Computers, Materials & Continua
DOI: 10.32604/cmc.2023.036216
Article

Tech Science Press

An Active Image Forgery Detection Approach Based on Edge Detection

Hüseyin Bilal Macit[1] and Arif Koyun[2,*]

[1]Department of IT Systems and Technologies, Mehmet Akif Ersoy University, Bucak Z. T. School of Applied Technology and Business, Burdur, 15300, Türkiye
[2]Department of Computer Engineering, Suleyman Demirel University, Engineering Faculty, Isparta, 32200, Türkiye
*Corresponding Author: Arif Koyun. Email: arifkoyun@sdu.edu.tr

Received: 21 September 2022; Accepted: 14 December 2022

Abstract: Recently, digital images have become the most used data, thanks to high internet speed and high resolution, cheap and easily accessible digital cameras. We generate, transmit and store millions of images every second. Most of these images are insignificant images containing only personal information. However, in many fields such as banking, finance, public institutions, and educational institutions, the images of many valuable objects like ID cards, photographs, credit cards, and transaction receipts are stored and transmitted to the digital environment. These images are very significant and must be secured. A valuable image can be maliciously modified by an attacker. The modification of an image is sometimes imperceptible even by the person who stored the image. In this paper, an active image forgery detection method that encodes and decodes image edge information is proposed. The proposed method is implemented by designing an interface and applied on a test image which is frequently used in the literature. Various tampering attacks are simulated to test the fidelity of the method. The method not only notifies whether the image is forged or not but also marks the tampered region of the image. Also, the proposed method successfully detected tampered regions after geometric attacks, even on self-copy attacks. Also, it didn't fail on JPEG compression.

Keywords: Image forgery; image tampering; edge detection

1 Introduction

Written texts are the most important sources of information for people in history. In today's world, the creation and distribution of visual media have become easier. People who interact with visual media tend to believe what they see, not what they read. It can be said that images are the most important source of information and they are the means of conveying information in today's world. Images are used on almost all platforms. Today, digital images are used in many fields, such as social media, law, industry, marketing, and medicine. Due to their widespread use, digital images are often manipulated and misused. The malicious manipulation of digital images to deceive people is called digital image forgery [1]. Forgery aims to make changes in the image without leaving a trace; that is, the changes made in the image are not easily detected [2].

In the days before digital photography, it was very difficult to modify an image. To create a forged photo, more than one photo or negative had to be cut, overlapped, and pasted [3]. The first image forgery in the literature is the fake photograph in which a French photographer named Hippolyte Bayard portrayed himself as a suicide victim in 1840 [4]. In another example, in 1860, after the Civil War in the USA, a forged image was distributed in which the head of Abraham Lincoln was placed over the body of vice president John Calhoun, who died in 1850 [5]. These forgeries are shown in Fig. 1.

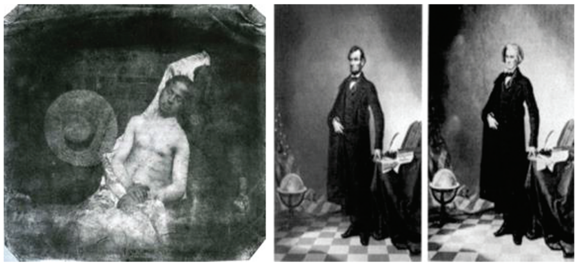


Figure 1: Image forgery before digital photography

With technological advances, almost all photographs are now created, stored, and transmitted digitally. In recent years, high-resolution cameras have become so cheap that many people can easily obtain them [6]. Powerful image processing software such as Adobe Photoshop, GIMP, Paintshop Pro, HitFilm Express, and Corel Paint are developed to edit digital images on the computer [7,8] Some of this software are paid and some are free [3]. Captured images with mobile phone-integrated high-resolution cameras can easily be edited with free software such as Google Photos, Snapseed, and Photoshop Express, and they can be sent end-to-end on the internet. These facilities enable even non-experts to do image forgery today. The use of forged images in e-mails, social media platforms, political campaigns, magazines, the fashion industry, and media organizations is increasing day by day [8]. This situation causes a decrease in trust in visual media. The forged images are mostly not detectable by the human visual system (HVS). Generally, whether an image is forged or not, there is nothing to worry about until it causes harm [3].

The most commonly used methods for image forgery can be expressed as copy-move attacks, image splicing attacks, retouching attacks, cropping attacks, and scaling attacks. The image forgery method, in which part of an image is copied and pasted into another part, is called copy-move forgery. It is also referred to as "cloning forgery" in the literature [2]. The purpose of this method is to hide a part of the image [1]. Since the copy-paste operation is performed on the same image, the basic properties of the image, such as noise, color, and texture do not change. Therefore, it is quite difficult to detect copy-move forgery [2]. In some forgery techniques, part(s) of one or more images are copied and pasted into another image [7]. These techniques are called "image splicing forgery". Professional software such as Photoshop is used to perform this forgery. Because the source and target images are different, the high-order Fourier statistics of the forged image are generally distorted [2] and forgery can be detected by pattern analysis [1]. Techniques in which the integrity of the image is not damaged but the image is enhanced and improved, are called "retouching forgery". In these techniques, operations such as smoothing, sharpening, and brightness and/or contrast changes can be performed on the whole or certain parts of the image. They are often used by photo editors to make the image more attractive [1]. There may be undesirable regions in some images, especially near the frame. The center of the image is magnified to remove these parts from the image. These forgery techniques are called “cropping forgery”. Especially in images with an embedded watermark or steganographic information in the spatial domain, the size or geometry of the image can be changed to destroy the hidden information. This type of attack is called a “scaling forgery”. Scaling forgery covers operations such as up-sampling, down-sampling, mirroring, skewing, and seam carving [3].

A forged image can be considered an original image by anyone as long as it is harmless. However, when an image causes harm, the image must be examined for forgery. Finding out whether the image has been manipulated is important to compensate for the damage caused by the image. The image may have been the subject of a court, news, insurance, or medical procedure [6]. There is a need for reliable methods that examine whether the image is original or manipulated [3]. In addition, if the image has been manipulated, the detection of the manipulated region is also important [3] and it is very difficult to detect [7]. The methods that perform these operations are called image forgery detection methods. As new image forgery detections are proposed, anti-forensic forgeries develop new image forgery methods to evade these techniques. Therefore, new image forgery detection techniques need to be constantly developed.

Image forgery detection methods are examined in different categories according to the detection approach. Let I be an image with m rows and n columns. I is composed of mn pixels, and the intensity of each pixel is expressed in 8 bits; that is with $2^8 = 256$ different intensity values. Assuming the I image is randomly generated, 256^{mn} different images can be generated. Assuming that the I image consists of 10×10 pixels, $256^{10.10}$ different I images can be generated. However, most of the randomly generated I images are meaningless, and HVS can easily distinguish whether the image is real or not [8]. Image forgery detection methods that predict the statistical meaninglessness of images are called “statistical forgery detection methods”. Some images may be lossy compressed with algorithms such as JPEG after being manipulated. Forgery detection on these images is very difficult and it is performed with format-based forgery detection methods. Some images are marked with special marks by the camera from which they were captured, just like the muzzle trace of the bullet coming out of the gun’s barrel. These signs can be sensor noise, camera filter array, chromatic signs, etc. The methods that detect a forgery in an image using these signs are called “camera-based forgery detection methods”. Apart from these, many methods are used to detect image forgery by using physical properties such as brightness, the direction of light, and contrast, or geometric properties such as focal point [2].

Image forgery detection techniques are divided into two classes: active and passive techniques. Active techniques need information that is already embedded in the image [9]. This information is a watermark or digital signature. While performing forgery detection, it can be decided whether the image is forged by checking the integrity and authenticity of the previously placed confidential information. In the digital watermarking method, I_w is created by encoding the watermark W into the original image I before the image is distributed. In the forgery detection stage, the watermark encoded in I_w is extracted as W_e, and the similarity between W and W_e is checked. The amount of similarity makes it possible to decide whether the image has been forged or not. In digital signature methods, the unique features of the original image are extracted as soon as the image is created. To detect forgery, the digital signature is reproduced from the image, which is taken from the distributed environment, and is compared with the original one [10]. Since active methods involve similar procedures, their advantages and disadvantages are also similar. Passive methods are also called blind methods or forensic methods and do not require pre-processing for image forgery detection. If there is particular information about the device that creates the image, such as a camera or scanner, we simply focus on identifying the image source. Generally, the device that produces the image stores some self-information, such as brand, model, manufacturer, image size, exposure time, and JPEG quantization matrix, as the image title data. If the title of the image is corrupted, it can be said that the image is forged [10]. If there is no prior information about the capturing device, we usually process the statistical data of the image. A manipulated image is most likely deconstructed and has various inconsistencies [6]. Passive techniques are applied based on pixels in the spatial domain or wavelet transforms in the frequency transform domain [1].

In this study, we proposed an active forgery detection method. Active methods check the authenticity of the image by looking at the integrity of the extracted watermark or digital signature. If the watermark is spread over the entire image, it can be easily recognized that there is a watermark in the image. This is a negative situation for digital image security. Also, most parts of the watermark may be distorted when geometric transformation or compression operations are applied to the image. In this case, the image may be detected as forged even if its integrity is not compromised. In this study, an active image forgery detection method is proposed that hides a very small and imperceptible watermark data only at the important points of the image, so that it can distinguish the real image in attacks that do not destroy the image integrity, and can also detect the forged region(s).

2 Proposed Method

A disadvantage of passive forgery detection methods with digital watermarking is that forgers can detect pre-processed images if the same watermark is embedded in all images during the preprocessing stage. A watermark can be perceived by the HVS when it is embedded in the frequency domain with an average watermark strength factor. A watermark embedded in the spatial domain is not easily detected by HVS and does not cause a significant change in the statistical values of the image. One of the most important pieces of information that reflects the details of an image is the edge data of the image. In this paper, we propose a new method that creates and embeds a specific watermark that contains particular edge information of the image. We used Lena's image as the distributed test image. We applied various image forgery techniques to the test image and evaluated the success of the proposed forgery detection method. The flow chart of the proposed method is shown in Fig. 2.

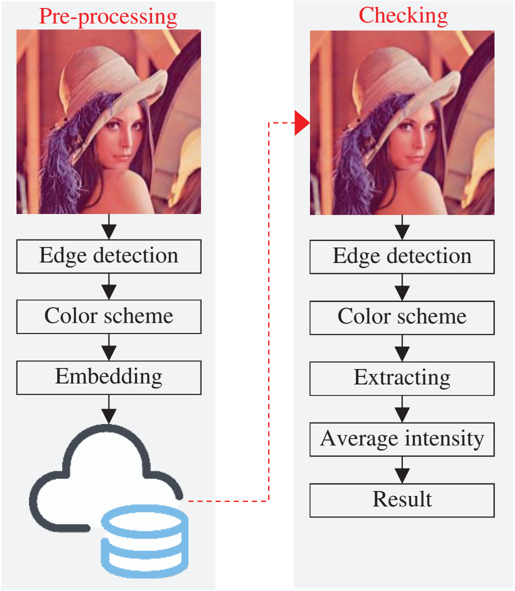


Figure 2: Flow chart of the proposed method

2.1 Pre-Processing

In the pre-processing phase, a pre-processed image is obtained by applying a series of operations on the original image. These operations are mapping the edge matrix of the original image, separating the image into color channels, and embedding the edge map into the corresponding color channels, respectively. The resulting image obtained at the end of these processes is aimed to be at a minimum distance from the original image, and it is ready to be distributed in an insecure digital environment.

2.1.1 Edge Detection

The sudden change in intensity while moving linearly in one direction over an image is called an edge [11]. The edge is the transition point from one piece of information to another one on the image. The methods that connect these points and localize the edges are called edge detection methods. An edge detection algorithm is based on the original image and locates the edge by obtaining the differentiation of the obvious gray changes in the image and it uses the gradient changes between the light and the shade [12]. Edge detection is frequently used in image processing applications to separate objects on the image from each other [13]. Edge detection is performed with different methods such as gradient sensitivity, object function, artificial neural network, Bayesian approach, wavelet transform, morphology, genetic algorithm, etc. The most widely used edge detection methods are classical methods such as Robert, Sobel, and Prewitt, which obtain gradients by processing neighboring pixels [14]. These methods use kernel matrices of different sizes depending on the application to obtain the gradient. These matrices are called “kernel” or “edge detection operators”. Edge detection is difficult on noisy images because the edges in such images contain high frequencies. Attempting to reduce noise may cause distortion or blurring of edges [15]. Conventional operators are sensitive to noise, but they cannot prevent interference [16]. The Canny operator is often used in applications that require a high signal-to-noise ratio (SNR) and detection sensitivity. Therefore, the Canny operator for edge detection is used in the method proposed in this paper.

The Canny operator is proposed by J. F. Canny [17] and is mentioned in the literature as a multi-scale optimal edge detector [12,18]. The main goals of the Canny algorithm are a low error rate, a minimal difference between real edge pixels and calculated edge pixels, and a single response to an edge. Let I be a 24-bit color image consisting of m rows and n columns to be preprocessed. For edge detection, the image needs to be monochrome. Therefore, the $I(m,n)$ image is configured as a $S(m,n)$ grayscale image (Fig. 3).

$$S\left(m,n\right) = \left\{x, y | 1 \le x \le m, 1 \le y \le n\right\}, x, y \in \left\{0, 1, 2, \ldots, 255\right\}$$

(a) (b)

Figure 3: (a) *I(m, n)* (b) *S(m, n)*

To remove possible noise in the S image, a two-dimensional Gaussian filter is applied to obtain a smoothed S image

$$S\left(m,n\right)=\frac{1}{2\pi\sigma^{2}}e^{-\frac{m^{2}+n^{2}}{2\sigma^{2}}}.$$

Here, σ denotes the standard deviation, which refers to the amount of smoothing on the image. If the σ value is less than 0.3, the Gaussian smoothing will have no noise reduction effect on the image [19]. If the σ value is selected to be greater than 2.5, the edges in the image can be softened so that they cannot be detected. Therefore, it is much better to choose the value of σ between 0.3 and 2.5.

Edge strength can be determined by finding which direction the brightness changes most in a neighboring pixel group in a monochrome image [20]. The gradient information is obtained on x and y coordinates by moving the Cx and Cy Sobel convolution matrices on the S image (Fig. 4).

-1	0	+1
-2	0	+2
-1	0	+1

(a)

+1	+2	+1
0	0	0
-1	-2	-1

(b)

Figure 4: (a) *Cx* and (b) *Cy* convolution matrices

The gradient of an image shows the change in color intensity as we move through the image in one direction. In other words, the gradient is a vector quantity with direction and magnitude (Fig. 5), and it is one of the fundamental parts of image processing.

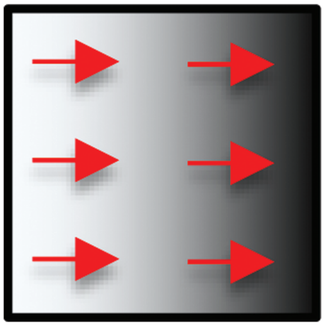


Figure 5: Sample image gradient vector

Let $P_x(i,j)$ and $P_y(i,j)$ be the first-order partial derivative in the x and y directions of a portion of the S image of size $i\times j$. In this case, the gradient size of this piece is calculated as follows

$$M\left(i,j\right)=\sqrt{P_x(i,j)^2+P_y(i,j)^2}.$$

And the direction of this gradient θ is

$$\theta\left(i,j\right)=\arctan\left[\frac{P_x\left(i,j\right)}{P_y\left(i,j\right)}\right].$$

θ is equated to the nearest applicable edge direction in a two-dimensional matrix

$$\theta\ (i,j) = \begin{cases} 0, & 22.5 > \theta\ (i,j) \geq 0. \\ 45, & 67.5 > \theta\ (i,j) \geq 22.5 \\ 90, & 112.5 > \theta\ (i,j) \geq 67.5 \\ 135, & 157.5 > \theta\ (i,j) \geq 112.5 \end{cases}$$

For example, if θ (i,j) is calculated to be 4, we round this value to 0 degrees. Edge detection is performed on $P_{x,y}$ with the rounded angle θ. If θ = 0 for $P_{x,y}$ as in Fig. 6, the edge line is on $P_{2,1}$, $P_{2,2}$, and $P_{2,3}$.

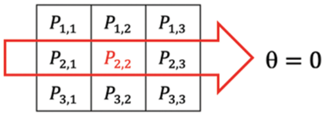


Figure 6: Sample image portion $P_{x,y}$ for convolution

We figure out the gradient of the S image in the x and y direction (the partial derivative of S concerning x and y) by operating Cx and Cy over the S image (Fig. 7).

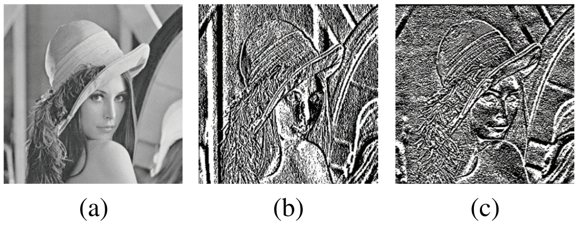

(a) (b) (c)

Figure 7: (a) $S(m, n)$ (b) h_x (c) h_y

$$h_x = \frac{\partial S}{\partial x} = C_x \cdot S$$

$$h_y = \frac{\partial S}{\partial y} = C_y \cdot S$$

The resulting gradient vector

$$\nabla \mathrm{C} = \begin{bmatrix} h_x \\ h_y \end{bmatrix}$$

Now, some regions are perceived as edges that are not actually edges due to high-frequency noise above ∇C. To solve this problem for each $P_x(i,j)$ and $P_y(i,j)$, their lesser neighbors along the gradient direction θ(i,j) are set to zero. The double threshold method is applied to eliminate the false edges and join the cut edges. For this, with threshold values $\tau_1 > \tau_2$, if $P(x,y) > \tau_1$; the pixel is marked as an edge pixel. If the neighbors of $P(x,y)$ are $P(x \pm 1, y \pm 1) > \tau_2$, the value of these neighbors is set to 1. Thus, the dashed border lines are joined. As a result, the binary matrix $E(m,n)$is created (Fig. 8), which corresponds to the edge map of the $S(m,n)$ image.

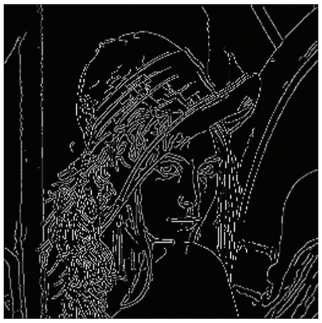


Figure 8: Binary edge matrix *E (m, n)* of test image

2.1.2 Color Scheme

A color scheme is a model that mathematically expresses the color information of an image [21]. In digital image processing applications, there are color-based Red Green Blue (RGB), tone-based Hue Saturation Value (HSV), Hue Saturation Intensity (HSI), Hue Saturation Lightness (HSL), and brightness-based luminance chrominance color scheme (YCbCr). The RGB color scheme is based on the mathematical expression of the intensity of the primary colors red, green, and blue for storing digital images. Tone-based color schemes are often used to distinguish regions of the desired color within an image. Brightness-based color schemes are frequently used in image compression methods. In this study, the YCbCr color scheme is chosen to embed the watermark in the spatial domain. Because, the embedding process is performed with a steganographic approach, and the YCbCr color space is the optimum technic for a spatial domain application [22]. Let, I be a color image with m rows and n columns

$$I\left(m,n\right)=\left\{x_{i,j}|1\le i\le m,1\le j\le n\right\}$$

Each $x_{i,j}$ represents a pixel of the I image. The I image is expressed in the RGB color scheme which is a combination of the $R(m,n)$, $G(m,n)$, and $B(m,n)$ matrices. To convert I to YCbCr color scheme

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix}=\begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix}\begin{pmatrix} R \\ G \\ B \end{pmatrix}+\begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}.$$

Thus, Y, Cb and Cr matrices are obtained. Y represents the luminance, Cb and Cr represent the chrominance of the image. Fig. 9 shows plotted Y, Cb, and Cr matrices of the test image.

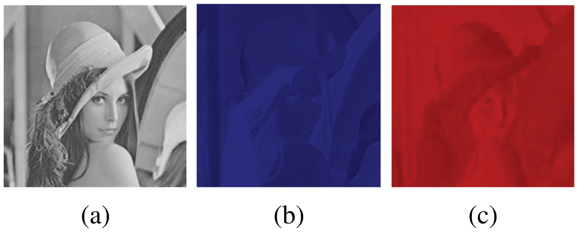


Figure 9: Y, Cb and Cr matrices of the test image

2.1.3 Embedding

Y channel represents the brightness of the pixel, and it is resistant to interference insertion attacks. *Cb* channel is blue chrominance and it resists clipping in the image. The *Cr* channel is red chrominance and it is resistant to spin attacks [23]. The HVS is very good at distinguishing brightness. Thus, a small alteration in the *Y* channel can be discerned by HVS, while larger alternations in *Cb* and *Cr* channels cannot. At this stage, the binary matrix *E* is embedded into the original image. Watermarking or steganographic approaches are used to embed an image into another one. The basic idea of watermarking and steganography is to hide a secret message, signal, or image in a cover image [24]. The general purpose of watermarking methods is to guarantee the robustness of the watermark [25]. In steganography methods, the aim is to ensure that confidential information cannot easily be detected by HVS while keeping the data payload as high as possible [26]. In both data embedding strategies, spatial domain and frequency domain techniques are used according to the application purpose. Confidential information (watermark) embedded with spatial domain techniques is fragile but not easily detectable by HVS. Transform techniques such as discrete wavelet transform (DWT), discrete cosine transform (DCT), and fast fourier transform (FFT) are used in the frequency domain, and secret information is more robust but usually distinguishable by HVS. In this paper, it is aimed to embed the generated *E* matrix into the *I* image in a way that is indistinguishable by HVS. Thus, *E* matrix can be so fragile and it may corrupt at the slightest tampering. Therefore, embedding is performed in the spatial domain into the *Cb* and *Cr* channels. *E* is embedded into the least significant 3 bits (3LSB) of the *Cb* and *Cr* channels of corresponding pixels. The advantages of the approach are the simplicity of the method and the data group carrying the *E* matrix can easily be corrupted in attacks such as filtering, adding noise, blurring, clipping, and copy-move. Let $cb_{i,j}$ be the i. row and j. column pixel of *Cb*, $cr_{i,j}$ be the i. row and j. column pixel of *Cr*, and $e_{i,j}$ be the i. row and j. column pixel of *E*

$cb_{i,j} \in Cb\,(m,n)\,, 0 < cb_{i,j} < 255$ and $cr_{i,j} \in Cr\,(m,n)\,, 0 < cr_{i,j} < 255.$

If $e_{i,j}$ represents an edge of *I*, it has a value of 1.

$$e_{i,j} \in E\,(m,n)\,, e_{i,j} = \begin{cases} 1, I_{i,j} \text{ is on the edge} \\ 0, I_{i,j} \text{ isn't on the edge} \end{cases}$$

Binary expression of the *Cb* and *Cr* layers of the *I* image, which is mapped with 8 bits of pixels in each color space are

$\left(cb_{i,j}\right)_{10} = (b_8b_7b_6b_5b_4b_3b_2b_1)_2$ and $\left(cr_{i,j}\right)_{10} = (b_8b_7b_6b_5b_4b_3b_2b_1)_2.$

Here, the last 3 bits in the binary expression of $cb_{i,j}$ and $cr_{i,j}$ are assumed to be the 3LSBs. The edge data is embedded which is stored by the *E* matrix in the last 3 bits of each pixel of *Cb* and *Cr*.

$$cb_{i,j}\left(b_{6,7,8}\right) = cr_{i,j}\left(b_{6,7,8}\right) = \begin{cases} 1, e_{i,j} = 1 \\ 0, e_{i,j} = 0 \end{cases}$$

The maximum data size stored by 3LSBs is $2^3 = 8$. Since the total data size is $2^8 = 256$, the change of all the 3LSBs in this color channel causes only a change of 8/256 = 3.125%. This negligible change in the *Cb* and *Cr* channels cannot easily be detected by the HVS. This is proved by vectorial proximity methods in the next sections of this paper. The process of embedding the *E* matrix in the *Cb* and *Cr* channels is shown in Fig. 10.

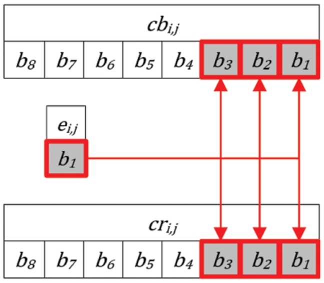

$cb_{i,j}$
b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1
$e_{i,j}$
b_1
$cr_{i,j}$
b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1

Figure 10: Embedding E binary matrix to 3LSBs of chrominance channels

The modified *Cb*′and *Cr*′ matrices are combined with the *Y* luminance matrix to obtain the *P* pre-processed image. The embedding on the test image is performed with the proposed method. The original and resulting pre-processed images are shown in Fig. 11.

(a) (b)

Figure 11: (a) Original test image *I*, (b) pre-processed test image *P*

If the embedded data in the pre-processed image is detected by the HVS, we can say that this method has failed. There are some mathematical methods for measuring how far an image deviates from the original after it has been manipulated. In this paper, Peak Signal-to-Noise-Ratio (PSNR), Normalized Correlation (NC), and Structural Similarity Index Measure (SSIM) values are measured to evaluate the perceptual distance from *I* to *P*. PSNR is a logarithmic quantity that calculates the possible noise generated in a signal in decibels by comparing the noisy signal with the original one [27].

$$PSNR = 10log_{10}\left[\frac{MI^2}{\frac{1}{m.n}\sum_m \sum_n (I-P)^2}\right]$$

Here, *MI* is the maximum intensity value. For the 24 bit mapped test image, each channel is coded in 8 bits, $MI = 2^8 - 1 = 255$. *m* and *n* are respectively row and column count of *I* and *P*. If the original image and the pre-processed image are the same, $PSNR = \infty$. The PSNR value of the test image after the pre-processing phase is calculated as 41.6506. PSNR is an engineering term and may not appeal to HVS. NC and SSIM are much closer to the perception of HVS. NC is a quality metric and a measurement of a time series. In digital images, NC measurement is usually made by hovering a pattern over the image to search for a pattern on the image. If the pattern is exactly the same as the region on the image, the NC value is calculated as 1 [25].

$$NC = \frac{\sum_m \sum_n \left(I_{m,n} - \mu_I \left(P_{m,n} - \mu_P\right)\right)}{\sqrt{\sum_m \sum_n \left[I_{m,n} - \mu_I\right]^2 \sum_m \sum_n \left[P_{m,n} - \mu_P\right]^2}}$$

μ_I and μ_P are the arithmetic mean of the I and P images, respectively. The NC value of the test image after pre-processing is calculated as 0.99993. SSIM is a method that measures structural similarity between two uncompressed images. SSIM is the closest mathematical metric to the perception of HVS. SSIM first calculates three parameters; luminosity, degradation, and degradation. These factors are calculated as in the equations, respectively:

$$l(I,P) = \left(\frac{2\mu_I \mu_{I_w} + k_1}{{\mu_I}^2 + {\mu_{I_w}}^2 + k_1}\right)$$

$$c(I,P) = \left(\frac{2\sigma_I \sigma_{I_w} + k_2}{{\sigma_I}^2 + {\sigma_{I_w}}^2 + k_2}\right)$$

$$s(I,P) = \left(\frac{2\sigma_{II_w} + k_3}{\sigma_I + \sigma_{I_w} + k_3}\right)$$

SSIM is calculated by l, c and s values

$$SSIM(I,P) = l(I,P)^{\alpha} \cdot c(I,P)^{\beta} \cdot s(I,P)^{\gamma}$$

If I and P images are the same, SSIM is calculated 1. As images perceptually differ to each other, SSIM goes for 0. The SSIM value of the test image after pre-processing phase is calculated as 0.99812.

2.2 Checking

The main purpose of this step is to detect whether P is forged or not after getting it from a communication channel or a storage device. For this, the operations of mapping the edge matrix of the image, separating the image into color channels, extracting the edge matrix, and calculating average intensity with threshold are performed, respectively (Fig. 1.). The edge detection and color scheme processes that were applied to the I image at the checking stage, are also applied to the P image in the same way at this stage. The next steps are described under the topics of extracting and average intensity, respectively.

2.2.1 Extracting

Let E' be the edge matrix obtained from P. There is definitely a noise difference between E and E' due to the change made in the 3 LSB of the Cr and Cb layers during the pre-processing stage of P. This noise difference is saved in the binary N matrix. For this, XOR operation is applied to E and E' matrices

$$N_{i,j} = E_{i,j} \oplus E'_{i,j}$$

For an unmanipulated P image, N contains only natural noise. But if P is manipulated, both natural and artificial noise in N is encountered. At this stage, it is necessary to clearly distinguish between natural noise and artificial noise.

2.2.2 Average Intensity

Natural noise is diffused, while artificial noise is concentrated in the manipulation area. So, the artificial noise in N gives the tampered region of the P image. We need to avoid natural noise. One way to remove it is to soften the image. For this, methods such as Gaussian smoothing can be used. But N is a binary matrix and smoothing may increase natural noise. A TM convolution matrix of size $m_t \cdot n_t$ is used with $TM < P$ to eliminate the natural noise in N. has hovered over N. If the arithmetic means μ_{TM} of the TM matrix is higher than the predefined threshold value $0 < \tau < 1$, the region on which the TM matrix falls on N is marked as forged.

$$\mu_{TM} = \sum_{1}^{tm} \sum_{1}^{tn} TM_{ti-i+1,tj-j+1}$$

$$N_{i,j} = \begin{cases} 1, & \mu_{TM} \geq \tau \\ 0, & \mu_{TM} < \tau \end{cases}$$

The results of the average intensity phase on the unforged test image are shown in Fig. 12. It is clearly seen that natural noise is completely eliminated.

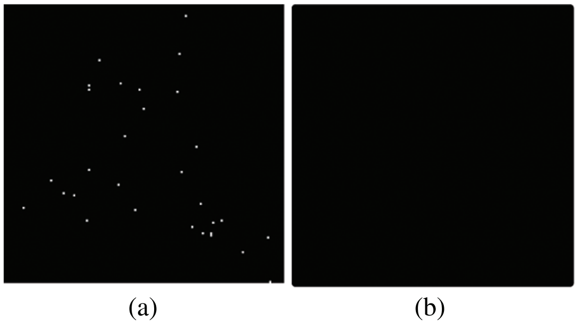


Figure 12: (a) N with natural noise (b) N after average intensity phase

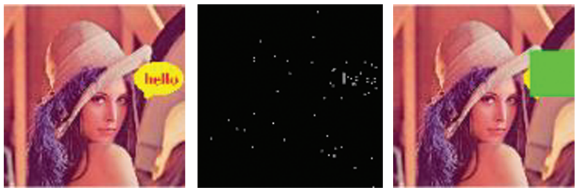
We can decide if the distributed image P is forged or not according to the sum of all pixel values of binary N matrix.

$$N\left(m,n\right) = \begin{cases} forged, & \sum_{1}^{m} \sum_{1}^{n} N_{i,j} > 0 \\ not\ forged, & \sum_{1}^{m} \sum_{1}^{n} N_{i,j} = 0 \end{cases}$$

It is important to select the correct TM sizes and a threshold value. This is evaluated by several iterations with various TM sizes and threshold values on the test image and is explained in subsequent sections of the article. A sample forgery scenario and detection with the proposed method on test image is shown in Fig. 13. Here, the convolution matrix size and threshold values are $m_t = 128, n_t = 128, \tau = 0.3$, respectively.

3 Results and Discussion

Generally, the performance of a forgery detection method is evaluated by HVS. The weakness of forgery detection methods is that they perceive the unforged image as forged or the forged image as unforged. Also, some methods have the ability to detect the tampered region of the image. But the weakness of these methods is that they may perceive the unforged region as forged or the forged region as unforged.



(a) (b) (c)

Figure 13: (a) Forged P (b) N of P (c) estimated forged region

In statistics, the probability that a value that is actually negative will be falsely classified as positive is called the False Positive Rate (FPR)

$$FPR = \frac{FP}{FP + TN}.$$

FPR is calculated as a decision maker for TM size and the threshold value. False Positive (FP) indicates the number of TMs that marked untampered regions as tampered, while True Negative (TN) indicates the number of TMs that correctly detected tampered regions. We artificially forged certain ratios of regions of P and implemented the proposed forgery detection method with various TM sizes and threshold values as shown in Table 1.

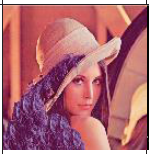
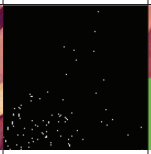
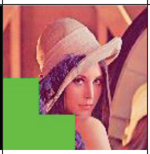
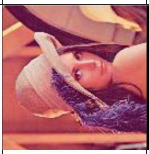
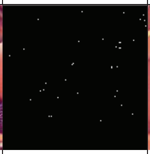
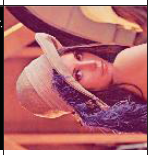
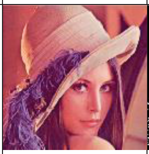
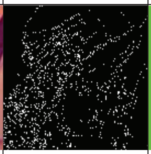
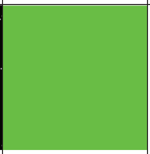
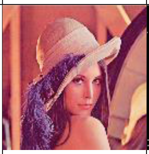
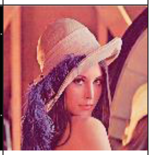
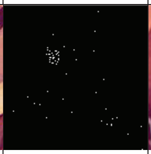
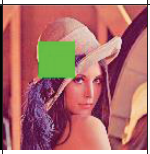
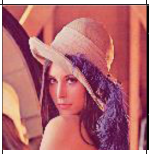
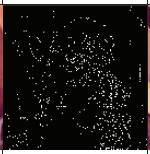
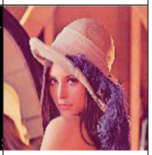
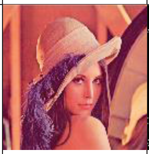
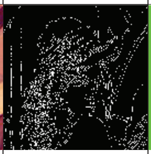
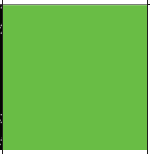
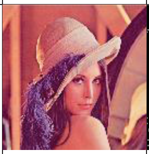
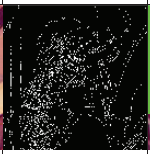
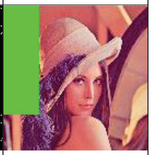
Table 1: FPRs of the proposed method with various TM sizes and T values on the test image

Tamper ratio	TM	τ				
		0.1	0.2	0.3	0.4	0.5
1.56%	**8** × **8**	0.94581	0.81587	0.671875	0.560033	0.477865
	16 × **16**	0.869963	0.730225	0.689002	0.663628	0.5
	32 × **32**	0.825439	0.726144	0.532715	0.597005	0.484375
	64 × **64**	0.793701	0.711182	0.690552	NaN	NaN
3.52%	**8** × **8**	0.916079	0.747396	0.621094	0.564174	0.588867
	16 × **16**	0.828425	0.633042	0.614014	0.523112	0.306641
	32 × **32**	0.726119	0.625366	0.500488	0.500488	0.12207
	64 × **64**	0.850146	0.812683	0.759684	0.546875	0.546875
6.25%	**8** × **8**	0.888485	0.682603	0.579985	0.537054	0.542799
	16 × **16**	0.773853	0.610618	0.594336	0.526425	0.427734
	32 × **32**	0.703055	0.597168	0.49646	0.368327	0.11499
	64 × **64**	0.707554	0.641553	0.631042	0.492188	0.546875

The lowest FPR is the best. When the results are analyzed, it is clearly seen that optimum values gather around $m_t = 32, n_t = 32, \tau = 0.5$ values. In other words, the proposed method detected image forgery at the highest rate when TM size is as $t_m = \frac{m}{16}$, $t_n = \frac{n}{16}$, and $\tau = 0.5$. It is also proved by several iterations on high-resolution images that are not mentioned in the article.

For an image to be considered forged, it must have been irreversibly altered. Copy-move, cropping, splicing, and retouching attacks irreversibly distort the image, but geometric attacks such as mirroring, scaling, and rotating only change the geometry of the image. Images exposed to geometric attacks are not considered as forged because these attacks do not make a semantic change to the image. We simulated the proposed method on all these attacks, and the results are shown in Table 2.

Table 2: Forgery detection results of test image

	Forged *P*	*E*	Result		Forged *P*	*E*	Result
Copy-move				Rotating			
Cropping				Scaling (up&down)			
Splicing				Mirroring			
Retouching				JPEG Compression			

The green areas are marked as tampered regions in the result column. As it is seen, the proposed method clearly identified the tampered regions on forged images, and it did not fail on geometric attacks. Also, it identified the JPEG compressed image as partially forged.

Table 3 shows the qualitative comparison of the proposed approach with state-of-the-art approaches The results of this table indicate that the proposed approach has pros and cons. The pros are: (1) it does not need any side information to detect forgery; (2) it can be used for image copyright protection; (3) it is robust to known attacks; (4) it is the only approach that performs tests and gives results on different ratios of forgery. The cons are: (1) It is an active method and needs pre-processing; (2) since it is an active method, it cannot be compared mathematically with passive methods.

Table 3: Comparison of the proposed approach with the state-of-the-art approaches

Paper	Year	Technic	Forgery (detection) type	Decision maker	Performance
[28]	2015	Passive	Copy-move forgery	Nearest neighbor distance ratio	TPR: 71.92% FPR: 1.22% Accuracy: 85.35%
[29]	2017	Passive	Copy-move forgery	Generalized 2 nearest neighbor	Precision: 93.3% Recall: 87.5%
[30]	2017	Passive	Block based copy-move forgery	Cascading matching using Euclidean distances	Accuracy rate: 98% FNR: 8%
[31]	2017	Active	Random	Threshold based classification	No mathematical results given.
[32]	2021	Passive	Face image manipulation	Convolutional neural network	Accuracy rate: 72.52%
[33]	2021	Passive	Random	Improved relevance vector machine	Accuracy rate: 92.22% Sensitivity rate: 88.4% Specificity rate: 97.6%
[34]	2021	Passive	Copy-move forgery	Convolutional neural network	FPR: 2%
[35]	2021	Passive	Copy-move forgery	Support vector machine	Accuracy: 98.44%
[36]	2022	Active	Fragile watermark analysis	Chaotic functions	No mathematical results given.
[37]	2022	Active	Random	Block based watermarking	Precision between 91% and 98% depending on forgery type
Proposed	2022	Active	Random	Watermarking based on edge detection	FPR between 0.11% 0.95% on several iterations with various forgery parameters

4 Conclusions

An image cannot protect itself when taken to storage or transmission media. Various methods have been proposed in the literature to ensure image security. In this paper, we propose an active image forgery detection method that performs image security by embedding edge information in the image's chrominance layers in an imperceptible way. We coded the proposed method with the MATLAB programming language and tested it with the Lena test image with several different parameters. The proposed method has detected 100% of the forgery attacks such as copy-move and image splicing that perform partial tampering of the image. In addition, it detected the whole forgery on retouched or cropped images. We measured the FPR values with different parameters to monitor situations where the proposed method might fail. Finally, the method showed its success by confirming the originality of the image in rotating, mirroring, and scaling attacks, which many image forgery detection methods perceive as a forgery.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study. And all authors have equally contributed.

References

[1] C. Kaur and N. Kanwal, "An analysis of image forgery detection techniques," *Statistics Optimization and Information Computing*, vol. 7, no. 2, pp. 486–500, 2019.

[2] A. Kashyap, R. S. Parmar, M. Agarwal and H. Gupta, "An evaluation of digital image forgery detection approaches," ArXiv, vol. 12, no. 15, pp. 4747–4758, 2017.

[3] S. Walia and K. Kumar, "Digital image forgery detection: A systematic scrutiny," *Australian Journal of Forensic Sciences*, vol. 51, no. 5, pp. 488–526, 2019.

[4] T. Qazi, K. Hayat, S. U. Khan, S. A. Madani, I. A. Khan *et al.,* "Survey on blind image forgery detection," *IET Image Processing*, vol. 7, no. 7, pp. 660–670, 2013.

[5] H. Farid, "Digital doctoring: How to tell the real from the fake," *Significance*, vol. 3, no. 4, pp. 162–166, 2006.

[6] K. Asghara, Z. Habiba and M. Hussain, "Copy-move and splicing image forgery detection and localization techniques: A review," *Australian Journal of Forensic Sciences*, vol. 49, no. 3, pp. 281–307, 2017.

[7] C. N. Bharti and P. Tandel, "Survey of image forgery detection techniques," in *Proc. of IEEE WiSPNET Conf.*, Chennai, India, pp. 877–881, 2016.

[8] H. Farid, "Image forgery detection, a survey," *IEEE Signal Processing Magazine*, vol. 26, no. March Issue, pp. 16–25, 2009.

[9] N. K. Gill, R. Garg and E. A. Doegar, "A review paper on digital image forgery detection techniques," in *Proc. of 8th ICCCNT*, Delhi, India, pp. 1–7, 2017.

[10] T. Mahmood, T. Nawaz, R. Ashraf, M. Shah, A. Khan *et al.,* "A survey on block-based copy move image forgery detection techniques," in *Int. Conf. on Emerging Technologies (ICET)*, Peshawar, Pakistan, pp. 1–6, 2015.

[11] S. R. Joshi and R. Koju, "Study and comparison of edge detection algorithms," in *Third Asian Himalayas Int. Conf. on Internet*, Kathmandu, Nepal, pp. 1–5, 2012.

[12] Z. Wang, K. Li, X. Wang and A. Lee, "An image edge detection algorithm based on multi-feature fusion," *Computers, Materials & Continua*, vol. 73, no. 3, pp. 4996–5009, 2022.

[13] A. Mouse, "Canny edge-detection based vehicle plate recognition," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 5, no. 3, pp. 1–7, 2012.

[14] C. X. Deng, G. B. Wang and X. R. Yang, "Image edge detection algorithm based on improved Canny operator," in *Proc. of the 2013 Int. Conf. on Wavelet Analysis and Pattern Recognition*, Tianjin, China, pp. 168–172, 2013.

[15] A. Himanshu, "Study and comparison of various image edge detection techniques," *International Journal of Image Processing*, vol. 3, no. 1, pp. 1–12, 2009.

[16] M. Sonka, V. Hlavac and R. Boyle, *Image Processing Analysis and Machine Vision*. Beijing: Posts & Telecom Press, 2002.

[17] J. F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[18] Z. Xu, X. Baojie and W. Guoxin, "Canny edge detection based on Open CV," in *IEEE 13th Int. Conf. on Electronic Measurement & Instruments*, Yangzhou, China, pp. 53–56, 2017.

[19] K. Çelik, "Gradyan uyarlamalı görüntü filtresi tasarımı," M.S. dissertation, Gazi University, Graduate School of Natural and Applied Sciences, Ankara, Turkey, 2015.

[20] A. S. Ahmed, "Comperative study among Sobel, Prewitt and Candy edge detection operators used in image processing," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 19, pp. 6517–6525, ISSN: 1992-8645, 2018.

[21] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat and J. Jatakia, "Human skin detection using RGB, HSV and YCbCr color models," *Advances in Intelligent Systems Research*, vol. 137, pp. 324–332, 2017.

[22] S. Hemalatha, U. D. Acharya and A. Renuka, "Comparison of secure and high-capacity color image steganography techniques in RGB and YCBCR domains," *International Journal of Advanced Information Technology*, vol. 3, no. 3, pp. 1–9, 2013.

[23] E. Vahedi, R. A. Zoroofi and M. Shiva, "On optimal color coordinate selection for wavelet-based color image watermarking," in *Int. Conf. on Intelligent and Advance Systems*, Kuala Lumpur, pp. 635–640, 2007.

[24] A. Baumy, A. D. Algarni, M. Abdalla, W. El-Shafai, F. E. Abd El-Samie *et al.,* "Efficient forgery detection approaches for digital color images," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3257–3276, 2022.

[25] C. Patvardhan, C. Kumar and C. V. Lakshmi, "Effective color image watermarking scheme using YCbCr color space and QR code," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12655–12677, 2018.

[26] S. A. Laskar, "High-capacity data hiding using LSB steganography and encryption," *International Journal of Database Management Systems*, vol. 4, no. 6, pp. 57–68, 2012.

[27] J. A. Hussein, "Spatial domain watermarking scheme for colored images based on log-average luminance," *Journal of Computing*, vol. 2, no. 1, pp. 100–103, ISSN:2151-9617, 2010.

[28] E. Silva, T. Carvalho, A. Ferreira and A. Rocha, "Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes," *Journal of Visual Communication and Image Representation*, vol. 29, no. 1, pp. 16–32, 2015.

[29] W. Zhang, Z. Yang, S. Niu and J. Wang, "Detection of copy-move forgery in flat region based on feature enhancement," in *Digital Forensics and Watermarking, Lecture Notes in Computer Science*, Vol. 10082. Germany: Springer, pp. 159–171, 2017.

[30] D. Huang, C. Huang, W. Hu and C. Chou, "Robustness of copy-move forgery detection under high JPEG compression artifacts," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 1509–1530, 2017.

[31] V. Tuba, R. Jovanovic and M. Tuba, "Digital image forgery detection based on shadow HSV inconsistency," in *5th Int. Symp. on Digital Forensic and Security*, Tirgu Mures, Romania, pp. 1–6, 2017.

[32] S. Lee, S. Tariq, Y. Shin and S. S. Woo, "Detecting handcrafted facial image manipulations and GAN-generated facial images using shallow-FakeFaceNet," *Applications Soft Computing*, vol. 105, no. 1, pp. 107256, 2021.

[33] N. K. Rathore, N. K. Jain, P. K. Shukla, U. S. Rawat and R. Dubey, "Image forgery detection using singular value decomposition with some attacks," *Natural Academy Science Letters*, vol. 44, no. 3, pp. 331–338, 2021.

[34] N. Goel, S. Kaur and R. Bala, "Dual branch convolutional neural network for copy move forgery detection," *IET Image Processing*, vol. 15, no. 3, pp. 656–665, 2021.

[35] I. T. Ahmed, B. T. Hammad and N. Jamil, "Image copy-move forgery detection algorithms based on spatial feature domain," in *EEE 17th Int. Colloquium on Signal Processing & Its Applications*, Langkawi, Kedah, Malaysia, pp. 92–96, 2021.

[36] O. Benrhouma, "Cryptanalysis of a hamming code and logistic-map based pixel-level active forgery detection scheme," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, pp. 663–668, 2022.

[37] M. Z. Salim, A. J. Abboud and R. A. Yıldırım, "Visual cryptography-based watermarking approach for the detection and localization of image forgery," *Electronics*, vol. 11, no. 1, pp. 136, 2022.