Robust Image Hashing Scheme using Laplacian Pyramid

Hira Hamid*, Fawad Ahmed* and Jawad Ahmad†
* Department of Electrical Engineering, HITEC University, Taxila, Pakistan.
† School of Computing, Edinburgh Napier University, United Kingdom.

***Abstract*—Due to tremendous growth in multimedia applications and services, people can easily create, distribute, broadcast and store information. The fact that multimedia content can be easily copied and tampered has motivated a large number of researchers to work upon devising content and image verification techniques using *Perceptual Image Hashing* (PIH). In PIH, essential features of an image are extracted and a hash is calculated which is used for image verification. A PIH scheme should be resilient to non-malicious manipulations and capable to detect minute level tampering. In this paper, a PIH technique using Laplacian pyramid is devised. Laplacian pyramids are multi-scale representations of an image and can be used to extract stable features. In the proposed scheme, two different pyramids are generated by using filters of different diameters. The difference of Laplacian is calculated to get a unique and robust hash. A number of experiments have been carried out to gauge the effectiveness of the proposed scheme. The results reveal that the proposed technique is robust against non-malicious manipulations and can detect minute level tampering.**

I. Introduction

Due to tremendous growth in multimedia technologies, there has been a widespread increase in digital multimedia applications and services. People can create, distribute, broadcast and store information effortlessly and can share it over social media networks such as Facebook, Instagram, Youtube, Snapchat, etc., as per their desire. Due to digitization and easy to copy nature, digital data can be easily tampered. Hence multimedia content authentication has become an important factor. Multimedia content authentication means *deciding whether the given object matches the original object or not and whether it is authentic or not* [1]. Multimedia data is a bit stream with exact data values whereas multimedia content refers to the meaning or semantic of the data [2]. Multimedia object such as an image can be effected by two types of distortions; malicious and non-malicious as shown in Figs. 1 and 2. Malicious distortions are those in which pixels value are changed such that the content of an image is altered whereas non-malicious manipulations are those which changes the pixel values, but keep the meaning or semantic of an image intact [3]. For example in Fig. 1(b) the upper portion of the lens has been altered. This is an example of malicious manipulation. An image can have multiple digital representations that all look same to human perception. These different digital representations can occur due to different image processing operations such as compression, histogram equalization, etc.

(a) Cameraman Image (b) Tampered Image (tampering is shown inside the circle)

Fig. 1: Original image and its maliciously tampered version.

To ensure authenticity and verification of content integrity, a number of image hashing techniques have been proposed. [3]–[5]. An image hash converts the input image into a short string that is also called image digest [6]–[10]. For visually identical images, it produces same or closely related hash values. For images having different contents, it generates different hashes. Image hashing is different from image stenography in which data is hidden in an image [11] , [12] and the hidden information can be used to authenticate the input image. Traditionally, data verification issues were addressed by cryptographic hashes such as SHA1 and MD5 [13] which are sensitive to every bit of an input message. As a result, integrity of the content could be validated only when each and every bit of the input message is unchanged. Multimedia data, like digital images often undergoes content preserving operations or non-malicious distortions like compression, filtering, etc., that changes the value of pixels but generally keeps the semantic of an image intact. For example, Fig. 2 shows the Cameraman image processed through several non-malicious distortions. These operations change the pixel value but generally keep the semantic of an image intact. Therefore, bit by bit verification using traditional cryptographic hash functions for such images is not suitable for multimedia authentication and identification. For authentication of digital images, perceptual hash functions have been proposed to establish *"perceptual equality"* of multimedia content [14].

In perceptual image hashing, robust, unique and stable features of multimedia image are extracted and a hash value is calculated using these features [14]. In order to authenticate an image, hash value of an original image is compared with the image to be authenticated using specific hash functions.

(a) JPEG compression (b) Gaussian noise

(c) Speckle noise (d) Average blurring

(e) Gaussian blurring (f) Motion blurring

(g) Image sharpening (h) Gamma correction

Fig. 2: Illustration of Cameraman image and its non-maliciously distorted versions.

These functions calculate either distance or similarity between the two perceptual hashes. The result depends upon the chosen threshold. A perceptual hash is also known as a fingerprint, a passive fingerprint, a perceptual checksum, a robust hash, or a soft hash. Perceptual hash is called passive fingerprint because the multimedia content itself does not change as hash does not embed any watermark. However, hash needs to be transmitted before or after the image that adds an additional overhead during transmission. Given an image I from a database D and its perceptually similar copy with small perturbations I_t, the image hash is given by h which may additionally depend upon a secret key k. To make the notation general, the symbol h_k is used to represent a perceptual hash function when it depends upon the secret key k [15], [16].

Lin and Chang [5] used mean value of Discrete Coefficient Transform (DCT) to propose a hashing algorithm. The scheme is resilient to non-malicious modifications, however, it is sensitive to malicious distortions. Sun and Chang [17] proposed an algorithm using cryptographic hash function to embed a watermark for image authentication. Swaminathan *et al*. [4] presented a robust and secure image hashing technique using Fourier transform features and controlled randomization. The technique is robust against non-malicious manipulations and secure against malicious manipulations such as estimation and forgery attacks. Ouyang *et al*. [18] used Quaternion Discrete Fourier Transform (QDFT) and generated a hash which is robust against rotation and commom image preserving operations. Bhattacharjee and Kutter [19] presented a model in which Discrete Wavelet Transform (DWT) is used for extracting the feature points. This scheme is resistant to malicious manipulations, however, it shows some errors in detecting the correct location of the feature points due to wrap around effect of wavelet transform. Monga and Evans [20] proposed a wavelet-based iterative feature detection algorithm. This scheme is robust against face morphing, noise and object addition. However, the feature points generated are not effective enough to cover the background, which makes its performance vague under small tampering. Zhao and Wei [21] used property of rotation invariance of magnitudes and phases of Zernike moments to generate a robust image hashing technique. This technique is efficient in detecting image forgery involving structural modifications. Zhao *et al*. [22] combined Zernike moments and Local features and proposed an image hashing scheme that can identify image forgery and its location. However, performance of the proposed scheme highly depends upon the accuracy of saliency detection. Tang *et al*. [23] used ring partitioning and Non-negative Matrix Factorization (NMF) to generate a rotation invariant hash.

Two important requirements of image hashing is robustness to non-malicious distortions and ability to detect tampering. Features that are used to generate an image hash plays a pivotal role to achieve these properties. Though most of the state-of-the-art image hashing algorithm possesses high robustness, however, it is not clear that along with high robustness, how much minute level of tampering could be detected by their schemes. This paper attempts to fill this gap by devising an

image hashing scheme which besides being robust to non-malicious distortions can also detect minute level of tampering. In this paper, Laplacian pyramids are used to generate image hash. A Laplacian pyramid decomposes an image into multiple scales. This property is used to devise a new PIH scheme which is both robust to content preserving operations and sensitive to detect minute level tampering. Following are the main contributions of this paper:

1) A robust PIH scheme using Laplacian pyramid decomposition has been proposed along with detailed experimental results.
2) The receiver operating characteristic analysis is performed which reveals that the proposed scheme provides high robustness against non-malicious manipulations and can detect minute level of malicious modifications.

The rest of paper is organized as follows: Section 2 illustrates the proposed scheme. Experimental results are presented in Section 3. Finally, the paper is concluded in Section 4.

II. The Proposed Scheme

In this paper, a new method to construct robust image hash using Laplacian pyramid is proposed. Laplacian pyramids are filter based, multi-scale representations and can be effectively used in image hashing. It provides information regarding edges; hence it is widely used for image analysis by decomposing images into multi-scales. Construction of Laplacian decomposition pyramid is not complex as it can be implemented using successive image resizing operations. This feature makes it suitable for image hashing. The lower levels of the Laplacian pyramid contain coefficients exhibiting detailed representation of the input image, whereas at higher levels, less detail is available. However at higher levels, the pyramid coefficients are generally more robust to non-malicious manipulations like compression, filtering, etc. Robustness increases as the decomposition level increases whereas discrimination capability decreases. There is a trade-off between robustness and discrimination. If robustness is to be increased by choosing a higher level in the pyramid for hash construction, then the generated hash will not be very sensitive to detect minute level tampering. Another important aspect of using Laplacian pyramid is the size of the hash. As the level of a pyramid is increased, the size of hash decreases. For example, if the input image is of size 256×256 pixels and level 2 is used, then size of the hash will be 128×128. The hash will contain 16384 data points. At this level, the hash will have more discriminative capability to detect malicious tampering but will be less robust to withstand non-malicious distortions. In case Laplacian pyramid level 4 is used, the hash size will be 32×32; the hash will contain only 1024 entries. At this level, the feature coefficients obtained to generate the hash will be more robust to withstand non-malicious distortions; however, their capability to detect minute level tampering will be less as compared to hash features obtained using level 2 Laplacian pyramid decomposition. Level 5 contains only 256 features, but exhibit very poor discrimination capability to detect malicious tampering. After doing a number of experiments, Laplacian pyramid level 4 was found to be the best in terms of robustness, tamper detection and size of hash.

To obtain hash of an image, two different Laplacian pyramids are generated up to level 4 by using disk filters of different diameters. The difference of Laplacian is then calculated at level 4 and the difference is used as a hash of the image. The initial experiments also revealed that disk filter with radii 0.8 and 6 give best results in terms of robustness and minute level tamper detection. Figure 3 illustrates the idea of using difference of Laplacian decomposition for the generation of image hash. The proposed scheme consists of two modules which are explained in the following sections.

A. Hash Generation Module

The block diagram of hash generation module is given in Fig. 4 and the steps are mentioned below.

1) Firstly, an arbitrary input image I of size $M \times N$ pixels is preprocessed. It is converted into a gray scale image and then standardized to 256×256 pixels. In this step, I_o is obtained as given by Eq 1.

$$I \rightarrow I_o \tag{1}$$

2) The processed image I_o is then subjected to Level 4 Laplacian pyramid decomposition to get N_1. To obtain Laplacian of the input image, the input image is downsampled then upsampled and finally it is subtracted from the original image. Let I_0 be the original image, I_1, I_2, I_3 and I_4 are the downsampled and blurred versions of the original image, I_1',I_2', I_3' and I_4' are the upsampled and blurred versions of I_1, I_2, I_3 and I_4, respectively. If L_1, L_2, L_3 and L_4 are the first, second, third and fourth level Laplacian, then Laplacian pyramid at Level 4 is calculated as:

$$L_1 = I_0 - I_1', \tag{2}$$

$$L_2 = I_1 - I_2', \tag{3}$$

$$L_3 = I_2 - I_3', \tag{4}$$

$$L_4 = I_3 - I_4'. \tag{5}$$

The filter used for blurring in this step is a disk filter, which is a circular averaging filter. The radius of the disk filter is taken as 0.8. Let the fourth level Laplacian be represented by

$$N_1 = L_4. \tag{6}$$

3) The image I_o is again subjected to Level 4 Laplacian decomposition to obtain N_2, but this time, the radius of the disk filter used for blurring is taken as 6 instead of 0.8. Laplacian decomposition for this step is given as follows:

$$L_1' = I_0 - \bar{I}_1'. \tag{7}$$

$$L_2' = \bar{I}_1 - \bar{I}_2'. \tag{8}$$

$$L_3' = \bar{I}_2 - \bar{I}_3'. \tag{9}$$

$$L_4' = \bar{I}_3 - \bar{I}_4'. \tag{10}$$

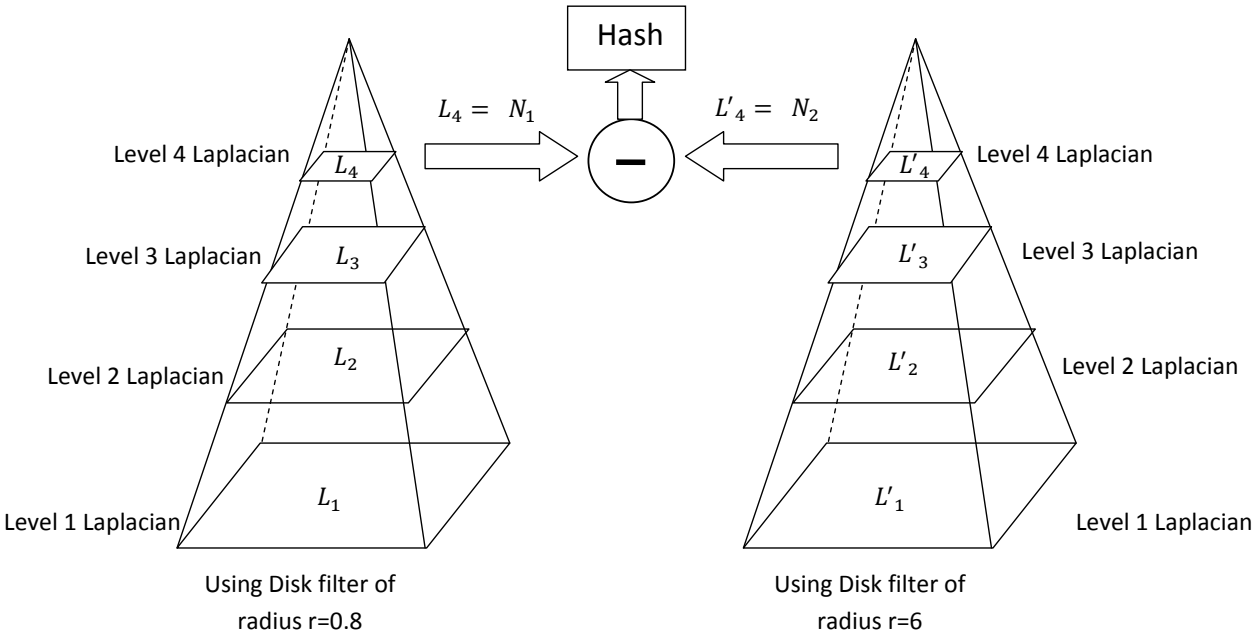


Fig. 3: Illustration of image hash by using Level 4 Laplacian of different radii.

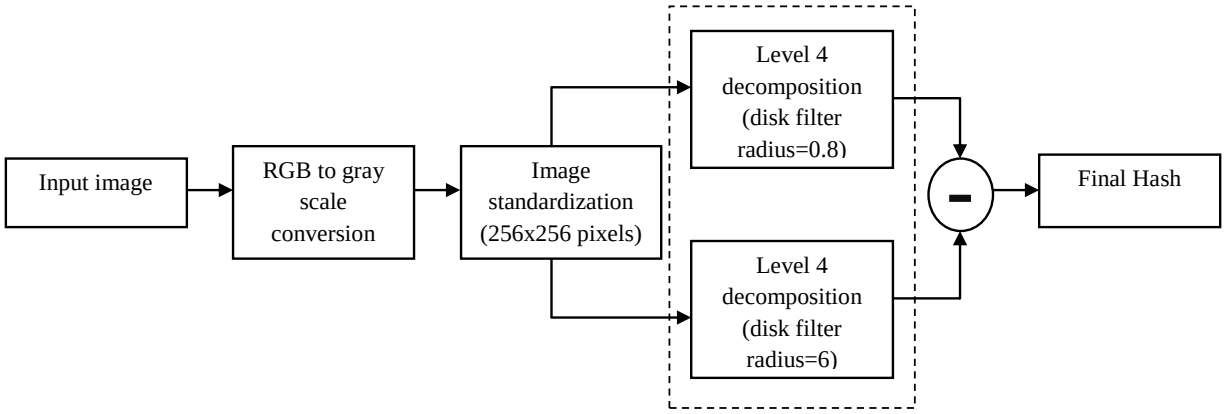


Fig. 4: Flow chart of hash generation module.

$$N_2 = L'_4, \tag{11}$$

where L'_1, L'_2, L'_3 and L'_4 are the first, second, third and fourth level Laplacian, respectively. $\bar{I}_1$, $\bar{I}_2$, $\bar{I}_3$ and $\bar{I}_4$ are the down-sampled and blurred versions of the original image. $\bar{I}'_1$, $\bar{I}'_2$, $\bar{I}'_3$ and $\bar{I}'_4$ are the up-sampled and blurred versions of $\bar{I}_1$, $\bar{I}_2$, $\bar{I}_3$ and $\bar{I}_4$, respectively.

4) Finally N_1 and N_2 are subtracted from each other and their absolute value is taken. This absolute value is the final hash h as given below.

$$h = |N_1 - N_2|. \tag{12}$$

Level 5 Laplacian decomposition can also be applied in Steps 2 and 3. The size of Level 5 decomposition is 16×16 for a 256×256 input image. Although this decreases the size of hash, but it also decreases the capability of tamper detection. For Level 4 Laplacian, the final size of image hash is 32×32 and hash formation is illustrated in Fig. 5. As discussed earlier, disk filter with radii 0.8 and 6 yield best results. The disk filter of radius 0.8 causes less blurring. When radius is increased to 6, the blurring effect of the disk averaging filter increases. This helps to get robust features that are also sensitive to tamper detection. This difference in two Laplacian of Level 4 to generate image hash is the key idea in this paper.

B. Hash Verification Module

1) The received image is subjected to Level 4 Laplacian pyramid decomposition and hash H' is calculated using Eqs. 6, 11 and 12.

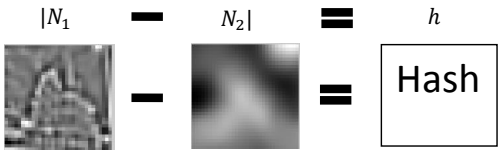


Fig. 5: Formation of final hash.

2) A difference matrix d is calculated as follows:

$$d = |H - H'|, \tag{13}$$

where H is the received hash and H' is the calculated hash of the received hash.

3) In the final step, each element of the difference matrix d is compared with a chosen threshold, t_r. If any element of d is greater than t_r, then the corresponding spatial area of the image would be considered as tampered.

III. Experimental Results

The evaluation of the proposed scheme by illustrating results of all malicious and non-malicious modifications is presented in the subsequent sections. Robustness and tamper detection results are discussed. All the experiments, i.e., robustness and tamper detection tests are applied on eight test images shown in Fig. 6. These images are selected to encompass variations in image contrast and texture.

A. Tamper Detection

In order to evaluate the capability of tamper detection, hashes of test images (Fig. 6) are compared with their tampered versions (Fig. 8), respectively. For all the images, tampered area has been encircled. Firstly, hash of original image is calculated, then hash of tampered image is calculated and finally maximum of absolute difference (d_{max}) is calculated between the two hashes. The results are tabulated in Table I. It is pertinent to mention that the ratio of tampered area with respect to total area of the image is less than 2% in all cases which is very small if compared with the tampering generally shown in the literature [20], [22] and [24]. As shown in Table I, the minimum absolute difference values for tamper detection is 28. This means that the value of threshold t_r should be less than d_{max}. Hence t_r should be less than 28 in order to detect minute level tampering of approximately 2% for the sample images taken into consideration. To further show how

TABLE I: Value of d_{max} when original image hash is compared with the hash of its maliciously tampered version.

Tampered images	Maximum of absolute difference
Cameraman image	28
Window image	38
Track image	31
Lena image	45
Bird image	39
Leaf image	100
Baboon image	32
Lion image	45

(a) Cameraman Image (b) Window Image

(c) Track Image (d) Lena Image

(e) Bird Image (f) Leaf Image

(g) Baboon Image (h) Lion Image

Fig. 6: Test Images.

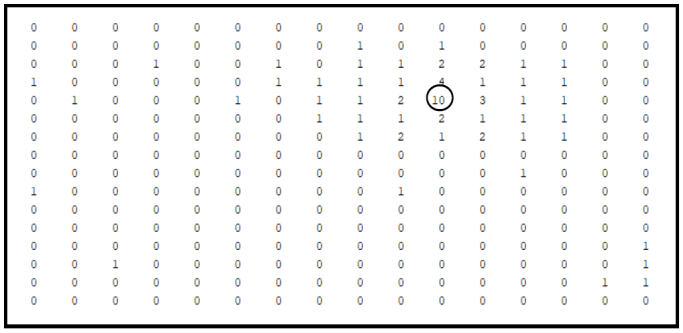
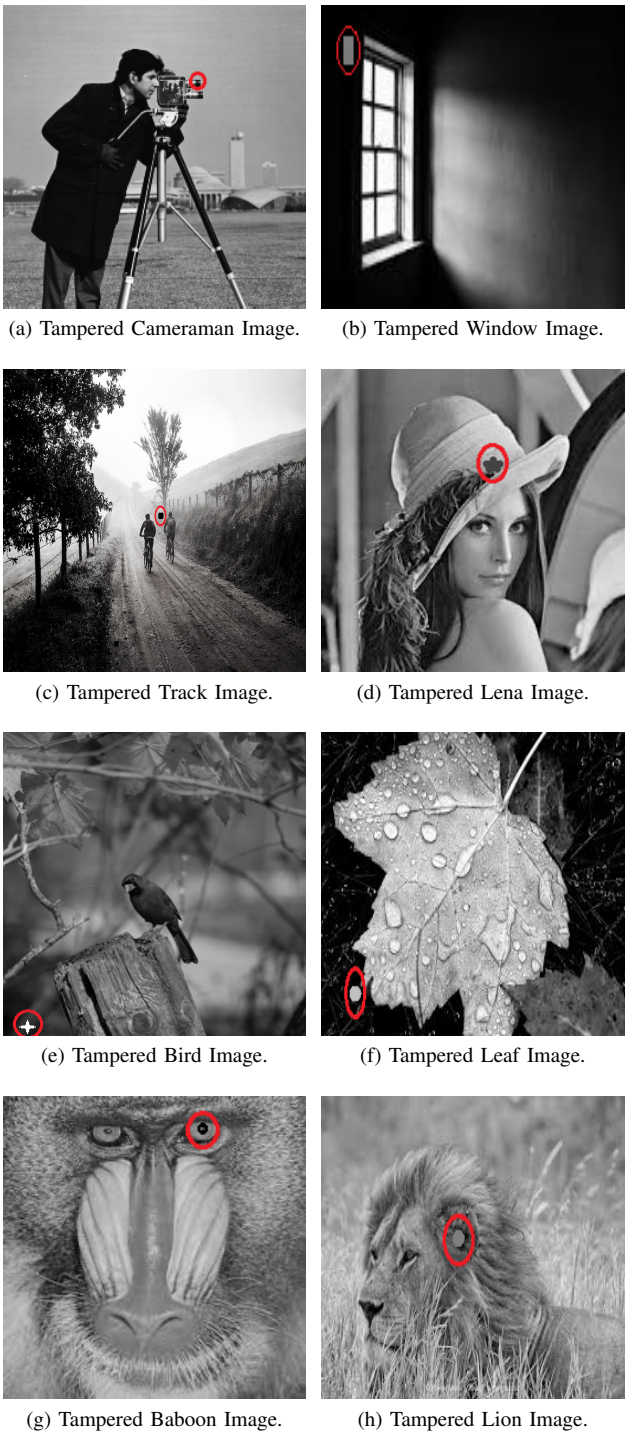


Fig. 7: Difference matrix between original and tampered Cameraman image.

the proposed scheme works, an example is now presented using the Cameraman image and its tampered version. These two images are shown in Figs 6a and 8a, respectively. In this example, for the ease of illustration, level 5 Laplacian pyramid is used to obtain the hash coefficients. Therefore, the hash size is 16×16 for a 256×256 input image. Figure 7 shows the difference matrix d obtained by calculating the absolute difference between the hash of the original Cameraman image and its tampered version. If the system threshold is selected as 7, then the number 10 encircled in Figure 7 would be a potential case of tamper detection as it is greater than 7. The element 10 is at position (5, 11). Each element of the difference matrix d represents 16×16 spatial area of the original image. This means that the corresponding tampered area in the original image is at the spatial position (64, 160) to (80,176). This 16×16 area can be highlighted in the input image to identify the detected tampering.

B. Robustness

In order to demonstrate robustness of the proposed scheme, hash of the original image was compared with the hash of manipulated versions of the same image. A number of content preserving distortions were applied such as noise, blurring, luminance changes, geometric attacks and filtering. The parameters such as variance (σ) of Gaussian noise, quality factor (Q) of JPEG compression, radius (r) of average blurring, etc., were varied. Maximum of absolute difference (d_{max}) between original image hash and distorted image hash was calculated for all the images. Graphs are plotted for d_{max} vs changing parameters as shown in Fig. 9. For example, JPEG compression at different quality factors was applied to all the test images and the maximum value of absolute difference was observed. Similarly, the test images were subjected to other non-malicious distortions such as Gaussian noise, Gaussian blur, etc. The idea is to thoroughly test robustness of the proposed hashing scheme under the effect of different non-malicious distortions. Table II shows different types of non-malicious distortions along with their range. The graphs shown in Fig. 9 reveal that there is a gradual increase in the value of d_{max} when the intensity of non-malicious distortions increases. Speckle noise shows abrupt behaviour after noise variance,



(a) Tampered Cameraman Image. (b) Tampered Window Image.

(c) Tampered Track Image. (d) Tampered Lena Image.

(e) Tampered Bird Image. (f) Tampered Leaf Image.

(g) Tampered Baboon Image. (h) Tampered Lion Image.

Fig. 8: Tampered versions of test images.(Tampering is shown inside the circles.)

TABLE II: Range of non-malicious distortion parameters.

Distortion type	Control parameter	Range
Gaussian Noise	Variance *(v)*	*v = 0.005 to 0.05*
Speckle Noise	Noise Variance *(Nv)*	*Nv= 0.006 to 0.06*
Gaussian Blurring	Standard Deviation (σ)	σ = 0.2 to 2
Motion Blurring	Linear Motion by pixel(L)	L=1 to 10
Average Blurring	Radius *r*	*r* = 1 to 10
Gamma Correction	Gamma (γ)	γ = 0.2 to 2
Image Sharpening	Sharpening amount *Fh*	*Fh* = 0.3 to 0.3
JPEG Compression	Quality factor *(Q)*	*Q = 10 to 100*

Nv=0.04. Average blurring and motion blurring change significantly by changing the radius and linear motion respectively. In case of gamma correction, the curve is more steep when gamma value is less than 1.

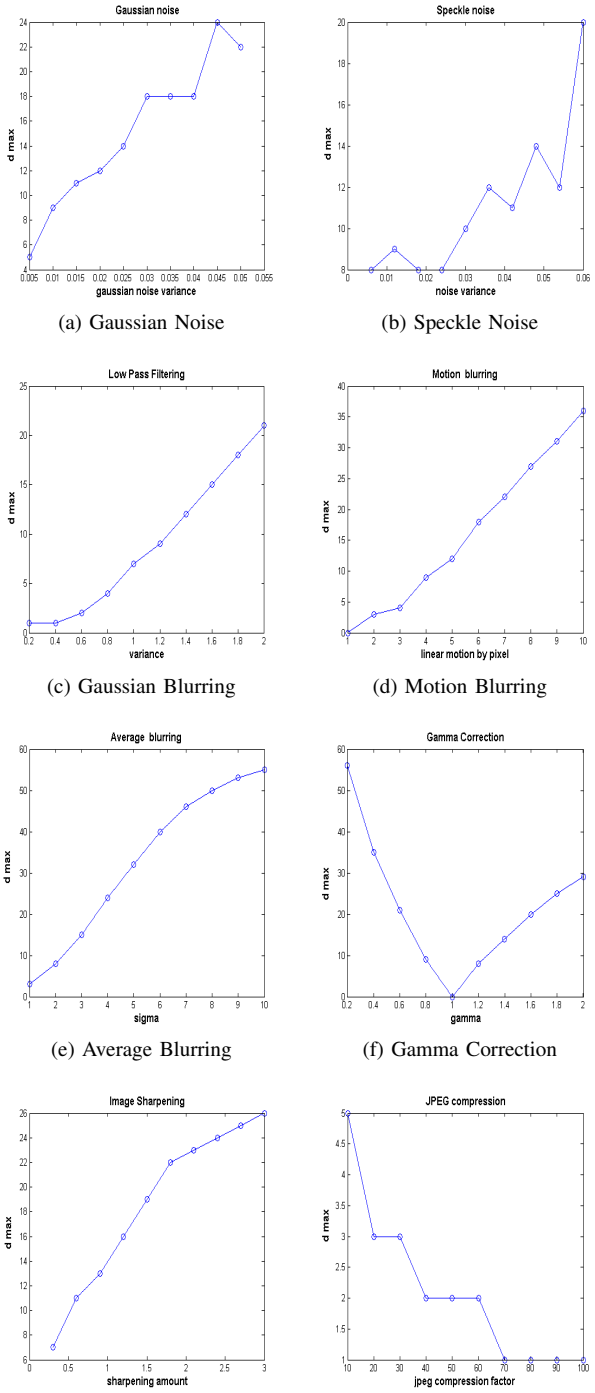
IV. System Threshold

In the proposed scheme, the size of image hash is 32×32. If two images are identical, then ideally all the values of the d matrix will be zero and would increase when the perceptual similarity between two images decreases. Hence a suitable threshold needs to be identified in order to differentiate malicious tampered images from non-malicious images. Let d_{ij} be the i^{th} row and j^{th} column of the difference matrix. If v is used for verification of image, '1' represents authentic image and '0' represents non-authentic image, then the relation of v, d and t_r is given by Eq 14.

$$v = \begin{cases} 1 & if \quad d_{ij} \leqslant t_r \qquad for \quad all \quad i,j. \\ 0 & if \quad d_{ij} > t_r \qquad for \quad all \quad i,j. \end{cases} \tag{14}$$

It has always been a challenge to determine a suitable threshold because robustness and discrimination capability have inverse relationship. There is a trade-off between robustness and tamper detection capability. From the tamper detection results shown in Table I, it is evident that the smallest value of d_{max} is 28. If for example, t_r is chosen as 28, this means that it is necessary that the value of d_{max} for all the non-maliciously distorted images should be less than 28. The result of non-malicious distortions in Fig. 9 shows a range of distortion parameters and the corresponding value of d_{max}. In fact, the maximum distortion parameter could be higher than what is shown in Fig. 9. An obvious question which comes in mind is that what should be the maximum value of non-malicious distortion parameter to gauge the robustness of the proposed scheme. A small value would make the scheme to appear more robust; but this would be a biased selection.

To solve this problem, a simple experiment was performed. The Cameraman image was selected as a test case and different non-malicious distortions with high value of distortion parameters was applied. The reason for selecting the Cameraman image is that it contains a lot of low texture regions thus making the distortions visibly prominent. For the purpose of illustration, results of several distortions are shown in Fig. 10. From the results, it is clear that such excessive distortions severely destroy the semantic of the image. We



(a) Gaussian Noise (b) Speckle Noise

(c) Gaussian Blurring (d) Motion Blurring

(e) Average Blurring (f) Gamma Correction

(g) Image Sharpening (h) JPEG Compression

Fig. 9: Plots of maximum of absolute difference vs changing parameter for different non-malicious distortions.

(a) Gaussian Noise v=0.045

(b) Speckle Noise Nv=0.06

(c) JPEG Compression Q=5

(d) Average Blurring r=4

(e) Motion Blurring L=7.5

Fig. 10: Illustration of extreme level distorted versions of Cameraman image.

feel that choosing a threshold which allows such high non-malicious distortion is practically not correct. Table III shows the value of non-malicious distortion parameters chosen after doing a number of experiments. The distorted versions of the Cameraman image after applying non-malicious distortions with parameters enlisted in Table III are shown in Fig. 11.

The chosen distortion parameters are very much consistent with the ones reported in literature, for example, [22], and [25]. Interestingly, the corresponding value of d_{max} is smaller than 28 for the chosen distortion parameters. This result is very encouraging as it shows that the proposed scheme can detect minute level tampering along with good robustness capability. The parameters shown in Table III are subsequently used for ROC analysis of the proposed scheme. The results of non-malicious manipulations when applied on all test images are tabulated in the Table IV. To get a more general idea as to how the proposed scheme reacts to non-malicious distortions, tests were also applied on a set of 100 images whose results are tabulated in Table V.

(a) Gaussian Noise

(b) Speckle Noise

(c) Gaussian Blurring

(d) Motion Blurring

(e) Average Blurring

(f) Gamma Correction

(g) Image Sharpening

(h) JPEG Compression

Fig. 11: Non-malicious distorted versions of Cameraman image taken into considerations.

The highest value of d_{max} obtained is 10. Hence, considering these results, it is concluded that if an image needs to be authenticated positively for the type of distortions considered, the value of threshold t_r should be taken as 20. With this threshold, the proposed algorithm would be highly robust towards non-malicious manipulations and detect minute level tampering.

V. The Receiver Operating Characteristics Curve

The performance of the proposed scheme is evaluated using the Receiver Operating Characteristics Curves (ROC). The ROC curve is plotted between *False Acceptance Rate* (*FAR*) and *False Rejection Rate* (*FRR*) w.r.t to changing threshold. Equations 15 and 16 define *FAR* and *FRR*.

$$FAR = \frac{I_{TDA}}{I_T} \tag{15}$$

$$FRR = \frac{I_{ADT}}{I_A} \tag{16}$$

In Eq 15, I_{TDA} represents the total number of tampered images detected as authentic and I_T represents the total number of images. Similarly in Eq 16, I_{ADT} represents the total number of authentic images detected as tampered and I_A represents total number of authentic images. False acceptance rate is the probability of a system to detect manipulated images as authentic images whereas, false rejection rate depicts the probability of detecting authentic images as manipulated images. For a system to be efficient, its false acceptance rate as well as false rejection rate should be small. If a system has high false acceptance rate then there will be a high risk of wrong image verification, thus decreasing the tamper detection capability of a system. On the other hand, high value of false rejection rate would frequently reject genuine images and will decrease the robustness of an authentication scheme. There is always a trade-off between FAR and FRR as they are inversely proportional to each other. This trade-off quantifies robustness and tamper detection capability of a scheme. Therefore, it is required to balance robustness and tamper detection at a desired value. The operating point of an ROC curve depends upon application of a hashing system. If a user needs the hashing system in which any sort of tampering is detected, then in this case, the false acceptance rate needs to be zero, which in turn would significantly increase the false rejection rate.

A. FAR and FRR Estimation

To estimate FAR, a database of 100 images was created. Ten different images were selected and each image was subjected to ten different types of tampering. The size of tampering was kept less than 2% of the total image area. This size in general is very small if we compare it with the size of tampering done in several papers, for example [20], [22] and [24]. The FAR for the system was calculated by comparing hash of the original image with its ten tampered versions. Hence, there were 100 comparisons in total. Then number of

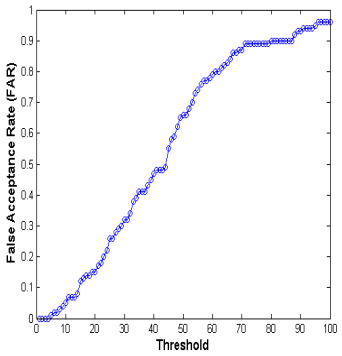
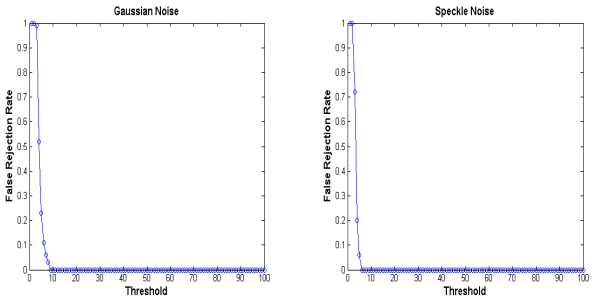
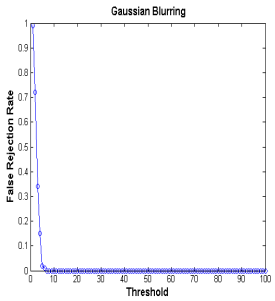


Fig. 12: Plot of False Acceptance Rate (FAR) vs Threshold



(a) Gaussian Noise (b) Speckle Noise



(c) Gaussian Blurring

Fig. 13: Plots of False Rejection Rate vs Threshold.

images that were authenticated as positive were noted and the false acceptance rate was calculated using Eq. 15. The FAR plot is shown in Fig. 12, when the system threshold was varied from 1 to 100. FAR increases with the increase in threshold (t_r). FAR is zero at threshold equal to zero. When threshold is between 1 to 10, the value of FAR varies from 0 to 0.05. At high threshold, the value of FAR increases mainly due to the fact that the tampering area considered in the paper is small. To obtain FRR, 100 different images were taken and non-malicious distortions were applied to each image. A total of 800 images were used to estimate FRR. The hash of original image was compared with the hash of the distorted image to obtain the number of images detected as non-authentic. False rejection rate was calculated by using Eq. 16 by changing t_r from 1 to 100. Plots of false rejection rate with respect to threshold are shown in Figs. 13 and 14.

FRR decreases as the threshold increases. In case of Gaussian noise and speckle noise, FRR remains 1 till threshold t_r = 5,

TABLE III: Non-malicious distortion parameters.

Distortion type	Details	Control parameters	Specifications
Noise	Gaussian Noise	Mean *(m), Variance (v)*	*m = 0,v = 0.01*
	Speckle Noise	Noise Variance *(Nv)*	*Nv= 0.02*
Blurring	Gaussian Blurring	Standard Deviation (σ), Window size *(Fs)*	σ = 1.6, Fs = [11x11]
	Motion Blurring	Linear Motion by pixel (L), Angle (θ)	L=5, θ = 90
	Average Blurring	Radius r	r = 3
Luminance Changes	Gamma Correction	Gamma (γ)	γ = 1.2
	Image Sharpening	Radius (r), Sharpening amount (Fh)	r = 2, $Fh = 1$
Geometric Attacks	JPEG Compression	Quality factor *(Q)*	*Q = 10*

TABLE IV: Value of d_{max} for high contrast images.

Non malicious distortions	d_{max}							
	Cameraman image	Window image	Track image	Lena image	Bird image	Leaf image	Baboon image	Lion image
Gaussian Noise *m = 0,v = 0.01*	5	8	6	4	3	6	4	5
Speckle Noise *Nv= 0.02*	4	11	3	5	3	6	4	4
Gaussian Blurring σ = 1.6, Fs = [11x11]	4	15	1	4	2	4	3	5
Motion Blurring L=5, θ = 90	2	12	1	2	1	3	2	3
Average Blurring *r*=3	4	15	1	4	2	4	3	4
Gamma Correction γ = 1.2	6	6	8	4	4	5	3	5
Image Sharpening *r*=2, *Fh*= 1	7	13	5	7	3	6	4	4
JPEG Compression *Q = 10*	3	5	3	3	3	3	3	4

TABLE V: Value of d_{max} for a set of 100 images.

Non malicious distortions	d_{max} for a set of 100 images
Gaussian Noise *m = 0,v = 0.01*	9
Speckle Noise *Nv= 0.02*	6
Gaussian Blurring σ = 1.6, Fs = [11x11]	7
Motion Blurring L=5, θ = 90	4
Average Blurring *r*=3	6
Gamma Correction γ = 1.2	10
Image Sharpening *r*=2, *Fh* =1	10
JPEG Compression *Q = 10*	5

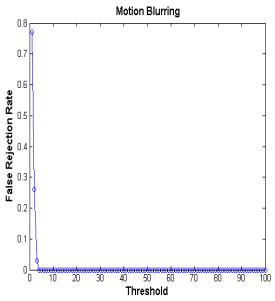
which means that most of the genuine samples will be rejected. For Gaussian blurring and average blurring, FRR becomes 0 at t_r = 5. This indicates that after threshold t_r=5, no genuine image will be rejected. For gamma correction and Gaussian noise, FRR becomes 0 at t_r=10. Hence, the scheme is robust against non-malicious manipulations. There is a rapid decrease in FRR for image sharpening and highpass filtering and it becomes 0 at t_r=10. The proposed scheme is highly robust against JPEG compression as FRR becomes 0 at t_r=4.

B. ROC Curves

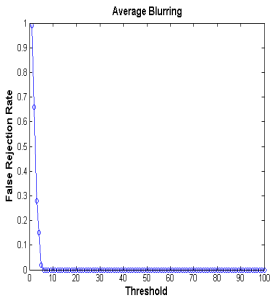
Figures 15, 16 and 17 shows the Receiver Operating Characteristic (ROC) curves for several non-malicious distortions like Gaussian noise, speckle noise, motion blurring, etc. These curves are obtained by plotting FAR on the x-axis and FRR on the y-axis. The values of FAR and FRR were calculated using Equations (15) and (16), respectively. To estimate FAR, a database of 100 images was created. Ten different images were selected and each image was subjected to ten different types tampering. The size of tampering was less than 2% of the total image area. Hence, there were 100 comparisons in total. By changing the threshold from 1 to 100, the values of FAR were calculated using Equation 15.

To obtain FRR, 100 different images were taken and were subjected to 8 different types of non-malicious manipulations mentioned in Table III, thus creating a database of 800 images for FRR estimation. Each image was compared with its eight distorted versions to calculate FRR using Equation 16 by varying the threshold from 0 to 100. For each value of threshold, the corresponding values of FAR and FRR were plotted to obtain the ROC curve. For each ROC curve shown in Figs. 15, 16 and 17, a specific non-malicious distortion was chosen to obtain FRR while all 100 tampered images were used to obtain FAR.

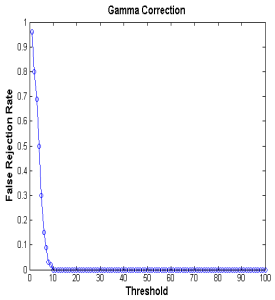
At low value of FAR, the value of FRR is high, exhibiting the fact that if the system is required to be extremely sensitive to detect malicious tampering, then it will reject genuine samples of the same image that were subjected to non-malicious distortions. For example, in case of image sharpening, Fig. 17a, the value of FRR is 0.5, in case if the required FAR is to be kept close to 0. This implies that if the threshold is kept such that the system is sensitive it should be sensitive to detect minute level tampering, then it would reject around 50% of the genuine images. On the other hand, if the system is to be made robust to withstand non-malicious distortions, then it will lose its tamper detection capability and may positively authenticate tampered version of the input



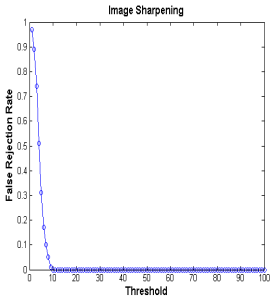
(a) Motion Blurring



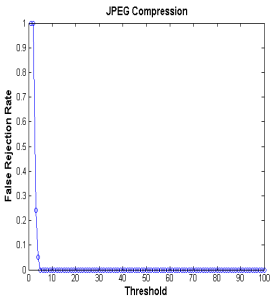
(b) Average Blurring



(c) Gamma Correction



(d) Image Sharpening



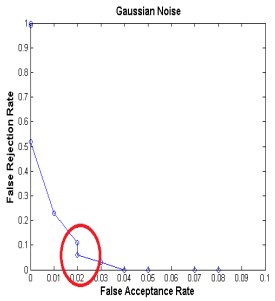
(e) JPEG Compression

Fig. 14: Plots of False Rejection Rate vs Threshold.

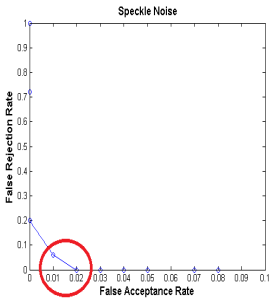
TABLE VI: Value FAR and FRR for non-malicious distortions.

Non malicious distortions	FAR	FRR
Gaussian Noise m = *0,v = 0.01*	0.02	0.06
Speckle Noise *Nv= 0.02*	0.01	0.06
Gaussian Blurring σ = 1.6, Fs = [11x11]	0.02	0.01
Motion Blurring L=5, θ = 90	0.01	0.05
Average Blurring r=3	0.01	0.02
Gamma Correction γ = 1.2	0.02	0.08
Image Sharpening r=2, *Fh*= 1	0.02	0.1
JPEG Compression Q = *10*	0.01	0

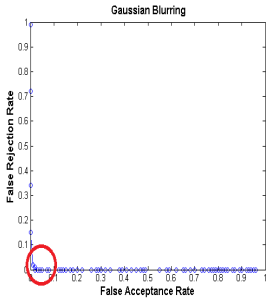
images. For example, in case of image sharpening, to obtain a low value of FRR like 0.01, the value of FAR is 0.04. To obtain a balance between robustness and tamper detection capability, the threshold should be adjusted such that both FAR and FRR are low. For the purpose of illustration, suitable values of FAR and FRR are encircled in Figs. 15, 16 and 17 for each case of non-malicious distortion. It is promising to note that low false acceptance and false rejection rates can be obtained simultaneously. The values of FAR and FRR are tabulated in Table VI.



(a) Gaussian Noise



(b) Speckle Noise



(c) Gaussian Blurring

Fig. 15: Receiver Operating Curves.

TABLE VII: Equal Error Rate for Non-malicious Distortions.

Non-malicious Distortions	Equal Error Rate (ERR)	Threshold
Gaussian Noise m = *0,v = 0.01*	0.03	8
Speckle Noise *Nv= 0.02*	0.03	9
Gaussian Blurring σ = 1.6	0.02	6
Motion Blurring L=5, θ = 90	0.01	5
Average Blurring r=3	0.01	5
Gamma Correction γ = 1.2	0.03	8
Image Sharpening r=2, *Fh*= 1	0.03	8
JPEG Compression Q = *10*	0.01	5

C. Equal Error Rate (ERR)

Equal Error Rate (EER) is a point where FRR is equal to FAR. EER helps the user to choose a suitable threshold. Since FRR and FAR have inverse relation, hence if one increases, the other decreases. If the threshold is increased, the FAR increases and FRR decreases. Hence robustness increases with the increase in threshold and tamper detection capability decreases. For finding out EER, FAR and FRR are ploted against threshold and the point of their intersection is noted. EER for different non-malicious manipulations is tabulated in Table VII.

All the distortions have different thresholds, where FAR is equal to FRR. Choosing the threshold accordingly makes the algorithm robust for the particular non-malicious distortion. For example, if threshold of value 4 is chosen, then the proposed algorithm will become highly robust against motion blurring. However, robustness against rest of the distortions will decrease at the given value of ERR.

D. Comparison of ROC Curves with Other Schemes

The results of proposed scheme are compared with the ROC curve results of three papers, [4], [20] and [26]. The first two

papers are among the most cited papers in image hashing and the third one is a recent paper. Although tampering in these papers is greater than the tampering used in this work, yet the proposed scheme shows very good results. FAR and FRR of the under consideration papers and of the proposed scheme are mentioned in Table VIII. In these papers, the result of non-malicious distortions were not shown separately rather the cumulative effect of all distortions was used to find FRR and FAR.

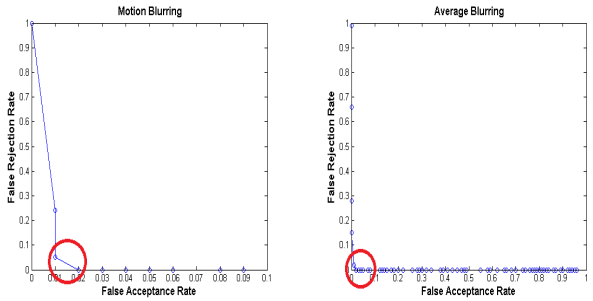
From the results mentioned in Table VIII, it is observed that for Gaussian blurring, average blurring and JPEG compression, the proposed scheme gives better FAR and FRR values than the schemes mentioned in [20] and [4], whereas JPEG compression outperforms all the schemes mentioned in Table VIII. For other non-malicious distortions, the results are quite close to the one reported in [20], [4] and [26].

VI. Conclusion

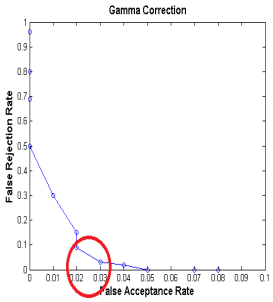
In this paper, a robust PIH scheme is presented using Laplacian pyramid decomposition. The proposed scheme uses fourth level Laplacian pyramid decomposition to generate a hash of size 32×32. A detailed analysis of the scheme has been carried out from two aspects; (i) perceptual robustness and (ii) tamper detection. For robustness analysis, a number of non-malicious distortions were used and their respective parameters were varied. The parameters upon which the scheme yields best results were suggested. It becomes very easy to select a suitable threshold under the defined parameters because there is sufficient gap between d_{max} of non-malicious and malicious modifications. ROC curves were plotted to analyze false detection and false rejection capability of the proposed scheme, when malicious and non-malicious manipulations are applied on the input images. ROC curves yield good results and low FRR and FAR were obtained. Thus, the proposed scheme is robust and can successfully detect tampering as small as 2% of the total image area. Comparison of the proposed scheme with some well known image hashing schemes also showed promising results. The proposed PIH scheme is however not rotational invariant. This limitation can be addressed by applying Fourier-Mellin transform in future work.

TABLE VIII: Value FAR and FRR for non-malicious distortions.

Other Schemes	FAR	FRR
Monga [20]	0.02	0.03
Swaminathan [4]	0.05	0.05
Qiang Ma [26]	0.06	0
Proposed Scheme	**FAR**	**FRR**
Gaussian Noise m = *0,v = 0.01*	0.02	0.06
Speckle Noise *Nv= 0.02*	0.01	0.06
Gaussian Blurring σ = 1.6, Fs = [11x11]	0.02	0.01
Motion Blurring L=5, θ = 90	0.01	0.05
Average Blurring r=3	0.01	0.02
Gamma Correction γ = 1.2	0.02	0.08
Image Sharpening r=2, *Fh*= 1	0.02	0.1
JPEG Compression Q = *10*	0.01	0

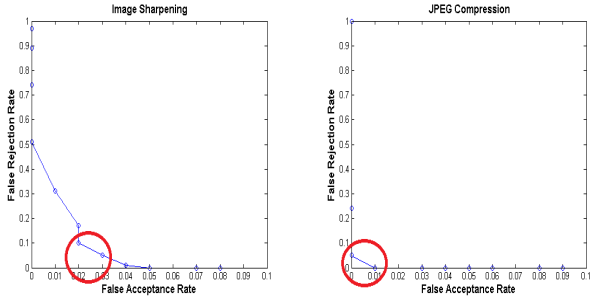


(a) Motion Blurring (b) Average Blurring



(c) Gamma Correction

Fig. 16: Receiver Operating Curves.



(a) Image Sharpening (b) JPEG Compression

Fig. 17: Receiver Operating Curves. a) ROC curve b) Enlarged view of ROC curve.

References

[1] N. Sidharthan, J. P. Jo, and E. George, "Robust image hashing for content authentication with tampering localization and image recovery," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, 2017, pp. 1767–1773.

[2] F. Ahmed, M. Y. Siyal, and V. U. Abbas, "A secure and robust hash-based scheme for image authentication," *Signal Processing*, vol. 90, no. 5, pp. 1456–1470, 2010.

[3] Z. Tang, X. Zhang, L. Huang, and Y. Dai, "Robust image hashing using ring-based entropies," *Signal Processing*, vol. 93, no. 7, pp. 2061–2069, 2013.

[4] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Transactions on Information Forensics and security*, vol. 1, no. 2, pp. 215–230, 2006.

[5] C.-Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing jpeg compression from malicious manipulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 153–168, 2001.

[6] Z. Tang, L. Chen, X. Zhang, and S. Zhang, "Robust image hashing with tensor decomposition," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[7] S. S. Kozat, R. Venkatesan, and M. K. Mihçak, "Robust perceptual image

hashing via matrix invariants," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 5. IEEE, 2004, pp. 3443–3446.
[8] M. K. Mıhçak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *ACM Workshop on Digital Rights Management*. Springer, 2001, pp. 13–21.
[9] R. F. Graveman and K. Fu, "Approximate message authentication codes," in *Proc. 3rd Annual Fedlab Symp. Advanced Telecommunications/Information Distribution*, vol. 1, 1999.
[10] C. Qin, C.-C. Chang, and P.-L. Tsou, "Robust image hashing using non-uniform sampling in discrete fourier domain," *Digital Signal Processing*, vol. 23, no. 2, pp. 578–585, 2013.
[11] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
[12] X. Liao, Z. Qin, and L. Ding, "Data embedding in digital images using critical functions," *Signal Processing: Image Communication*, vol. 58, pp. 146–156, 2017.
[13] B. Schneier, "Applied cryptography, 1996 john wiley & sons," *Inc, USA*.
[14] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," 2010.
[15] B. Coskun and N. Memon, "Confusion/diffusion capabilities of some robust hash functions," in *Information Sciences and Systems, 2006 40th Annual Conference on*. IEEE, 2006, pp. 1188–1193.
[16] R. Radhakrishnan and N. Memon, "On the security of the digest function in the sari image authentication system," *IEEE transactions on circuits and systems for video technology*, vol. 12, no. 11, pp. 1030–1033, 2002.
[17] Q. Sun and S.-F. Chang, "A robust and secure media signature scheme for jpeg images," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 41, no. 3, pp. 305–317, 2005.
[18] J. Ouyang, G. Coatrieux, and H. Shu, "Robust hashing for image authentication using quaternion discrete fourier transform and log-polar transform," *Digital Signal Processing*, vol. 41, pp. 98–109, 2015.
[19] S. K. Bhattacharjee and M. Kutter, "Compression tolerant image authentication." in *ICIP (1)*, 1998, pp. 435–439.
[20] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: performance evaluation and tradeoffs," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3452–3465, 2006.
[21] Y. Zhao and W. Wei, "Perceptual image hash for tampering detection using zernike moments," in *Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on*, vol. 2. IEEE, 2010, pp. 738–742.
[22] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using zernike moments and local features," *IEEE transactions on information forensics and security*, vol. 8, no. 1, pp. 55–63, 2013.
[23] Z. Tang, X. Zhang, L. Huang, and Y. Dai, "Robust image hashing using ring-based entropies," *Signal Processing*, vol. 93, no. 7, pp. 2061–2069, 2013.
[24] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance." *IEEE Trans. Information Forensics and Security*, vol. 11, no. 1, pp. 200–214, 2016.
[25] D. Wu and X. Niu, "A self-synchronized image hash algorithm," in *Communications and Mobile Computing (CMC), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 13–15.
[26] Q. Ma, L. Xu, L. Xing, and B. Wu, "Robust image authentication via locality sensitive hashing with core alignment," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7131–7152, 2018.