ROBUST IMAGE HASHING FOR TAMPER DETECTION USING NON-NEGATIVE MATRIX FACTORIZATION

Mrs. Bhate Gauri*

Abstract

The proliferation of digital images creates problems for managing large image databases, indexing individual images, and protecting intellectual property. Further complication arises from the fact that two images appearing identical to the human eye may have distinct digital representations, making it difficult to compare a pair of images. An image may undergo various digital manipulations, e.g., de-noising, contrast enhancement, geometric transformation, and JPEG compression. Since these normal operations do not change the main contents in the image, they should not significantly change the hash value. On the other hand, image may be tampered by malicious attackers. Unacceptable changes should produce a completely different hash. The non-negative matrix factorization is used to construct the image hash in this work. First image is scaled to fix size, low pass filtering is performs for smoothening the image. Secondary image is obtained by rearranging the pixel. Coefficient matrix obtained after performing is used to generate the hash value. Similarity betweens hash values measured by hamming distance.

***Keywords*— Image hashing, NMF, Image Authentication,**

* (M.Tech-Electronics)Lecturer, D.Y.Patil College of Engineering, Akurdi, Pune, India

I. INTRODUCTION

The proliferation of digital images creates problems for managing large image databases, indexing individual images, and protecting intellectual property. Images are being transferred over the Internet and are readily available for access from any part of the world and without introducing authentication mechanism. One cannot determine if an image already exists in a database without exhaustively searching through all the entries. Further complication arises from the fact that two images appearing identical to the human eye may have distinct digital representations, making it difficult to compare a pair of images. This has spurred interest in developing algorithms to generate suitable image identifiers, or image authentication system or *image hash* functions. Image indexing technique may be called an image hash function.

One possible option to derive content-dependent short binary strings from the image is the use of conventional cryptographic hashes such as message digest 5 (MD5) and secure hash algorithm (SHA-1) [2]. The traditional cryptographic hash functions such as MD5 and SHA- 1 maps

input data to a short string with a fixed size. But they are unsuitable for images. For cryptographic hashes, any small changes in the input, even a single bit, will significantly change the hash value. An image may undergo various digital manipulations, e.g., de-noising, contrast enhancement, geometric transformation, and JPEG compression. Since these normal operations do not change the main contents in the image, they should not significantly change the hash value. For this reason, such an image hash function is termed as a *perceptual image hash.* On the other hand, image may be tampered by malicious attackers. Unacceptable changes should produce a completely different hash

Image hashing derives content based compact representation of image called image hash.

Hash function (h), applied on message (m) of arbitrary length, gives fixed length o/p value h (*m*).

In general, an ideal image hash should have the following properties:

General Properties of Hash Function:

- H is applicable to a block of data of any size.
- H produces fixed length of output.
- H should be one way function so that their signature does not disclose messages.

- It should be computationally infeasible given a message and its hash value to compute another message with same hash value.

II. PROPOSED WORK:

The proposed work on image hashing scheme uses the property of non-negative matrix factorization. The image is first converted into a normalized monochrome pixel array. By re-arranging its entries, a secondary image is obtained. NMF is applied to produce a feature-bearing coefficient matrix, which is then coarsely quantized to achieve high-rate compression. The obtained binary string is scrambled to generate the image hash.

The proposed image hashing scheme is composed of the following five steps:

1) Image pre-processing– Pre-processing includes image resizing, low pass filtering.
2) Construction of a secondary image subject to NMF
3) Data reduction with NMF to obtain a low-rank approximation of the secondary image, and coarse quantization of the obtained coefficient matrix.
4) To produce hash string for given images.
5) Check the similarity- use of hamming distance to measures similarity between hash values.

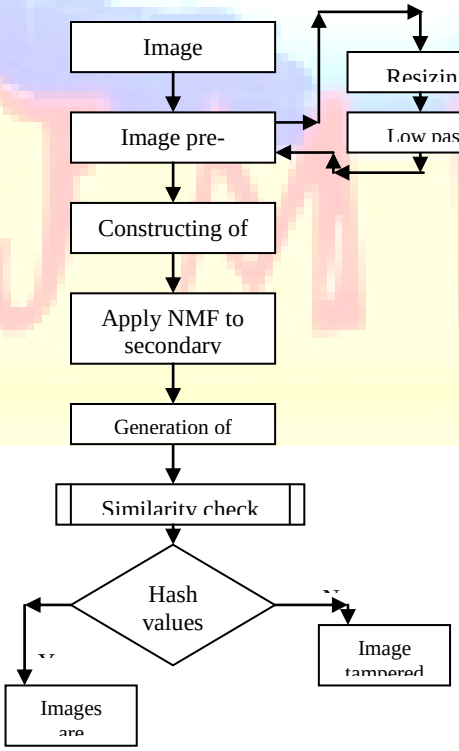


Fig. 1 Flowchart of proposed Work.

Image pre-processing, or normalization:

The field of Digital Image Processing refers to processing digital image by means of digital computer. The original image undergoes a sequence of pre-processing, or normalization, including image re-sizing, color space conversion, and low-pass filtering.

- *Image Resizing*:

Image re-sizing changes the image into a standard size Q X Q. This is done to ensure that the generated image hash has a fixed-length. Resize the image using interpolation (bilinear).

- *Low pass filtering:*

The purpose of low-pass filtering is to alleviate influences of minor image modification on the final hash value. The modifications include noise contamination and filtering. A low-pass filter, also called a "blurring" or "smoothing" filter, averages out rapid changes in intensity. The simplest low-pass filter just calculates the average of a pixel and all of its eight immediate neighbours. The result replaces the original value of the pixel. The process is repeated for every pixel in the image.

2. *Formation of Secondary image:*

Secondary image is formed by reorganizing the original image pixels. The pre-processed image is divided into t non-overlapping blocks. '512 X 512' image is divided in to 64 non overlapping blocks each of 64 X 64. All blocks are rescaled to 8 x 8. The secondary image is formed by arranging the blocks.

3. *Non-Negative Matrix Factorization:*

Non-negative matrix factorization (NMF) has previously been shown to be a useful decomposition for multivariate data. Non-negative matrix factorization is a *linear*, *non-negative* approximate data representation. A non-negative matrix V of size *M* X *N* can be viewed as N column vectors, each sized M X 1. The aim of NMF is to find two nonnegative matrix factors, B of size *M* X *R* and C of size *R* X *N*, to approximately represent the original matrix V such that V ≈ BC be viewed as N column vector, each sized M X 1. Where B and C are called the base matrix and the coefficient matrix (or encoding matrix), respectively.

Usually r is chosen to be smaller than n or m, $R < \min(M; N)$ so that B and C are smaller than the original matrix V. This results in a compressed version of the original data matrix.

B can be regarded as containing a basis that is optimized for the linear approximation of the data in V. Since relatively few basis vectors are used to represent many data vectors, good approximation can only be achieved if the basis vectors discover structure that is latent in the data.

The new value of B or C is found by multiplying the current value by some factor that depends on the quality of the approximation.

$$B_{m,r} \leftarrow B_{m,r} \frac{\sum_{n=1}^{N} C_{r,n} V_{m,n} / (\mathrm{BC})_{m,n}}{\sum_{n=1}^{N} C_{r,n}} \quad (1)$$

$$C_{r,n} \leftarrow B_{r,n} \frac{\sum_{m=1}^{M} B_{m,r} V_{m,n} / (\mathrm{BC})_{m,n}}{\sum_{m=1}^{M} B_{m,r}} \quad (2)$$

where $m = 1; 2; : : : ;M$; $n = 1; 2; : : : ;N$; $r = 1; \quad 2; : : : ;R$.

B - Base Matrix.

C - Coefficient Matrix.

V – Secondary image.

4. *Generation of Hash value:*

To coefficient matrix of image (reference & presented) is used to generate the hash value.

Steps for generation and comparison of hash value

1) Convert the coefficient matrix (n X m) into 1 X nm matrix

2) Convert the 1 X nm matrix into binary matrix

$h_i = 0$ if $h_i(1,j) < h_i(1,j+1)$ 1

otherwise

5. *Hash Value Comparison:*

1) Calculate the hamming distance to measure similarity two hash values.

$d = \sum_{I=1}^{n} XOR (h1,h2)$

h1 - hash value of reference image

h2 - hash value of presented image.

2) If the distance is greater than pre-determined threshold (Th), corresponding images are different one.

III. EXPERIMENTAL RESULT

A. *PRESENT IMAGE IS TOTALLY DIFFERENT:*

Some standard color images sized 512£512, including cameraman, lena_gray, living room, mandril_gray, pirate, woman_blonde, woman_darkhair

TABLE I

HAMMING DISTANCE OF DIFFERENT IMAGE.

Sr. No.	Images	Hamming Distance
1	Cameraman-lena_gray	62
2	Cameraman-, living room	53
3	Cameraman-, mandril_gray	51
4	Cameraman-pirate,	63
5	Cameraman-woman_blonde	56
6	lena_gray-living room	67
7	lena_gray-mandril_gray	57
8	lena_gray--pirate	73
9	lena_gray-woman_blonde	56
	living room-	64

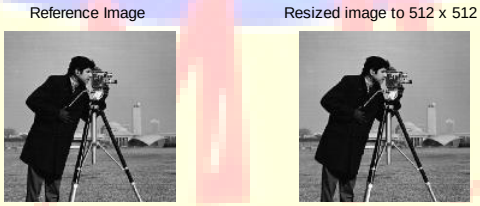
	mandril_gray	
	living room-pirate,	62
	living room-woman_blonde	53
	mandril_gray-pirate	62
	mandril_gray-woman_blonde	53
	Pirate-woman_blonde	55
	woman_blonde - woman_darkhair	43

From above result the threshold value

Th = 30.

The Final o/p:

Reference image:



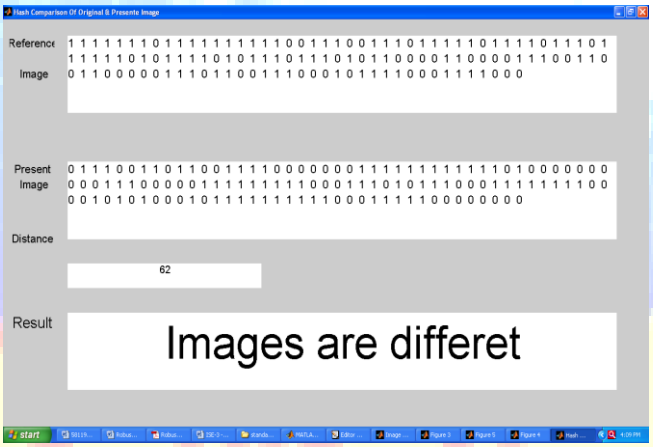
Presented image:

Presented Image

Resized image to 512 x 512

filtered image O/P 512 X 512

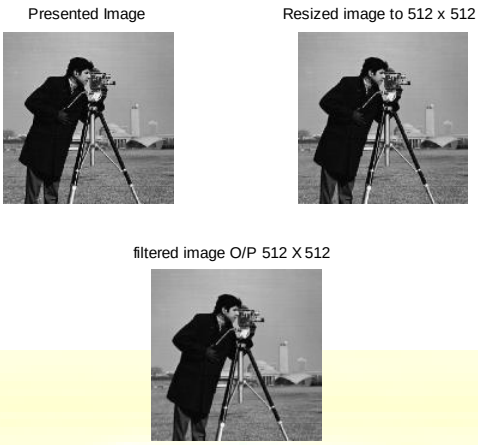
Check similarity of Hash & Comparison o/p:



B. Image scaling :

Image scaling is the process of resizing a digital_image. Scaling is a non-trivial process that involves a trade-off between efficiency, smoothness and sharpness. As the size of an image is increased, so the pixels which comprise the image become increasingly visible, making the image appears "soft". Conversely, reducing an image will tend to enhance its smoothness and apparent sharpness. Zooming requires: creation of new pixel locations, and the assign gray level to those location. Image shrinking is done in similar manner that for zooming. The equivalent process of pixel replication is row-column deletion.

Presented image scaled to 2 of its original size.



Check similarity of Hash & Comparison o/p:

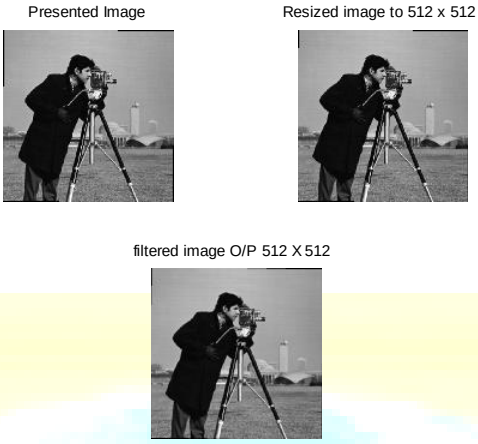
C. Rotating Image:

To rotate an image, use the imrotate function. imrotate accepts two primary arguments:

- The image to be rotated.
- The rotation angle

Rotates image A by angle degrees in a counter clockwise direction around its centre point. To rotate the image clockwise, specify a negative value for angle. 'imrotate' makes the output image B large enough to contain the entire rotated image. 'imrotate' uses nearest neighbour interpolation, setting the values of pixels in B that are outside the rotated image to 0 (zero). To make the o/p image of same size that of i/p use cropping.

Image rotated by 1°



Check similarity of Hash & Comparison o/p:

D. Image Enhancement:

The gray values of the image are mapped to another set of gray values, so that image is suitable for specific application. This is image enhancement.

Image Enhancement by gray level transformation:

The value of pixels before and after processing will be denoted by r and s, respectively.

$s = T(r)$

Where T is transformation that maps pixel value r to pixel value s

Power-Law transformation:

Power law transformations have basic form

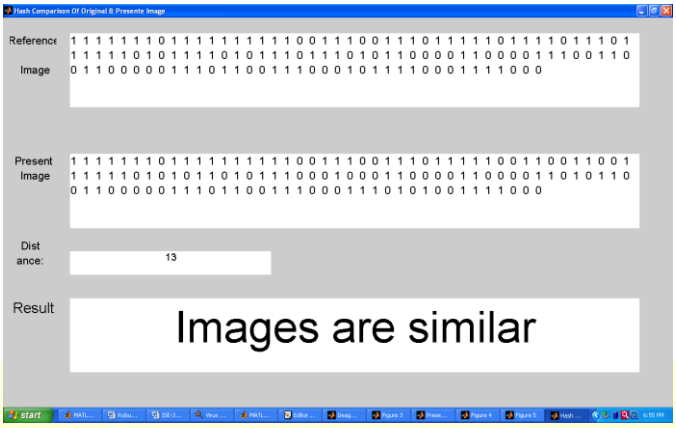
$$s = cr^{\gamma}$$

Where c, γ are positive constant. It reduce to identity transformation when c= γ=1.

Image after Enhancement:

After Pre-Processing the enhanced image:

Check similarity of Hash & Comparison o/p:



IV. CONCLUSION

By using a non-negative matrix factorization to extract image features, we have developed a perceptual image hashing scheme. A quantization rule can be defined to produce a binary string. This forms the image hash. The obtained hash is robust against perceptually acceptable image modifications such image scaling, image enhancement. Even 1˚ rotation in original image detects the change in image. Probability of collision between hashes of different images used for experiment is very low.

Further research on image modification such as image compression, watermark, image de-noising. Try to apply the algorithm for different types of image such as face images, different textures available with standard image data base. Check for threshold value for such different type of images.

REFERENCES

[1] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, Advances in Neural Information Processing Systems, vol. 13, pp. 556-562, 2000.

[2] Che-Yen Wen.; Kun-Ta Yang, Image authentication for digital image evidence, Forensic Science Journal 2006

[3] V. Monga and M. K. Mihcak, *Robust and secure image hashing via non-negative matrix factorizations*, IEEE Transactions on Information Forensics and Security, vol. 2, no. 3, pp. 376-390, 2007.

[4] Robust Image Hashing for Tamper Detection Using Non-Negative Matrix Factorization Z Tang, S Wang, X Zhang, W Wei, S Su - Journal of Ubiquitous Convergence Technology, 1,May2008 - VOL. 2,

[5] Image Database: - NRCS Photo Gallery, http://photogallery.nrcs.usda.gov or Ground Truth Database Available: http://www.cs.washington.edu/research/imagedatabase/ ground truth/ The USC-SIPI image database." http://sipi.usc.edu/database/, 2004. or other source.

[6] Matlab:- www.mathswork.com

[7] Digital Image Processing – 2nd Editions, Gonzalez & Woods

[8] Digital Image Processing Using Matlab-E.S. Gopi