image details are represented by higher frequencies component, while the image structure is represented by lower frequencies component [15]. In this algorithm, image details are eliminated by removing the high frequency. And low frequency of the image is taken into consideration which represents the structure of the image [13].

Steps to generate Average hash as described in [6].

1) Resize image dimension. (Reduce it to 8 × 8 dimension which contain 64 pixels. Remove high frequency of image.)
2) Reduce color. (Reduce image with 64 pixels (64 red, 64 green, 64 blue) to 64 pixels where pixel value ranges from 0-255.)
3) Average the color. (Take the average of 64 pixels.)
4) Generate the hash as shown in (1). (If the pixel is brighter than the mean value considers it as 1 else 0).

This algorithm is simple, fast and which have low computing cost [6]. In Fig .1

$$h = \begin{cases} 1. A(i,j) \geq avg \\ 0, A(i,j) < avg \end{cases}$$

Hash value is 7FE3 8181 81C3 C3FF

Hash value is 7FE3 C181 8181 C3FF

Fig. 1. A-hash values for two images

Hamming difference of these 2 images is 3 even though they are different. Even though this is simple, fast and cost effective method [9], it produces positive result for false condition. So it is not efficient algorithm.

3.2 Difference Hash (D-hash)

This algorithm gives better accuracy then A-hash algorithm. This algorithm tracks the image gradient. Steps of D-Hash algorithm are similar to A-hash algorithm. In both algorithms first step is to remove image details [14] (high frequency part of the image). Then as name suggests compute the difference between adjacent pixels.

Steps involved in generating D-hash.
Resize image dimension. (Reduce it to 9 × 8 dimension which contain 72 pixels. Remove high frequency of image.)
Reduce color. (Reduce image with 72 pixels (72 red, 72 green, 72 blue) to 72 pixels where pixel value ranges form 0-255.)
Compute the difference. (Here compute the difference between the adjacent pixels. That is M (i, j) = A(i, j+ 1) − A(i , j).(This result 8×8 with 64 bits value. In this step it tracks the relative gradient direction.)
Generate the hash value as show in (2). (Here assign the bit 1, if (i, j) > 0 , else assign 0. If previous pixel is brighter than the current one the value of the bit is 1 or otherwise it is 0.

$$h = \begin{cases} 1, M(i,j) \geq 0 \\ 0, M(i,j) < 0 \end{cases} \quad (2)$$

3.3 Perceptual Hashing (PHASH)

Perceptual hashing high frequency of an image will be eliminated by performing discrete cosine transform (DCT). This algorithm works on frequency domain. This is extended version of average hashing.
Steps to generate P-Hash [16]
Resize the image dimension. (Reduce the size to 32 × 32 dimension. The objective here is to speed up the DCT computation rather than removing high frequency of an image.)
Reduce color. (reduce to grayscale).
Compute the DCT. (here converting into elementary frequency component. this consist of low, medium, high frequency [10])
Consider upper-left 8 × 8 sub matrix of the DCT coefficients. (Consider upper left corner of the DCT
coefficient. The top left 8 × 8 DCT coefficients generally have larger values, which implies they carry more information.)
Compute the average DCT value.
Generate the Hash value as shown in (3) (If the pixel is brighter than the mean value of DCT coefficient consider it as 1 else 0).

$$h = \begin{cases} 0, D(i,j) \leq davg \\ 1, D(i,j) > davg \end{cases} \quad (3)$$

A and B are input and output image respectively, and the two dimensional DCT for A and B can be defined as shown in (4) [19],

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{m-1} \sum_{n=0}^{n-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N}, \quad 0 \le p \le M-1, \; 0 \le q \le N-1$$

Where,

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, p = 0 \\ \sqrt{\frac{2}{M}}, 1 \le q \le N-1 \end{cases}$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, q = 0 \\ \sqrt{\frac{2}{N}}, 1 \le q \le N-1 \end{cases} \quad (4)$$

3.4 Feature Extraction Hashing

In perceptual hashing result may vary for large geometric transform [4]. To overcome this drawback, feature extraction based hashing can be used which is invariant to geometrical transform [17, 18].

So this method can be used for accurate image similarity detection. The above perceptual hashing algorithm are faster than this algorithm, this is because extraction of feature point is time consuming step. And the execution time also depend on which Feature detector is used (example SURF, SIFT, MSER etc.). The complexity of this system is depending upon which feature detector tool is used. Here experiment has been done using MSER feature detector tool.

Steps to generate feature extraction based hashing

1) Convert the image into grayscale. (Reduce image with 64 pixels (64 red, 64 green, 64 blue) to 64 pixels where pixel value ranges form 0-255.)
2) Extract the feature points of the image. (Using SURF feature point object can be extracted. And MSER will extract Maximal Stable Extremal Regions as an object.)
3) Extract the interest point descriptor. (Returns feature vector for the extracted feature point. This is a M×N matrix of M feature vectors or descriptor and N is the length of each descriptor.)
4) Multiply transpose of descriptor matrix with the descriptor matrix. ($P = M^T \times M$ with dimension of 64×64.)
5) Consider median of each row in the resultant matrix. (Consider the resultant matrix as $R_{1\times64}$).
6) Consider mean of each row in the resultant matrix. (Consider the resultant matrix as $R_{1\times64}$).
7) The mean and median vectors are then used for comparison with those of the other images already uploaded to the cloud.

4. Experiment

The proposed system has been designed as shown in fig 2.

1) The user selects the images that he or she wants to upload through the user interface which has been designed in matlab.
2) The images are then processed by the feature extraction algorithm designed in matlab.
3) The algorithm generates hash value for the images.
4) Each of the hash value is compared to other hash values present in the database.
5) The comparison of hash values in existing algorithms like A-hash, D-hash and P-hash is by finding the hamming distance. In the proposed system the hash values will be compared using correlation coefficient.
6) The correlation coefficient of the mean vector of the image to be uploaded and the mean vector of an image which is already in the database is obtained.
7) Similarly, the correlation coefficients for the median vector of the two images are also obtained.
8) The two correlation coefficients are used to determine whether the two images are similar or not. If the values are closer to one, then the images are similar else they are different.
9) Repeat step 6 to 8 until the image to be uploaded is similar to any of the images in the cloud.
10) If the image is different from those already in the cloud, the mean and median vectors are inserted into the database and image is uploaded to the cloud. Otherwise it is not uploaded.

5. Results

The proposed system has been tested for various images with twelve different variations (varying intensities, contrast, rotations etc.) of each of these images. The result for each of the cases has been shown in the Fig .3.

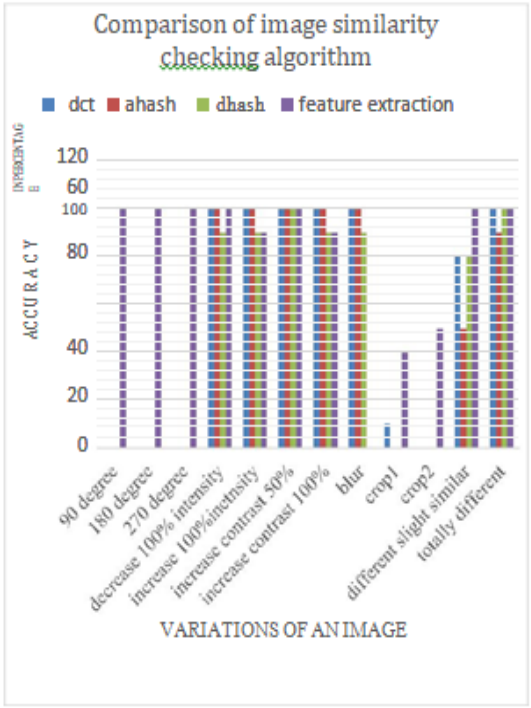

Fig.3. Comparison of image similarity checking algorithm

The results show, the proposed system provides 100% accuracy for rotation (90,180 and 270-degree rotation of images). In the existing algorithms like A-hash, D-hash, P-hash, the pixel matrix will change when an image is rotated. Hence the hash value generated will be very different. Since the proposed system is based on the features of an image, the values generated will be same irrespective of angle of rotation of the image.

The proposed system provides nearly 95% of accuracy for intensity and contrast variations. When the intensity and contrast of an image is slightly increased, the Maximally Stable Extremal Regain(MSER) will not change. When there is large variation in the intensity and contrast value of an image the MSER region may change at times. So the proposed system provides 90% accuracy when the intensity

and contrast is increased by 100%. It also gives 100% accuracy in identifying images which are different from test image.

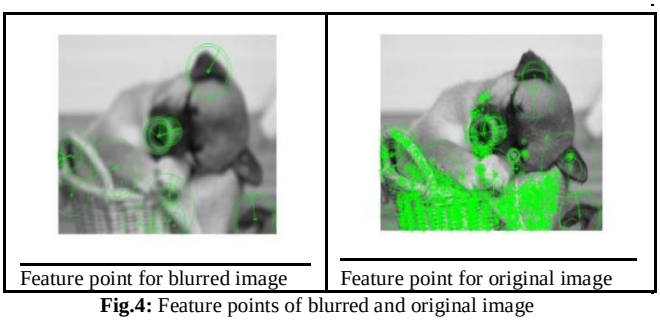

Fig.4: Feature points of blurred and original image

However, it does not identify the blurred versions of the same image. This is because the features cannot be easily detected when the image is blurred as shown in fig .4

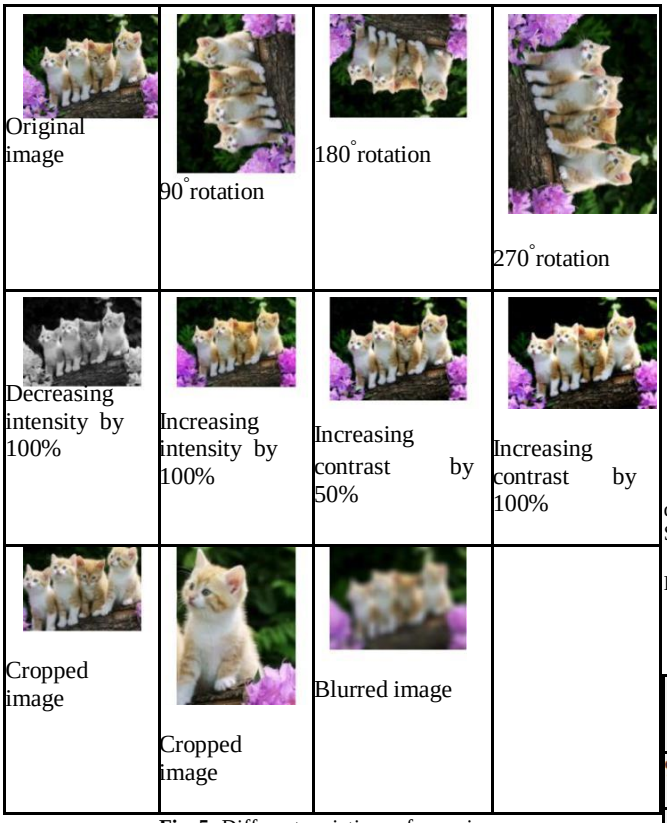

Fig. 5: Different variations of same image

It can also be observed that the accuracy in identifying the cropped version is low but it is better when compared to other perceptual hash algorithms. The outcome of the proposed system depends on which region of the image has been cropped. In fig 5. there are two cropped images. In the first cropped image almost all the feature points are detected, hence it is detected as "similar" images. In the second image there are many features which are not present; hence the outcome is "different".

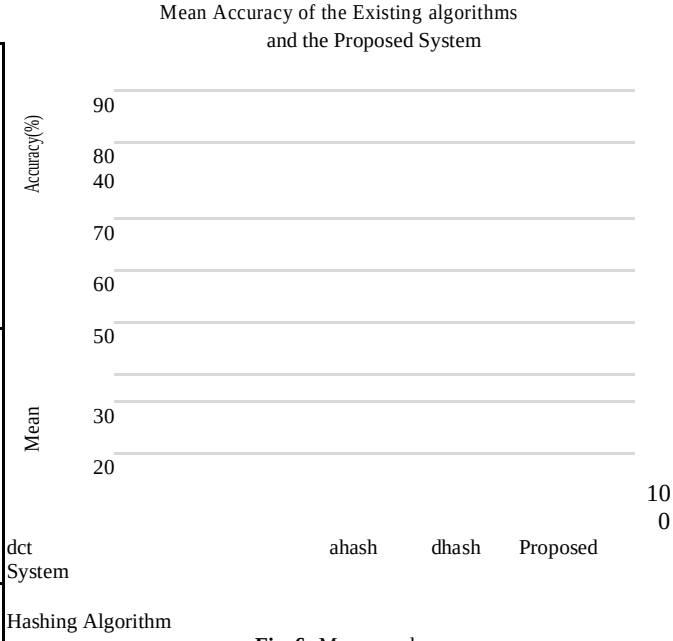

Fig. 6: Mean graph

Table (1)

	A-Hash	**D-Hash**	**P-Hash**	**Proposed system**
90° rotation	Different	Different	Different	Same
180° rotation	Different	Different	Different	Same
270° rotation	Different	Different	Different	Same
Decreasing intensity by 100%	Same	Same	Same	Same
Increasing intensity by 100%	Same	Same	Same	Same

Increasing contrast by 50%	Same	Same	Same	Same
Increasing contrast by 100%	Same	Same	Same	Same
Cropped image 1	Different	Different	Different	Same
Cropped image 2	Different	Different	Different	Different
Blurred image	Same	Same	Same	Different
Different image	Different	Different	Different	Different

Fig 6 shows the overall (mean) accuracy of the four algorithms including the proposed system algorithm. The overall accuracy of each algorithm has been calculated by taking mean of the accuracies obtained for each of the variations of the images.
It can be seen that P-Hash has more accuracy than A-Hash and D-Hash as stated earlier. The proposed system has highest overall accuracy of 79.16%.
The table (1) shows result of four algorithms for fig .5. image set. This shows that proposed system gives better result than A-Hash, D-Hash and P-Hash algorithms. But it fails to identify the blurred versions of the same image.

6. Conclusions

There is no global algorithm for image hashing. Depend upon the application and requirement, appropriate algorithm should be considered. In this proposed system accuracy or robustness is considered as main requirement then the execution time.
Execution of D-hash, P-hash and A-hash is much faster compared to proposed system. But these algorithms generate different hash value for geometrical transform like rotation and cropping. To overcome this, hash value can be generated by considering the feature points as explained in the paper. The proposed system is invariant to geometrical transformation as well as contrast and intensity variations. So this hashing technique will help in determining similar images accurately for keeping only one copy of those images in the cloud to reduce the wastage of space in cloud.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding 110,pp 346–359, 2008
[2] Vishal Monga, Arindam Banerjee, and Brian L. Evans, "A Clustering Based Approach to Perceptual Image Hashing", IEEE Trans. on Information Forensics and Security, pp 68-97, 21 Feb 2006
[3] MengjuanFei, ZhaojieJu, Xiantong Zhen, Jing Li, "Real-time Visual Tracking Based on Improved Perceptual Hashing", vol 76, Issue 3, pp 4617-4634, Feb 2017,
[4] Vipul Bajaj, Sanket Keluskar, Ravi Jaisawala and Prof .Rupali Sawantb,"Plagiarism Detection of Images", International Journal of Innovative and Emerging Research in Engineering, vol 2, Issue 2, pp 140-144, 2015
[5] Google books [Online].Available: https://books.google.co.in/books?id=vchMDwAAQBAJ&pg=PA1291&lpg=PA1291&dq=hash+algorithm+not+suitable+for+rotated+images&source=bl&ots=ZBmKUR8ue4&sig=aRtMnGo3YstHGvp5Q8IXGrrVLAA&hl=en&sa=X&ved=0ahUKEwi01uudxt_ZAhVKuo8KHf3ZBv4Q6AEIbzAM#v=onepage&q=hash%20algorithm%20not%20suitable%20f or%20rotated%20images&f=false
[6] Sam Farisa Chaerul Haviana and Dedy Kurniadi, "Average Hashing for Perceptual Image Similarity in Mobile Phone Application", Journal of Telematics and Informatics (JTI),Vol.4, No.1, pp. 12-18, March 2016.
[7] MatLab website [Online].Available: https://www.mathworks.com/discovery/feature-extraction.html
[8] Google books [Online].Available: https://books.google.co.in/books?id=6ElXDQAAQBAJ&printsec=front cover#v=onepage&q=Interest%20point%20detector&f=false
[9] Longjiang Yu and Shenghe Sun, "Image Robust Hashing based on DCT Sign", IEEE, 26 December 2006,
[10] Wikipedia [Online].Available:https://en.wikipedia.org/wiki/Cryptographic_hash_fu nction
[11] Harald Baier, Frank Breitinger , "Security Aspects of Piecewise Hashing in Computer Forensics", IEEE, 27 June 2011
[12] Ashwin Swaminathan, "Robust and Secure Image Hashing,", IEEE, 17 September 2007
[13] Francisco Vega, Jose Medina , Daniel Mendoza, Vıctor Saquicela, and Mauricio Espinoza, "A Robust Video Identification Framework using Perceptual Image Hashing", IEEE, 21 December 2017
[14] HackerFactor Website [Online].Available:http://www.hackerfactor.com/blog/index.php?/archiv es/529-Kind-of-Like-That.html
[15] Andrea Drmic, Marin Silic, Goran Delac, Klemo Vladimir, Adrian S.
[16] Kurdija, "Evaluating Robustness of Perceptual Image Hashing Algorithms", MIPRO 2017, May 22- 26, 2017.
[17] [Online].Available:https://www.d2dcrc.com.au/blog/perceptual-hashing-of-images/
[18] Xudong Lv and Z. Jane Wang, "Shape Context based Image Hashing using local feature points", IEEE, 2011.
[19] Xudong Lv, Robust, "Digital Image Hashing Algorithms for Image Identification", University of British Columbia, 2013.
[20] MatLab website [Online].Available: https://in.mathworks.com/help/images/ref/dct2.html