Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1645658, 17 pages
https://doi.org/10.1155/2022/1645658

Research Article

A Secure and Robust Autoencoder-Based Perceptual Image Hashing for Image Authentication

Abdul Subhani Shaik,[1,2] **Ram Kumar Karsh**,[1] **Mohiul Islam**,[3] **and Surendra Pal Singh**[4]

[1]*Department of ECE, National Institute of Technology Silchar, 788010, Assam, India*
[2]*Department of ECE, CMR College of Engineering & Technology, 501401, Hyderabad, Telangana, India*
[3]*School of Electronics Engineering, Vellore Institute of Technology, 632014, Vellore, Tamil Nadu, India*
[4]*Surveying Engineering Department, Wollega University, Nekemte City, Ethiopia*

Correspondence should be addressed to Surendra Pal Singh; surendra.geomatics@gmail.com

Received 28 May 2022; Revised 25 September 2022; Accepted 14 October 2022; Published 31 October 2022

Academic Editor: Shaohua Wan

Copyright © 2022 Abdul Subhani Shaik et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the advancement of technology, it has become easier to modify or tamper with digital data effortlessly. In recent times, the image hashing algorithm has gained popularity for image authentication applications. In this paper, a convolutional stacked denoising autoencoder (CSDAE) is utilized for producing hash codes that are robust against different content preserving operations (CPOs). The CSDAE algorithm comprises mapping high-dimensional input data into hash codes while maintaining their semantic similarities. This implies that the images having similar content should have similar hash codes. To demonstrate the effectiveness of the model, the correlation between hash codes of semantically similar images has been evaluated. Subsequently, tampered localization is done by comparing the decoder output of the manipulated image with the hash of the actual image. Then, the localization ability of the model was measured by computing the $f1$ scores between the predicted region and the original tampered region. Based on the comparative performance and receiver-operating characteristics (ROC) curve, we may conclude that the proposed hashing proposed algorithm provides improved performance compared to various state-of-the-art techniques.

1. Introduction

Recent developments in sophisticated image editing tools have made it very convenient for an impostor to tamper or forge the image contents. These tools allow us to add or remove content from an image very easily. The identification of manipulation becomes very important to establish image validity [1, 2]. In general, perceptual image hashing strategies resolve this problem. These techniques are used to extract the most important features from an image for calculating a hash. The hash codes may be produced by traditional hashing algorithms [3–5] like MD5 or SHA-256. However, these techniques are susceptible to the data, i.e., any bit changes result in different hash codes. This behavior is undesirable because digital images are constantly subjected to unintended improvements such as compression and enhancement. The objective of perceptional hash algorithms is to generate hash codes that take only changes in the district region into account. In contrast to image hashing algorithms, few other techniques such as watermarking [6], cryptography [7], and image encryption [8–10] have been developed to transmit data through a secured channel or hide it. However, with such methodologies, it is difficult to detect and localize the tampered region. The effectiveness of a perceptual hash can be measured by its robustness against various content preserving operations and its sensitivity to different malicious content removing or adding functions [11–14].

Autoencoders have been found as a very effective technique for unsupervised learning of image hash functions due to their ability to discover the essential features from unlabeled data. Most of these are fully integrated forward feed networks with a code layer regularization that allows the model to understand data collections rather than only

copying the input into the output. This motivated us to use a convolutional stacked autoencoder to tackle the issue of creating a hash with a perceptual image that is robust against enhancement and compression changes in the image while being sensitive to the content removing operations.

Recently, machine learning and deep learning techniques have been employed in various fields of image processing [15, 16]. Similarly, the autoencoder utilized in this article is based on an artificial neural network (ANN). The network is trained hierarchically to represent input images in the latent space having 1024 dimensions and mapping the same back to the dimensions of the input image. We rely on the ability of the autoencoder to learn the underlying features of the dataset without the need for labeled data. L2 regularization is applied to the hash code layer to help the model to better generalize the training dataset and to prevent the model from learning an identity function. The network is influenced by two effects of the L2 regularization that was chosen. Firstly, the hash code deletes unnecessary components by using the smallest combination to solve the problem of learning. Secondly, it eliminates the impact of static noise to improve the generalization of the model. Tampering detection is done by comparing the correlation between the initial image hash code and the hash code of the manipulated image with a typical threshold set to 0.98. Both of these hash codes are generated by the proposed model. Any image having a correlation less than the threshold is considered to tamper. The decoder part of the proposed model then compares the decoder outputs generated from the respective hash codes to create the probably tampered region in the images.

The robustness of the hash codes is evaluated by measuring the true-positive and false-positive rates for different CPOs on the image. The localization ability of the proposed model is evaluated by computing the $f1$ scores between the raw difference of both the real and the tampered image and the raw difference of the decoder outputs of the original image and the tampered image.

2. Related Work

2.1. Local and Global Feature-Based Hashing. In the last few decades, several research works have been done in the field of image hashing. In 2005, Monga and Evans [1] utilized both visually significant features and used probabilistic quantization features to construct the robust image hash, and the disadvantage associated with this method is that it does not allow the exploration of alternative image recognition and representation based on pseudorandom signals. Later on in 2007, Monga and Mihçak [3] utilized the nonnegative matrix factorization (NMF) method for generating image hashes. However, the primary disadvantage of this approach is its computational time. In this method, the time taken for hash extraction was 2.03 seconds when the hash length was 300 bits. Likewise, in 2006, Swaminathan et al. [2] also proposed a new algorithm based on Fourier transformation and supervised randomization for the generation of an image hash. This technique suffers from the limitation of randomized quantization.

In 2009, Wu et al. [4] built a Radon and wavelet transform-based printed-scan-resistant image hashing algorithm. In the same year, a virtual watermark detection-based image hashing technique was proposed by Khelifi and Jiang [5]. However, these techniques failed to address various geometric attacks such as shearing and were unable to provide adequate performance. In 2010, Ahmed et al. [17] added a hidden key for modulating pixels dynamically, leading to transformed space. The picture hash is then computed using the key-dependent transformed function space. A 4-bit quantizing scheme to reduce the hash is also proposed; however, the drawback of this approach is that it does not resist other criteria such as brightness changes, contrast improvement, and tampering that involves smooth changes in gray level values.

2.2. Transform-Based Hashing. In 2011, Lei et al. [11] incorporated the Radon transform (RT) along with DFT for constructing the robust image hash. In 2011, Tang et al. and Choi and Park [12, 13] suggested a method for creating image hashes based on a lexicographically structured architecture. Dictionary development and upkeep, as well as hash production, are two aspects of the scheme, but the shortcoming of this method is that it does not address issues such as image rotation, color features, more complex dictionary creation, and mechanisms related to maintenance. To build the image hash, in 2012, Li et al. and Lv and Jane Wang [14, 18] presented a solid hash function dependent on dithered lattice vector quantization and random Gabor filtering. In 2013, Tang et al. [19] discussed a method that produces a hash by transforming the original image into a normalized version, i.e., by separating the image into sections and obtaining the entropies based on rings. Afterward, in 2014, Tang et al. [20–24] developed another effective image hash using a ring partition and an NMF. But the drawback of this technique is that it is not rotation-invariant. Moreover, it cannot address issues such as detection of tampering in small areas, localization of tampering, and effective extraction of color attributes.

In 2015, Sebastian et al. [25] proposed a technique for hashing images that use Haralick and modified local binary pattern features, as well as luminance and chrominance channels. In the same year, Ouyang et al. [26] utilized log-polar and Quadrature DFT transform for the generation of image hash, but both these algorithms are sensitive to geometric operations. In 2016, various image hashing methods were developed based on invariant vector distance and ring partition [27], adaptive and local feature extraction [28], quaternion Fourier-Mellin transforms (QFMT) [29], block truncation coding (BTC) [30], local linear embedding (LLE) with DCT [31], Canny operator with color vector angle [32], center-symmetric local binary patterns [33], projected gradient nonnegative matrix factorization (PGNMF), and ring partition [34]. However, all these techniques are unable to provide adequate performance in terms of tamper localization and content recovery. In 2017, Tang et al. [35] introduced multidimensional data scaling (MDS) in producing robust image hash for data analysis and object retrieval. In this same year, Karsh et al. [36] used the singular value

decomposition (SVD) method to find a low-rank matrix followed by discrete wavelet transform (DWT) to generate a robust image hash and failed to detect color forgery and is more sensitive to translation.

2.3. Statistical Feature-Based Hashing. Later in 2018, several new techniques based on image features for the hashing algorithm were developed, which include extraction of structural features from color images [37], dual-cross pattern-based textural features [38], progressive feature point selection [39], and a geometric correction-based technique using local and global features to counter the rotation scaling translation (RST) attacks [40]. However, these algorithms are unable to verify the validity of all types of images from all across the globe. Similarly, in 2019, Tang et al. [41] developed a new methodology on the basis of tensor decomposition. In that year, Qin et al. [42] integrated local texture and color angle characteristics in generating a robust image hash. However, these methods cannot be applied to video hashing. Recently, to improve the performance of image hashing, various researchers have proposed different techniques such as a Binary Multi-View Perceptual Hashing (BMVPH) [43], a Gray-level cooccurrence matrix-based hashing [44], Fourier-Mellin transform and fractal coding-based technique to create a fingerprint image [45], fractal image coding and ring partition-based hashing [46], quadtree structure and color opponent component- (COC-) based technique for forging detection and tampering localization [47], and a Laplacian pyramid-based hashing technique [48]. However, these algorithms fail to provide adequate performance against some attacks like rotation invariant. Additionally, these techniques are also unable to provide satisfactory performance in case of tamper localization.

Apart from the above categories, some other hashing algorithms have also been proposed. These are based on deep ordinal hashing [49], deep-network-based hashing [50], deep transfer networks (DTNs) [51, 52], image fusion [53, 54], etc. Some of the hashing algorithms [49, 50] have utilized the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique for visualization of the learned hash features. The t-SNE is a technique for dimensionality reduction that is particularly well suited for the exploration and visualization of high-dimensional data into low-dimensional space, and it finds the patterns in the data based on similarity of data points. Most of the above-stated works are concerned with making hash values more robust, and others concentrated on localizing the tampered areas. Our objective is to train a single-layered convolutional autoencoder to produce hash values resistant to various geometric attacks while detecting and localizing tampered regions.

2.4. The following Are the Contributions of the Proposed Algorithm

(1) Existing literature suggests that although most of the methods are robust to content preserving operations, however, the techniques are very sensitive to geometric operations. In this work, an autoencoder-based image hashing algorithm has been developed to overcome this problem

(2) The proposed image hashing algorithm is capable of detecting and localizing minor tampering portions in the images, unlike most of the existing algorithms

(3) Experimental results suggest that the presented algorithm is capable of proving improved performance irrespective of types of images from various databases. For instance, experiments were performed on CASIA Tampered image detection evaluation database [55], NITS Image hashing database [56], USC-SIPI Image database [57], and Ground Truth Database [58] for tampering detection and localization to check the ability of the proposed algorithm

(4) A comparative analysis with different state-of-the-art techniques suggests the competitiveness of the proposed algorithm. The performance parameters such as the area under the ROC curve (AUC), true-positive rate (TPR), and false-positive rate (FPR) [59–63] are utilized to evaluate the algorithms

The remaining part of the paper has been structured in the following manner. Section 2 discusses the relevant literature. Section 3 presents the method being developed, the model architecture, the proposed approach, and implementation details. The experimental results and discussions have been presented in Section 4. The comparison with existing works has been discussed in Section 5, while the whole work has been concluded in Section 6.

3. Proposed Model Architecture

The model is built on stacked convolutional autoencoders. In this model, the encoder network maps input images into latent space, and it is decoded and recreated into the original image. Our network of encoders includes five numbers of convolutional autoencoders followed by a fully connected network whose activations are subjected to L2 regularization. This is the layer generating the hash code. Each convolutional autoencoder comprises conv-relu-batch norm-max pooling in the encoder and upsampling-batch norm-conv in the decoder part. The architecture of the proposed model is shown in Figure 1, whereas the visualization of the learned hashing features with Gradcam and Gradcam++ technique of original host image is presented in Figure 2. Similarly, the visualization of the learned hashing features for a tampered image is also done and demonstrated in Figure 3. It is observed that the heatmap generated and Gradcam technique gave identical results. Hence, the effectiveness of the proposed method has been proved by the visualization of feature activation shown by Gradcam.

3.1. Convolutional Autoencoder. M convolutional layers make up the network, with a completely connected layer in the middle providing the hash code. Then, the output given by the encoder can be given by $x_i \in R^{W_i \mathrm{XH}_i \mathrm{XC}_i}$, where I denote the i^{th} level of the encoder network. Here, W and H indicate

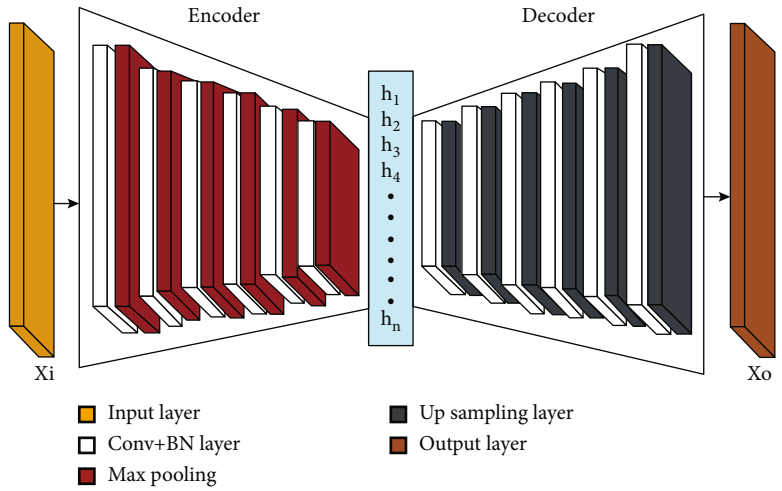
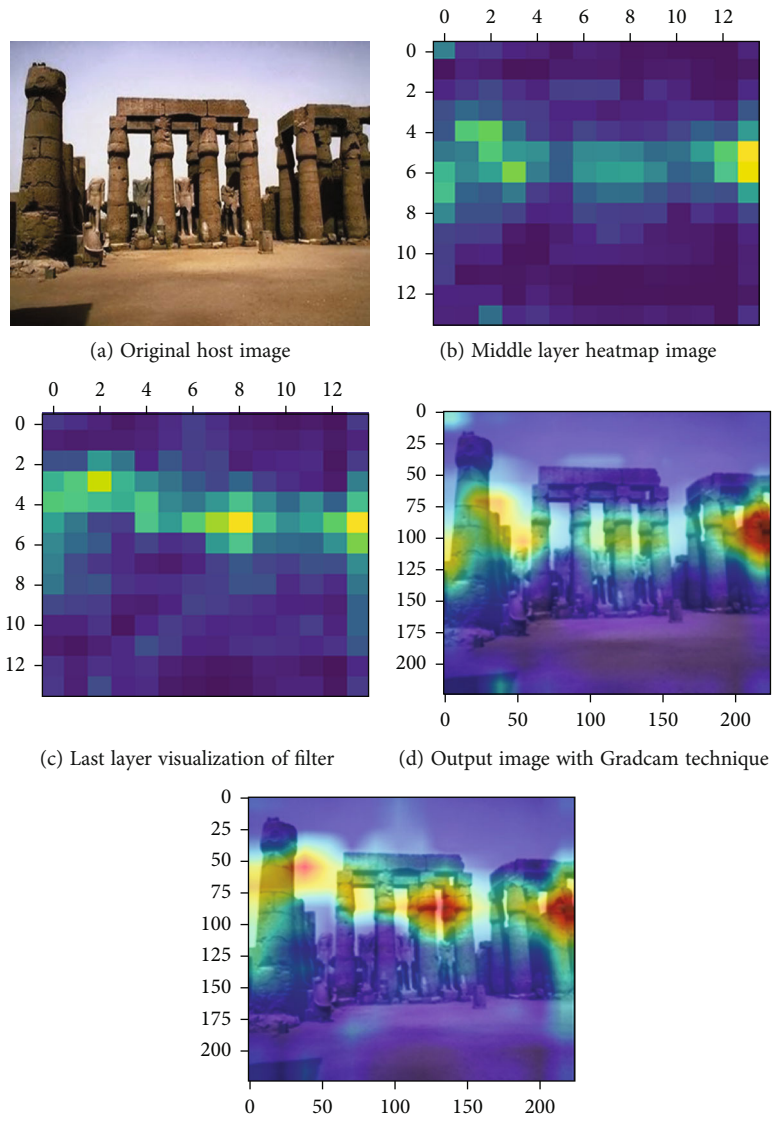


Figure 1: Architecture of the proposed model.



(a) Original host image

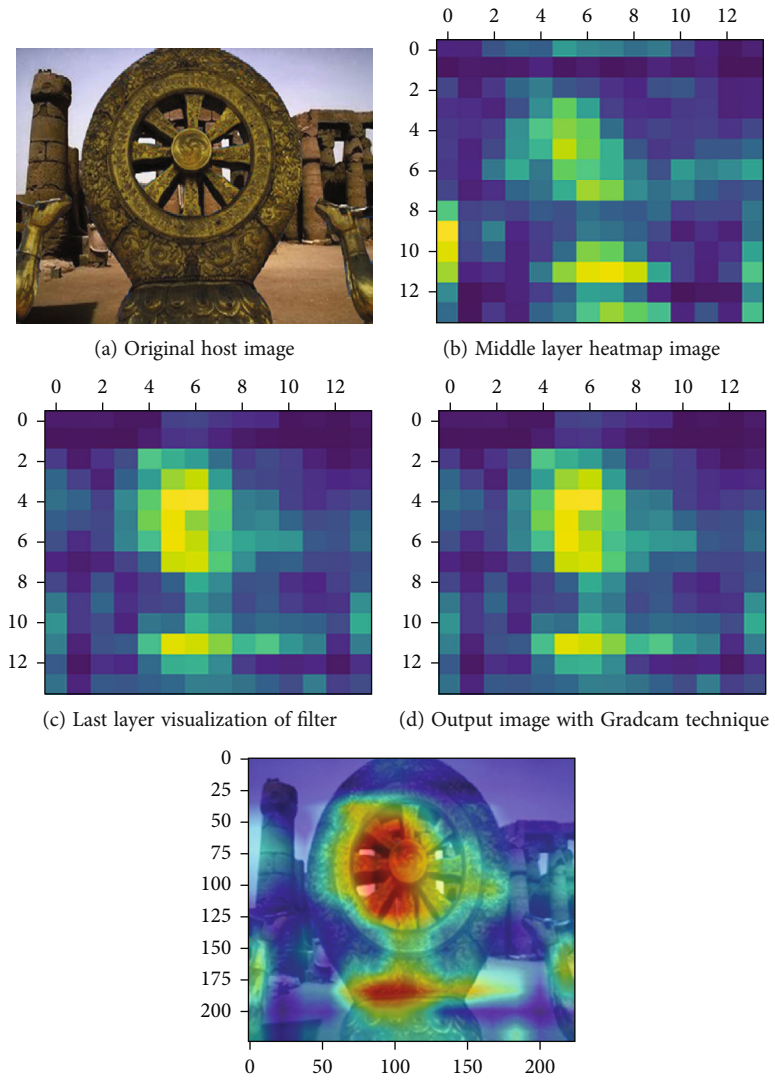
(b) Middle layer heatmap image

(c) Last layer visualization of filter

(d) Output image with Gradcam technique

(e) Output image with Gradcam++ technique

Figure 2: Visualization of the learned hashing features with Gradcam and Gradcam++ technique of original image.



(a) Original host image
(b) Middle layer heatmap image
(c) Last layer visualization of filter
(d) Output image with Gradcam technique
(e) Output image with Gradcam++ technique

Figure 3: Visualization of the learned hashing features with Gradcam and Gradcam++ technique of tampered image.

the image width and height, and C indicates the channel number. In the decoder part, the reconstructed $y_i \in R^{W}{}_{i}{}^{\mathrm{XH}}{}_{i}{}^{\mathrm{XC}}{}_{i}$ denotes the i^{th} level of the decoder. The input and the output images are denoted by x_0 and y_0. The activations of the K_{th} layer are given by max-pool $(\Omega(X_{k-1} * W + b))$ in the encoder and (upsample $(Y_{k-1}) * W^{'} + b^{'}$) in the decoder part. Here, upsampling refers to the bilinear interpolation done to the image, after which convolution is done in the decoder part. (W, b) and $(W^{'}, b^{'})$ refer to the weights and biases of the convolution layer in the encoder and the decoder network, respectively. The activation function used in our case is ReLu (rectified linear unit). This is used everywhere in the middle of the network except in the fully connected middle layer, where the sigmoid activation function is used. The max-pool refers to the max pooling operation with a stride of 2. The parameters of the model are learned by using the Adagrad optimizer. The model learns by minimizing the mean square error among the input x_0 and the output y_0:

$$\text{minimize } l_1 = \frac{1}{N} \sum_{\substack{j=1 \\ W,b}}^{N} \left\| x_0^j - y_0^j \right\|^2, \tag{1}$$

where N is the total number of samples in the dataset.

3.2. Fully Connected Autoencoder. Both the encoder and the decoder network, a completely connected autoencoder, is placed to reduce the size of the hash code much more. It consists of three fully linked layers. The coding for the images is provided by the hidden layer. We test both a totally stacked and a fully linked autoencoder. It is also simpler to train and refine the completely convolutional encoder. Because of weight distribution in the convolutional layers, it is, therefore, less capable of approximating. The same MSE loss Equation (1) is used to train completely convolutional and convolutional plus fully connected layers.

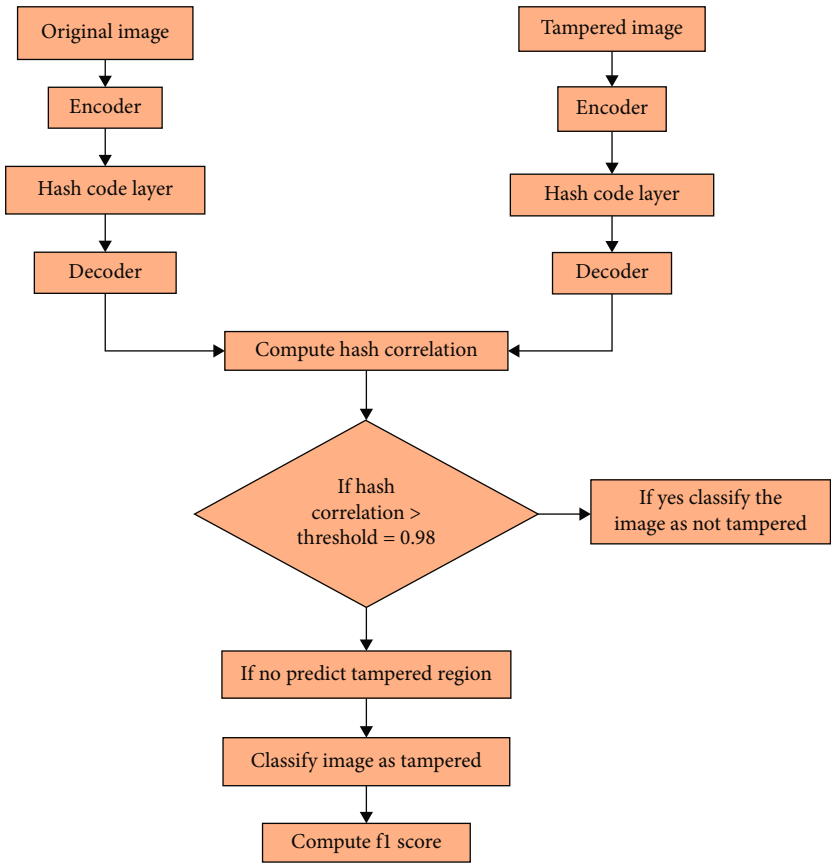


Figure 4: Flowchart for the entire process.

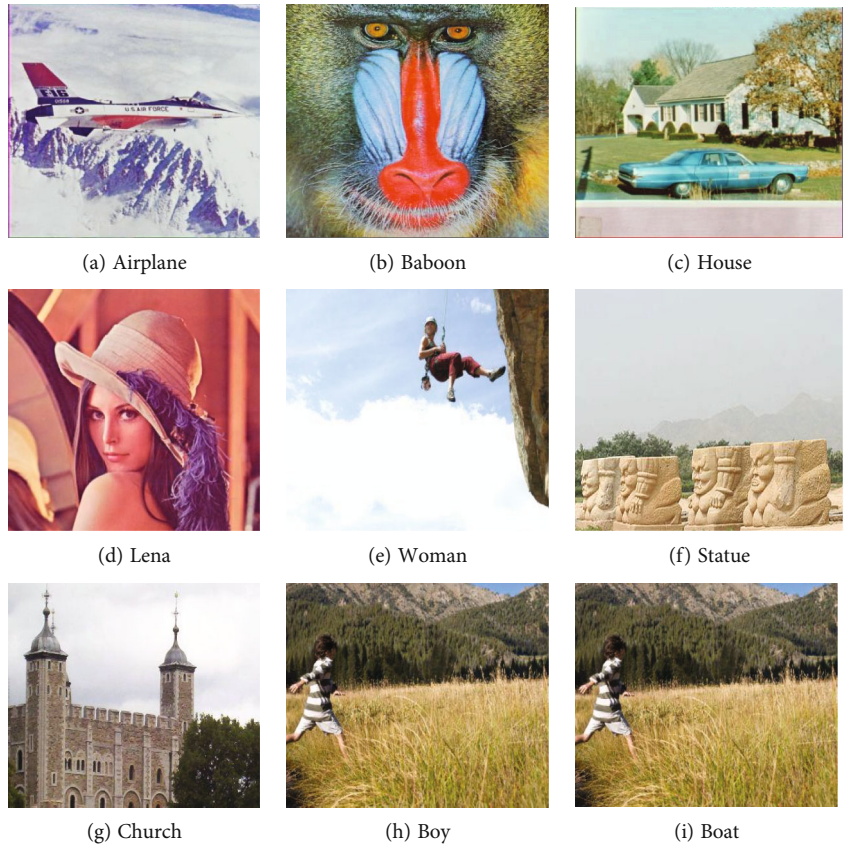
Table 1: TPR and FPR scores for "operations Indonesia, Italy, Japan ".

Operation	True-positive score (Indonesia)	False-positive score (Indonesia)	True-positive score (Italy)	False-positive score (Italy)	True-positive score (Japan)	False-positive score (Japan)
Brightness	0.916	0.083	0.863	0.136	0.867	0.133
Compression	1	0	1	0	1	0
Gamma	0.875	0.125	0.93	0.068	0.9	0.099
Rotation	0.177	0.823	0.1704	0.829	NA	NA
Speckle	1	0	1	0	1	0
Watermark	0.99	0.009	0.936	0.045	1	0
Gaussian	0.993	0.006	1	0	1	0
Scaling	0.99	0.009	0.954	0.0454	1	1

3.3. Approach. The convolutional layer is trained on input images having dimensions of (128, 128, 3). The model is trained on the USC-SIPI dataset [57]. The model is trained on 40000 input images. Noisy images are given to the model as input, and the corresponding denoised images are given as output to train the model. The noisy images in this context refer to the original images which have undergone any one of the following operations, viz., changes in brightness, contrast, gamma correction, adding of Gaussian, salt and pepper, and speckle noise, image scaling, rotation, and compression. The input images are passed through 5 convolutional units, which make the encoder network. Each of these convolutional units comprises a convolutional layer having 16 filters, a batch-normalization layer, which is accompanied by a max-pooling layer with a filter scale of (3, 3) and a stride of 2. This is passed to a fully connected autoencoder which generates the hash code of (1, 48) dimensions. The activations of the middle layer of this fully connected autoencoder are subjected to L2 regularization. This is then passed to a decoder network that attempts to reconstruct the image. The decoder has five convolutional units, each of which comprises an upsampling layer [60]. Each of

Table 2: Operations that preserve content for various parameters.

Software	Manipulation	Parameter	Parameter values	Image pairs
Stir Mark	JPEG compression	Quality factor	30, 40, 50, 60, 70, 80, 90, 100	10
Stir Mark	Watermark embedding	Strength	10, 20, 30, 40, 50, 60, 70, 80, 90, 100	12
Stir Mark	Scaling	Ratio	0.5, 0.75, 0.9, 1.1, 1.5, 2.0	8
Stir Mark	Rotation	Rotation angle in degree	±5, ±15, ±30, ±45, ±90	12
Photoshop	Brightness adjustment	Photoshop scale	±10, ±20	8
Photoshop	Contrast adjustment	Photoshop scale	±10, ±20	8
Matlab	Gamma correction	γ^1	0.75, 0.9, 1.1, 1.5	6
Matlab	3×3 Gaussian low pass filter	Standard deviation	0.4–1 (with step size 0.1)	10
Matlab	Salt and pepper noise	Density	0.001–0.01 (with step size 0.001)	12
Matlab	Speckle noise	Variance	0.001–0.01 (with step size 0.001)	12
Total			74 parameter values	98



(a) Airplane (b) Baboon (c) House

(d) Lena (e) Woman (f) Statue

(g) Church (h) Boy (i) Boat

Figure 5: Standard images used for testing.

these units uses ReLu as an activation function. The model finally learns by optimizing the following.

$$\text{minimize } l_1 = \frac{1}{N} \sum_{\substack{j=1 \\ \mathbf{W},\mathbf{b}}}^{N} \left\| x_0^j - y_0^j \right\|^2 + \frac{\alpha}{N} \sum_{j=1}^{N} \left\| h_j \right\|_2^2, \tag{2}$$

where α is the regularization coefficient and $\|h_j\|_2^2$ is the squared L2 norm of h_j which are the activations of the hash code layer for the jth image. For our experiments, we take the hash code layer to have 48 dimensions.

3.4. Implementation Details. To construct the model, we used the Keras library with a Tensorflow backend. The model is trained using the online Google-Colab deep learning platform and uses Tesla K80GPU for training. Each of the weights of the convolutional layer is taken from a glorot-uniform distribution which is the default setting for Keras. The bias for each of these layers is set to 0. We train

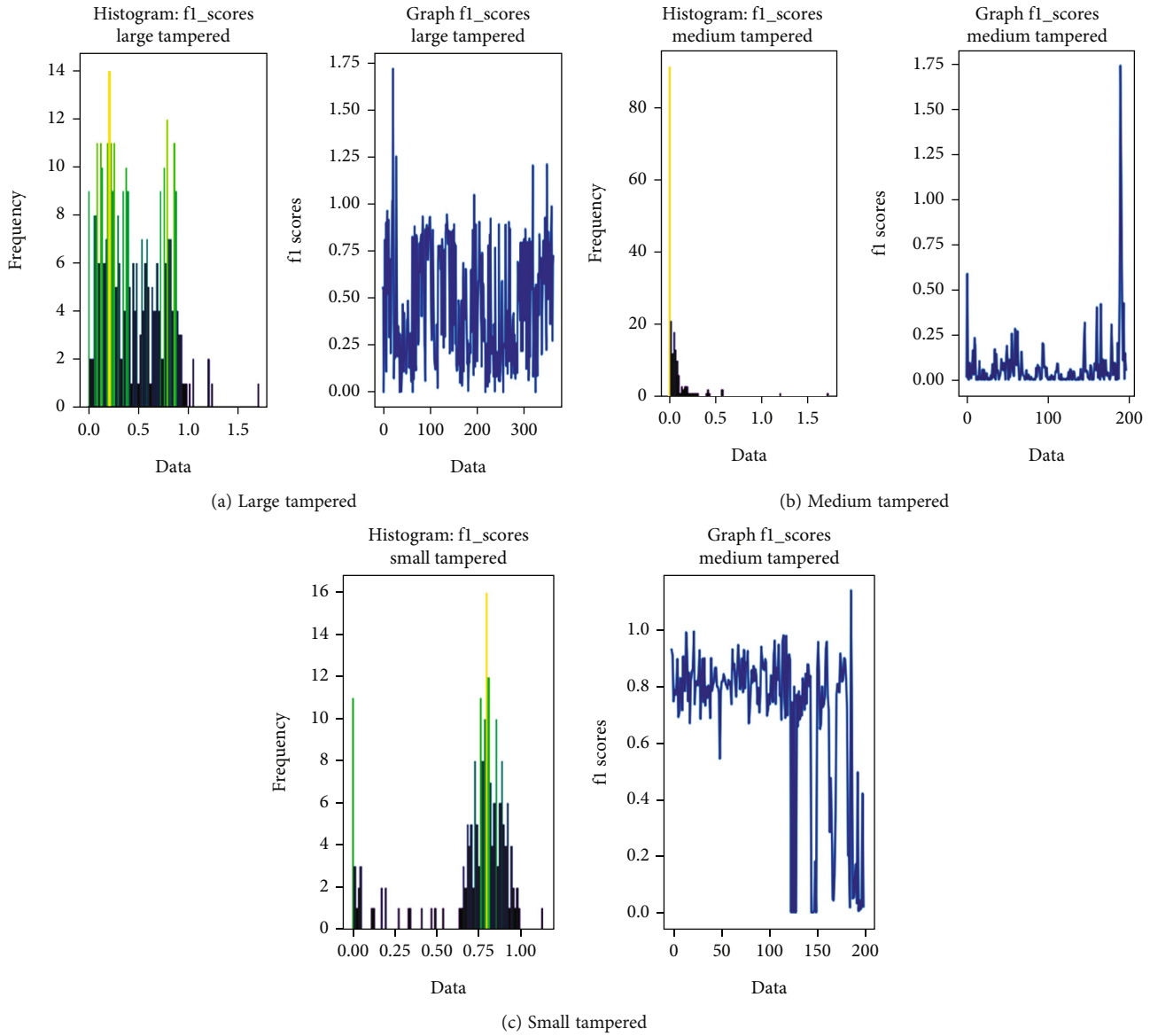
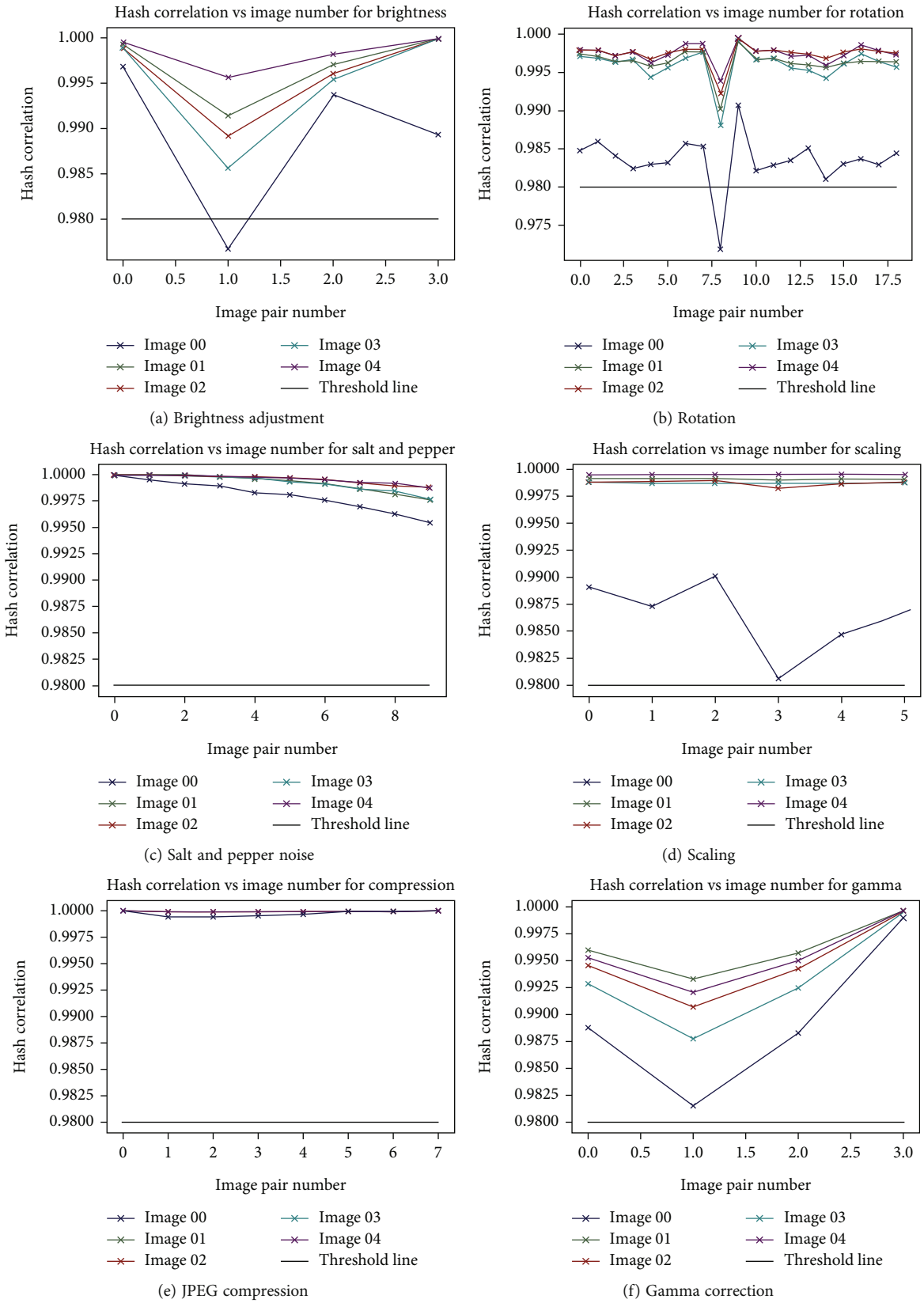


FIGURE 6: $f1$ scores for different tampered images.

a model using input images with sizes of $512 \times 512 \times 3$ and $128 \times 128 \times 3$. The model trained on $512 \times 512 \times 3$ image input has a faster convergence time and appears to have oscillatory precision and loss curves, while the model trained on $128 \times 128 \times 3$ image input appears to have comparatively smooth curves. We used Adagrad for the optimization of Equation (2).

The aforementioned dataset [58] comprises the following: "scenery_bmp," "animals," "aerials," "Indonesia," "Japan," "Italy," "operations Italy," "operations animals," "operations Indonesia," "operations scenery." The folders having the name "operations" as their prefix contain the results of the different content preserving operations done on the original image. For example, the "animals" folder comprises the original images, while the "operations-animals" folder contains the tampered images produced by the different content preserving operations for each image in the "animals" folder. The first three "animals," "scenery_bmp," and "aerials" and their corresponding "operations_" folders are used for training the model, while the rest of the dataset is used for testing the robustness of the model. The model is tested on 18334 images from "operations Italy," "operations Japan," and "operations Indonesia."

The robustness of the hash codes produced is tested on the basis of the two metrics "true-positive rate" (TPR) and "false-positive rate" (FPR) scores which are defined below. We take the TPR and FPR scores for different operations on the image for the three "operations folders," i.e., "Indonesia," "Italy," and Japan. A typical value of 0.98 is taken as the threshold for hash correlation. During the testing phase, both the original image and the tampered image are passed through the model, which produces their respective hash codes. If the correlation between the hash codes is less than that of the threshold, the tampered image is classified as



(a) Brightness adjustment

(b) Rotation

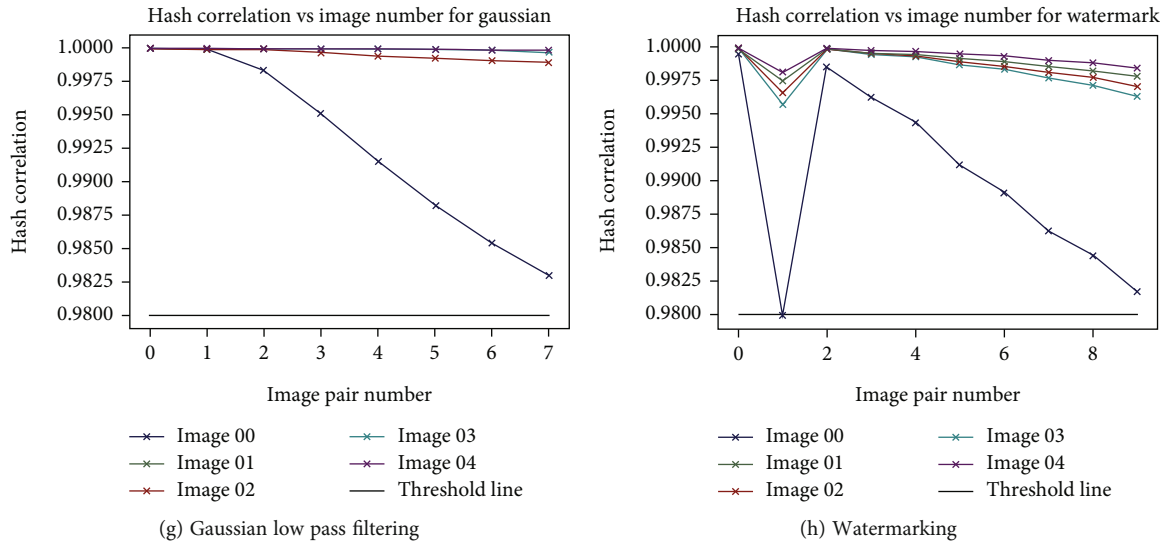
(c) Salt and pepper noise

(d) Scaling

(e) JPEG compression

(f) Gamma correction

Figure 7: Continued.



(g) Gaussian low pass filtering (h) Watermarking

Figure 7: Robustness test based on the standard images.

being "false-positive" or otherwise "true-positive." This process is repeated for all the images and for all the aforementioned operations.

The model is then tested for its tampering localization ability.

$$\text{PTPR} = \frac{n_{\text{same}}}{N_{\text{Similar}}}, \tag{3}$$

$$\text{PFPR} = \frac{n_{\text{different}}}{N_{\text{different}}}. \tag{4}$$

In Equations (3) and (4), n_{same} and $n_{\text{different}}$ are same image pairs that are detected and different image pairs that are mistakenly detected; N_{similar} and $N_{\text{different}}$ are total pairs of similar images and total pairs of different images.

During the tampering-localization test, the model receives the tampered image and the hash code of the original image as the input. This hash code was generated when the corresponding original image was put in through the encoder network of the model. The tampered image is passed through the encoder and then the decoder network to give the reconstructed tampered image n_1. The hash code of the original image is passed through the decoder to produce n_2. Then, the tampered region is in

$$R(s) = \{(i,j)\text{such that}|n_1(i,j) - n_2(i,j)| < 0.5\}. \tag{5}$$

The predicted tampered region R is compared with the actual tampered region T using the $f1$ score in

$$f1(R,T) = \frac{n(R \cap T)}{n(T)}. \tag{6}$$

Here, $n(R \cap T)$ refers to the total number of tampered pixels that are detected by the model, and $n(T)$ refers to the total number of tampered pixels in the image. T is the set containing the indexes of all the tampered pixels corresponding to that image. The entire process is represented in Figure 4.

4. Experimental Results

In the following subsections, the training procedure and results have been presented. We also compare the localization ability of the proposed model by comparing the $f1$-scores of the proposed model against the cutting-edge methods.

4.1. Training. The autoencoder is trained in a layer-wise fashion. Here, we freeze the weights of all convolutional units except a single one and train the model for 50 epochs. This entire process is repeated for all the convolutional units in both the encoder and the decoder network before being fine-tuned as a whole. The model is trained during the tuning process in 100 epochs with a small size of 400 images each of $128 \times 128 \times 3$ size. The λ coefficient of L2 regularization is kept to 0.01 for the entire duration of the training.

4.2. Hash Robustness Test. We test the robustness of the hash codes produced by the encoder on the "operations Indonesia," "operations Italy," and "operations Japan" folders of our custom dataset [58]. The true-positive rate and the false-positive scores are shown in Table 1. The true-positive rate refers to the fraction of the total testing images which are untampered and are classified to the untampered by our model. Similarly, the false-positive rate refers to the untampered images which are classified to be tampered by our model. In this test, the hash correlation threshold is taken as 0.98. Content preserving operations for various parameters are shown in Table 2. Few standard images used for the robustness test are shown in Figure 5.

From Table 2, we should conclude that the proposed model is completely immune to all the operations listed except "rotation operations" on the image, which shows a lower value of true positive rate in comparison to all the other operations. The hash correlation for different

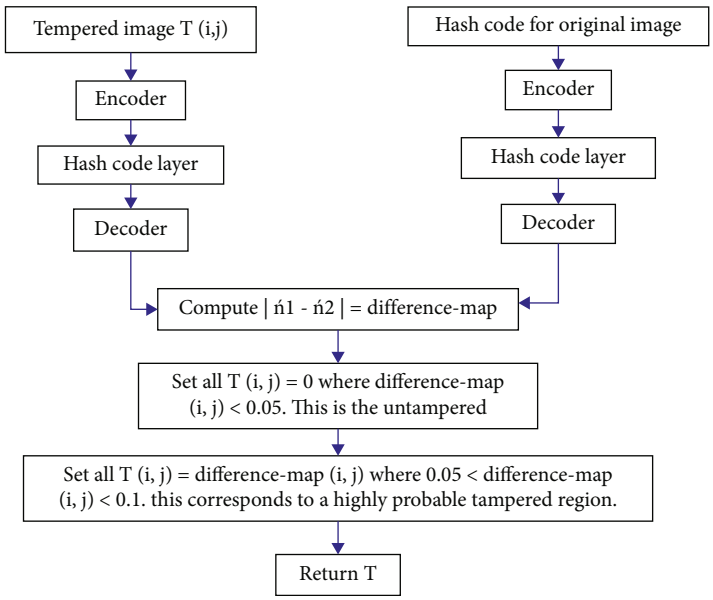
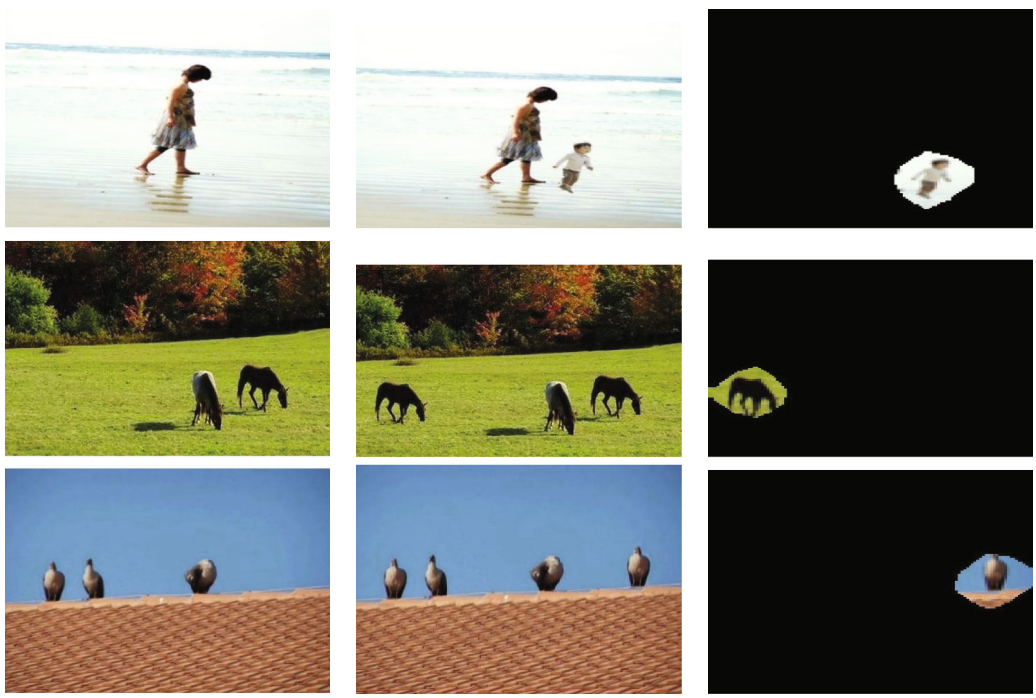


Figure 8: Flowchart for tampering localization test.



(a) Original images (b) Tampered images (c) Tampered Localization

Figure 9: Localization of tampered regions.

operations corresponding to the three testing folders is shown in Figure 6. The robustness test based on standard images is shown in Figure 7.

4.3. Localization Capability Test. The tampering ability is tested on 763 tampered images [55]. These images are categorized into 3 "large-tampered," "medium-tampered," and "small-tampered" folders, depending on the degree of tampering that the image has undergone. There are 365 large-tampered images, 198 medium-tampered images, and 200 small-tampered images. The tampering localization test process can be summed up in the flowchart shown in Figure 8. Each sample comprises three images, the first being the original image, the second being a tampered version of that

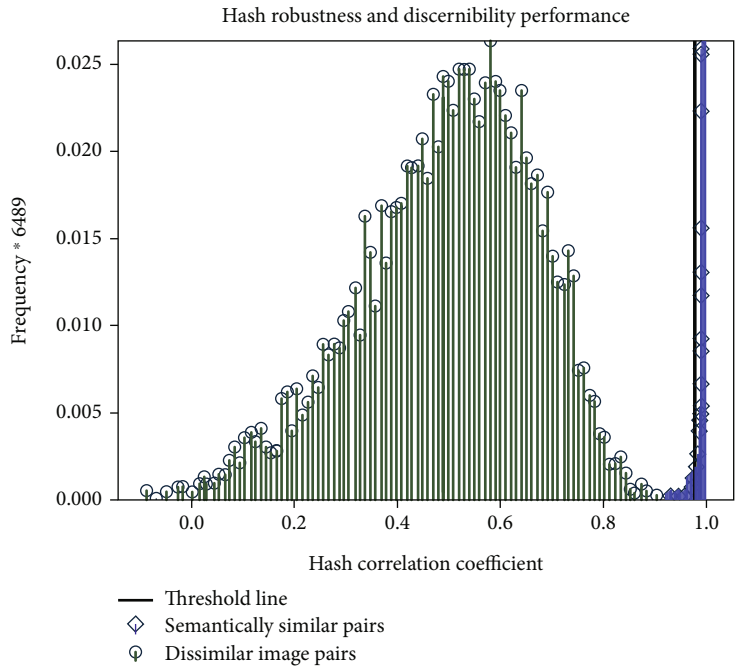


Figure 10: Discernibility or discrimination test based on 200 different images.

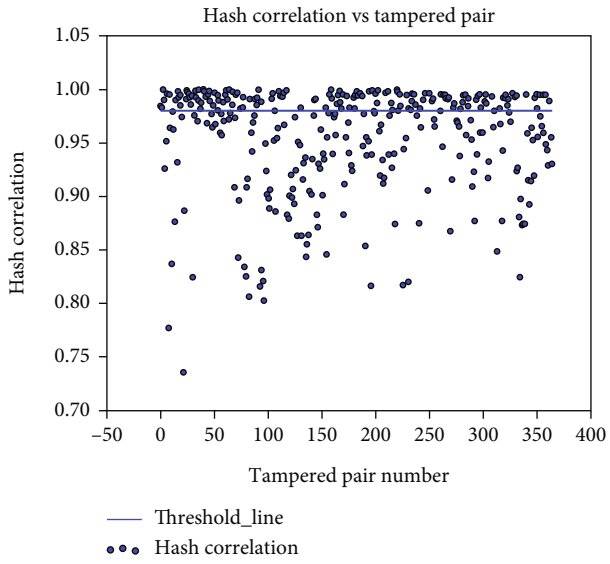


Figure 11: Hash correlation vs. tempered image pairs on 200 different images.

original image, and the third being the difference between the original image and the tampered image. Figure 9 shows some samples where our model has been successfully able to detect all the tampered regions in an image.

4.4. Discernibility Test. Here, the proposed model is tested for its ability to distinguish between two semantically dissimilar images. We took 200 different images [56] and made combinations of 2 images each, thereby making $^{200}C_2$ samples. We tested the discerning capability of the model by computing the true-positive rate scores and the false-positive rate scores. The TPR shows the percentage of the total samples which the model classifies to be different. If the hash correlation between the hash codes produced by the encoder network given the two different images of a sample is greater than 0.98, then the model classifies the two images to be semantically same, else different. The histograms of the hash correlation of different image samples are shown in Figure 10. Hash correlation between different tampered image pairs is shown in Figure 11.

Table 3: A comparison of the performance of various hashing methods.

Reviewed techniques	L. Chen et al. [41]	X. Zhang et al. [19]	R. K. Karsh et al. [40]	Q. Shen et al. [64]	Z. Tang et al. [27]	R. K. Karsh et al. [36]	C. Qin et al. [42]	H. Hamid et al., [48]	Z. Tang et al. [35]	C. Qin et al. [30]	Proposed
Execution time (s)	0.17	NA	NA	NA	0.61	NA	0.112	NA	0.22	1.1	0.6
AUC	0.9993	NA	0.9914	NA	NA	NA	0.9601	0.9264	0.9909	NA	0.999
Average time (s)	0.3021	0.0164	0.3425	0.0617	0.2863	2.1	0.2213	0.6651	0.9966	0.8164	0.2234
Optimal TPR when FPR = 0	0.9827	0.9998	NA	0.998	0.953	0.9823	0.991	NA	NA	NA	0.8162
Optimal FPR when TPR = 1	9.4×10^{-3}	4.13×10^{-6}	NA	2.22×10^{-5}	0.4749	0.0051	0.1273	NA	NA	NA	0.0621

Table 4: The main algorithms of the compared methods.

Authors	Main algorithms used
Z. Tang et al. [35]	Multidimensional scaling (MDS)
L. Chen et al. [41]	Tensor decomposition (TD)
R. K. Karsh et al. [40]	Geometric correction
Z. Tang et al. [27]	Ring partition and invariant vector distance (RP-IVD)
C. Qin et al. [30]	Block truncation coding (BTC)
Q. Shen et al. [64]	Color opponent component and quadtree structure
C. Qin et al. [42]	Weber local binary pattern and color angle representation
R. K. Karsh et al. [36]	DWT-SVD and spectral residual
X. Zhang et al. [19]	Three-dimensional color structure features and luminance gradient
H. Hamid et al., [48]	Laplacian pyramids

5. Comparison with Existing Work

This section compares the proposed algorithm results against classical perceptual hash-producing algorithms. In Table 3, Table 4, Table 5, and Table 6, the proposed approach is compared to some of the existing perceptual image hashing algorithms. It is observed from the table that the hash codes produced by the proposed model are robust against scaling, rotation, watermarking, jpeg compression, gamma correction, and Gaussian low pass filter. Table 1 shows true-positive rates for different degrees of rotation. The true-positive rates here refer to the total untampered, rotated images that the model classifies to be the same (i.e., the hash correlation between the hashes of the original and the rotated image is less than the threshold). From Table 1, it becomes evident that the robustness of the hash codes produced by the proposed approach is competitive with those produced by the state-of-the-art approaches. Despite the fact that we are not using any previously annotated data during training, the proposed model performance is better or comparable to the other conventional perceptual hashing algorithms in execution time, area under ROC curve (AUC), and average time. The ROC curve in Figure 12 consists of several points (TPR, FPR) with different thresholds and is used to assess the equilibrium between robustness and discrimination.

Table 5: Hash length comparison of various hashing algorithms.

Authors	Hash length
Z. Huang et al. [44]	720 bits
C. M. Pun et al. [39]	262 digits
C. P. Yan et al. [28]	302 bits
Z. Tang et al. [20]	206 bits
F. Khelaifi et al. [46]	100 digits
H. Lao et al. [31]	316 bits
Q. Shen et al. [64]	452 digits
L. Du et al. [43]	512 bits
Proposed method	64 bits

Table 6: Geometric attack comparison of various hashing algorithms.

Name of the attack	Parameter	SIFT-SVD [65]	SIFT-DWT [66]	Proposed method
JPEG compression	Quality factor	20–90 with step size 10	30 : 10 : 100	30–100 with step size 10
Watermark embedding	Strength	30–80 with step size 10	30 : 10 : 90	10–100 with step size 10
Scaling	Ratio	2–7 with step size 1	0.5 : 0.1 : 1.5	0.5–2.0 with step size 0.25
Rotation	Rotation angle in degree	±5 to ±45	±2, ±5, ±10, ±5,…. ±60	±5, ±15, ±30, ±45, ±90
Brightness adjustment	Photoshop scale	±10, ±30	−20 : 10 : 20	±10, ±20
Contrast adjustment	Photoshop scale	±10, ±30	−20 : 10 : 20	±10, ±20
Gamma correction	Gamma	0.6-1.5 with step size 0.1	0.85, 0.95, 1.05, 1.15, 1.25	0.75, 0.9, 1.1, 1.5
3×3 Gaussian low pass filter	Standard deviation	0.6–1 (with step size 0.1)	0.3–1 (with step size 0.1)	0.4–1 (with step size 0.1)
Gaussian noise	Variance	0.3–1 with step size 0.1	0.2–1 with step size 0.1	0.1–1 with step size 0.1
Median filter	Filter size	2–7 with step size 1	1–8 with step size 1	1–9 with step size 1

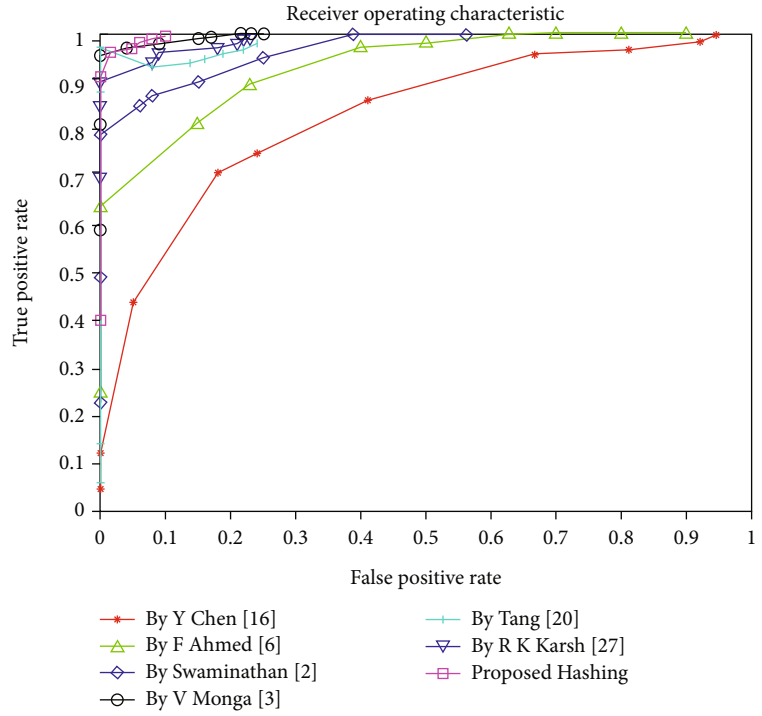


Figure 12: ROC comparison of proposed approach with several state-of-the-art methods for robustness and discrimination trade-off.

6. Conclusions and Future Scope

In this paper, a stacked convolutional autoencoder with an L2 regularization has been proposed to produce hash values. The hash values are not only robust against the enhancement and geometric attacks but would also be robust against various content preserving operations like image compression, the addition of Gaussian, speckle noise, scaling, and watermarking. The convolutional units help us to learn high-level semantic information from the data manifolds. The investigative conclusion on the massive pairs of images indicates that the system can detect and locate minor tampering in the images. Moreover, it offers a more favorable contrast between FPR and TPR. After training only for 100 epochs, the proposed model shows a competitive performance with those of the state-of-the-art approaches in both the hash robustness test and tampering localization test. It may even localize tampering, in spite of tampering and image rotation taking place at the same time, which is a significant drawback of current approaches.

The future scope of this work will consist of pretraining the network's weight matrices with stochastic and generative neural networks such as Boltzmann distribution to accomplish quicker convergence, which helps to reduce mean square error loss. The robustness of CPOs can also be examined in Dense Nets. This method will be devised in order to shorten the hash code while maintaining machine performance.

Data Availability

The processed data are available upon request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to acknowledge all the research scholars belonging to the Image Processing Lab of ECE Department, NIT Silchar, Assam, India, and all the members of the R&D Section, CMR College of Engineering & Technology, Hyderabad, India, for providing valuable suggestions and essential facilities in completing this work.

References

[1] 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.

[2] 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.

[3] V. Monga and M. K. Mihçak, "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] D. Wu, X. Zhou, and X. Niu, "A novel image hash algorithm resistant to print-scan," *Signal Processing*, vol. 89, no. 12, pp. 2415–2424, 2009.

[5] F. Khelifi and J. Jiang, "Perceptual image hashing based on virtual watermark detection," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 981–994, 2010.

[6] M. Islam, G. Mallikharjunudu, A. S. Parmar, A. Kumar, and R. H. Laskar, "SVM regression based robust image watermarking technique in joint DWT-DCT domain," in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pp. 1426–1433, Kerala, India, 2017.

[7] M. Samiullah, W. Aslam, M. A. Khan et al., "Rating of modern color image cryptography: a next-generation computing perspective," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 7277992, 20 pages, 2022.

[8] U. Zia, M. McCartney, B. Scotney et al., "Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains," *International Journal of Information Security*, vol. 21, no. 4, pp. 917–935, 2022.

[9] D. S. Laiphrakpam and M. S. Khumanthem, "A robust image encryption scheme based on chaotic system and elliptic curve over finite field," *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 8629–8652, 2018.

[10] K. M. Singh, L. D. Singh, and T. Tuithung, "Improvement of image transmission using chaotic system and elliptic curve cryptography," *Multimedia Tools and Applications*, vol. 1-22, pp. 1–22, 2022.

[11] Y. Lei, Y. Wang, and J. Huang, "Robust image hash in Radon transform domain for authentication," *Signal Processing: Image Communication*, vol. 26, no. 6, pp. 280–288, 2011.

[12] Z. Tang, S. Wang, X. Zhang, W. Wei, and Y. Zhao, "Lexicographical framework for image hashing with implementation based on DCT and NMF," *Multimedia Tools and Applications*, vol. 52, no. 2–3, pp. 325–345, 2011.

[13] Y. S. Choi and J. H. Park, "Image hash generation method using hierarchical histogram," *Multimedia Tools and Applications*, vol. 61, no. 1, pp. 181–194, 2012.

[14] Y. Li, Z. Lu, C. Zhu, and X. Niu, "Robust image hashing based on random Gabor filtering and dithered lattice vector quantization," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1963–1980, 2012.

[15] S. S. Skandha, M. Agarwal, K. Utkarsh, S. K. Gupta, V. K. Koppula, and J. S. Suri, "A novel genetic algorithm-based approach for compression and acceleration of deep learning convolution neural network: an application in computer tomography lung cancer data," *Neural Computing and Applications*, pp. 1–23, 2022.

[16] L. Saba, S. S. Sanagala, S. K. Gupta et al., "A multicenter study on carotid ultrasound plaque tissue characterization and classification using six deep artificial intelligence models: a stroke application," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

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

[18] X. Lv and Z. Jane Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1081–1093, 2012.

[19] 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.

[20] Z. Tang, X. Zhang, and S. Zhang, "Robust perceptual image hashing based on ring partition and NMF," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 711–724, 2014.

[21] Z. Tang, F. Yang, L. Huang, and X. Zhang, "Robust image hashing with dominant DCT coefficients," *Optik*, vol. 125, no. 18, pp. 5102–5107, 2014.

[22] R. Sun and W. Zeng, "Secure and robust image hashing via compressive sensing," *Multimedia Tools and Applications*, vol. 70, no. 3, pp. 1651–1665, 2014.

[23] Y. Chen, W. Yu, and J. Feng, "Robust image hashing using invariants of Tchebichef moments," *Optik*, vol. 125, no. 19, pp. 5582–5587, 2014.

[24] Z. Tang, L. Ruan, C. Qin, X. Zhang, and C. Yu, "Robust image hashing with embedding vector variance of LLE," *Digital Signal Processing*, vol. 43, pp. 17–27, 2015.

[25] L. S. Sebastian, A. Varghese, and T. Manesh, "Image authentication by content preserving robust image hashing using local and global features," *Procedia Computer Science*, vol. 46, pp. 1554–1560, 2015.

[26] 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.

[27] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE*

Transactions on Information Forensics and Security, vol. 11, no. 1, pp. 200–214, 2016.

[28] C. P. Yan, C. M. Pun, and X. C. Yuan, "Multi-scale image hashing using adaptive local feature extraction for robust tampering detection," *Signal Processing*, vol. 121, pp. 1–16, 2016.

[29] C. P. Yan, C. M. Pun, and X. C. Yuan, "Quaternion-based image hashing for adaptive tampering localization," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2664–2677, 2016.

[30] C. Qin, X. Chen, D. Ye, J. Wang, and X. Sun, "A novel image hashing scheme with perceptual robustness using block truncation coding," *Information Sciences*, vol. 361-362, pp. 84–99, 2016.

[31] Z. Tang, H. Lao, X. Zhang, and K. Liu, "Robust image hashing via DCT and LLE," *Computers & Security*, vol. 62, pp. 133–148, 2016.

[32] Z. Tang, L. Huang, X. Zhang, and H. Lao, "Robust image hashing based on color vector angle and Canny operator," *AEU-International Journal of Electronics and Communications*, vol. 70, no. 6, pp. 833–841, 2016.

[33] R. Davarzani, S. Mozaffari, and K. Yaghmaie, "Perceptual image hashing using center-symmetric local binary patterns," *Multimedia Tools and Applications*, vol. 75, no. 8, pp. 4639–4667, 2016.

[34] R. K. Karsh, R. H. Laskar, and B. B. Richhariya, "Robust image hashing using ring partition-PGNMF and local features," *Springerplus*, vol. 5, no. 1, p. 1995, 2016.

[35] Z. Tang, Z. Huang, X. Zhang, and H. Lao, "Robust image hashing with multidimensional scaling," *Signal Processing*, vol. 137, pp. 240–250, 2017.

[36] R. K. Karsh, R. H. Laskar, and Aditi, "Robust image hashing through DWT-SVD and spectral residual method," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, Article ID 31, 2017.

[37] C. Qin, M. Sun, and C. C. Chang, "Perceptual hashing for color images based on hybrid extraction of structural features," *Signal Processing*, vol. 142, pp. 194–205, 2018.

[38] C. Qin, X. Chen, X. Luo, X. Zhang, and X. Sun, "Perceptual image hashing via dual-cross pattern encoding and salient structure detection," *Information Sciences*, vol. 423, pp. 284–302, 2018.

[39] C. M. Pun, C. P. Yan, and X. C. Yuan, "Robust image hashing using progressive feature selection for tampering detection," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 11609–11633, 2018.

[40] R. K. Karsh, A. Saikia, and R. H. Laskar, "Image authentication based on robust image hashing with geometric correction," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25409–25429, 2018.

[41] Z. Tang, L. Chen, X. Zhang, and S. Zhang, "Robust image hashing with tensor decomposition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 549–560, 2019.

[42] C. Qin, Y. Hu, H. Yao, X. Duan, and L. Gao, "Perceptual image hashing based on weber local binary pattern and color angle representation," *IEEE Access*, vol. 7, no. 1, pp. 45460–45471, 2019.

[43] L. Du, Z. Chen, and A. T. S. Ho, "Binary multi-view perceptual hashing for image authentication," *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 22927–22949, 2021.

[44] Z. Huang and S. Liu, "Perceptual image hashing with texture and invariant vector distance for copy detection," *IEEE Transactions on Multimedia*, vol. 23, pp. 1516–1529, 2021.

[45] S. M. Abdullahi, H. Wang, and T. Li, "Fractal coding-based robust and alignment-free fingerprint image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2587–2601, 2020.

[46] F. Khelaifi and H. J. He, "Perceptual image hashing based on structural fractal features of image coding and ring partition," *Multimedia Tools and Applications*, vol. 79, no. 27-28, pp. 19025–19044, 2020.

[47] X. Wang, X. Zhou, Q. Zhang, B. Xu, and J. Xue, "Image alignment based perceptual image hash for content authentication," *Signal Processing: Image Communication*, vol. 80, article 115642, 2020.

[48] H. Hamid, F. Ahmed, and J. Ahmad, "Robust image hashing scheme using Laplacian pyramids," *Computers and Electrical Engineering*, vol. 84, article 106648, 2020.

[49] L. Jin, X. Shu, K. Li, Z. Li, G. J. Qi, and J. Tang, "Deep ordinal hashing with spatial attention," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2173–2186, 2019.

[50] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan, "Instance-aware hashing for multi-label image retrieval," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2469–2479, 2016.

[51] X. Shu, G. J. Qi, J. Tang, and J. Wang, "Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation," in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 35–44, Brisbane, Australia, 2015.

[52] J. Tang, X. Shu, Z. Li, G. J. Qi, and J. Wang, "Generalized deep transfer networks for knowledge propagation in heterogeneous domains," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 4s, pp. 1–22, 2016.

[53] X. Shu, J. Yang, R. Yan, and Y. Song, "Expansion-squeeze-excitation fusion network for elderly activity recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5281–5292, 2022.

[54] C. P. Yan and C. M. Pun, "Multi-scale difference map fusion for tamper localization using binary ranking hashing," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2144–2158, 2017.

[55] CASIA, *Tampered Image Detection Evaluation Database*http://forensics.idealtest.org/.

[56] NITS, *Image Hashing Database*, 2017, https://rishabhphukan.wixsite.com/rkkarsh.

[57] *USC-SIPI Image Database*, 2007, https://sipi.usc.edu/database/.

[58] *Ground Truth Database*http://imagedatabase.cs.washington.edu/groundtruth/.

[59] S. Debnath, F. A. Talukdar, and M. Islam, "Combination of contrast-enhanced fuzzy c - means (CEFCM) clustering and pixel based voxel mapping technique (PBVMT) for three-dimensional brain tumor detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2421–2433, 2021.

[60] M. Paul, A. J. Thakuria, R. K. Karsh, and F. A. Talukdar, "Robust color image hashing using convolutional stacked denoising auto-encoders for image authentication," *Neural Computing and Applications*, vol. 33, no. 20, pp. 13317–13331, 2021.

[61] A. S. Shaik, R. K. Karsh, M. Islam, and R. H. Laskar, "A review of hashing based image authentication techniques," *Multimedia Tools and Applications*, vol. 81, no. 2, pp. 2489–2516, 2022.

[62] A. S. Shaik, R. K. Karsh, M. Suresh, and V. Gunjan, "LWT-DCT based image hashing for tampering localization via blind geometric correction," in *ICDSMLA 2022*Springer, Singapore.

[63] A. S. Shaik, R. K. Karsh, and M. Islam, "Robust image hashing using chromatic channel," in *Proceeding of Fifth International Conference on Microelectronics, Computing and Communication Systems*Springer, Singapore.

[64] Q. Shen and Y. Zhao, "Perceptual hashing for color image based on color opponent component and quadtree structure," *Signal Processing*, vol. 166, article 107244, 2020.

[65] K. M. Singh, A. Neelima, T. Tuithung, and K. M. Singh, "Robust perceptual image hashing using SIFT and SVD," *Current Science*, vol. 117, no. 8, 2019.

[66] L. N. Vadlamudi, R. P. V. Vaddella, and V. Devara, "Robust image hashing using SIFT feature points and DWT approximation coefficients," *ICT Express*, vol. 4, no. 3, pp. 154–159, 2018.