redin Revista Facultad de Ingeniería

UNIVERSIDAD DE ANTIOQUIA 1803

Title: **Hybrid Algorithm for the detection of Pixel-based digital image forgery using Markov and SIFT descriptors**

Authors: Jimmy Alexander Cortés-Osorio, José Andrés Chaves-Osorio and Cristian David López-Robayo

DOI: **10.17533/udea.redin.20211165**

To appear in: *Revista Facultad de Ingeniería Universidad de Antioquia*

Received: January 30, 2021
Accepted: October 29, 2021
Available Online: November 02, 2021

Please cite this article as: J. A. Cortés, J. A. Chaves and C. D. López. Hybrid Algorithm for the detection of Pixel-based digital image forgery using Markov and SIFT descriptors, *Revista Facultad de Ingeniería Universidad de Antioquia*. [Online]. Available: https://www.doi.org/10.17533/udea.redin.20211165

Hybrid Algorithm for the detection of Pixel-based digital image forgery using Markov and SIFT descriptors

Algoritmo híbrido mediante descriptores Markov y SIFT para la detección de la falsificación de imágenes

Jimmy Alexander Cortés-Osorio*https://orcid.org/0000-0002-0413-807X , José Andrés Chaves-Osorio https://orcid.org/0000-0002-9434-9224, Cristian David López-Robayo https://orcid.org/0000-0002-7937-4683

Facultad de Ciencias Básicas, Universidad Tecnológica de Pereira. Carrera 27 #10-02. C. P. 660003. Pereira, Risaralda

Corresponding Author: Jimmy Alexander Cortés-Osorio

E-mail: jacoper@utp.edu.co

ABSTRACT: Today, image forgery is common due to the massification of low-cost/high-resolution digital cameras, along with the accessibility of computer programs for image processing. All media is affected by this issue, which makes the public doubt the news. Though image modification is a typical process in entertainment, when images are taken as evidence in a legal process, modification cannot be considered trivial. Digital forensics has the challenge of ensuring the accuracy and integrity of digital images to overcome this issue. This investigation introduces an algorithm to detect the main types of pixel-based alterations such as copy-move forgery, resampling, and splicing in digital images. For the evaluation of the algorithm, CVLAB, CASIA V1, Columbia, and Columbia Uncompressed datasets were used. Of 7100 images evaluated, 3666 were unaltered, 791 had resampling, 2213 had splicing, and 430 had copy-move forgeries. The algorithm detected all proposed forgery pixel methods with an accuracy of 91%. The main novelties of the proposal are the reduced number of features needed for identification and its robustness for the file format and image size.

KEYWORDS

Copy-Move; DCT; Markov; Resampling; SIFT; Splicing

Copiar-Mover; DCT; Markov; Remuestreo; SIFT; Empalme

RESUMEN: Hoy en día, la falsificación de imágenes es común debido a la masificación de las cámaras digitales de alta resolución y bajo costo, junto con la accesibilidad de los programas de computadora para el procesamiento de imágenes. Todos los medios de comunicación se ven afectados por este tema, lo que hace que el público dude de la noticia. Aunque la modificación de imágenes es un proceso común en el entretenimiento, cuando las imágenes se toman como evidencia en un proceso legal, la alteración no puede considerarse trivial. La ciencia forense digital tiene el desafío de garantizar la precisión y la integridad de las imágenes digitales para superar este problema. Esta investigación introduce un algoritmo para detectar los principales tipos de alteraciones basadas en píxeles, como copy-move, resamplig y splicing en imágenes digitales. Para la evaluación del algoritmo se utilizaron las bases de datos CVLAB, CASIA V1, Columbia y Uncompressed Columbia. Se evaluaron 7.100 imágenes, de las cuales 3666 eran auténticas, 791 tenían resampling, 2213 tenían splicing y 430 tenían falsificaciones de copy-move. El algoritmo detectó todas las alteraciones basadas en pixeles con una precisión del 91%. Las principales novedades de la propuesta son el reducido número de características necesarias para la identificación y su robustez al formato y tamaño de la imagen.

1. Introduction

Image manipulation is more common today due to the massification of low-cost/high-resolution cameras, along with the availability of programs for image processing such as Inkscape, Photoshop, and Corel Draw, among others. Although image alteration is common in entertainment, when images are taken as evidence in a legal process maintaining the integrity of the original image is fundamental. Thus, digital forensics has the challenge of ensuring the accuracy and integrity of images in a legal process.

Various researchers have made significant efforts to identify manipulated images using digital image processing algorithms. This is because the process of visually identifying alterations to digital images is complicated for the human eye. Researchers have found that the process of falsifying an image modifies the neighborhood and statistics of the host image [1]. Thanks to these traces of alterations, it is possible to detect altered images. The scientific literature describes different approaches to detect the falsification of digital images, including methods based on the type of camera, as well as the format, physics, geometry, and pixels of the image [2]. One of the main advantages of pixel-based methods is that they do not require knowledge of either the camera manufacturer's parameters or the original image.

Among the most common alterations in pixel-based methods are copy-move, resampling, and splicing. According to [3], copy-move alteration takes place when a region of an image is copied and pasted into the same image without any geometric transformation. In contrast, if at the time of pasting the copied region some geometric transformation is performed, this is known as resampling. Splicing is generated by copying a region of an image and pasting it in a different one to add or hide important information. Although these operations may be visually imperceptible, it is possible to find statistical changes in the image due to a correlation change between neighboring pixels at the edges of the base image [4].

1.1. Copy-move detection

Various state-of-the-art approaches are available to identify copy-move alterations. However, for most of the algorithms studied, the detection of forgeries consists of four main stages: preprocessing feature extraction, pairing, and visualization [5]. One of the most used methods for copy-move forgery identification is the Scale-Invariant Feature Transform (SIFT), as this method works on the image's key points extraction. In [6], authors presented a hybrid method for detecting copying on an image using SIFT and the Principal Component Analysis (PCA) kernel to extract the main points by blocks. A variation of the previous method was presented in [7], in which they used the KAZE point detector together with SIFT to extract more crucial points. A method to identify copy-move forgeries with the

Discrete Wavelet Transform (DWT) was proposed in [8]. Initially, they applied the DWT on the image to obtain both the detail and approximation coefficients; then, they extracted the critical points with SIFT on the detail coefficients. Subsequently, they compared the feature vectors to identify which regions were falsified. Another recognized feature extractor is the Speeded Up Robust Features (SURF), which detects interest points and descriptors [9, 10]. A technique to detect copy-move forgeries based on SURF and KD-Tree for the comparison of multidimensional data was presented in [11]. Recently, many researchers applied deep learning methods in areas such as mechanics [12], medicine [13], and a solution for copy-move detection [14]. [15] proposed a deep learning approach with a transfer learning model that uses VGG-16 custom design convolutional neural network (CNN). A deep learning technique based on the CNN model with multi-scale-input and multi-stages of the convolutional layer was proposed [16]. This method used three phases: encoder block, decoder block to extract feature maps and classification. In the encoder block, the images are downsampled in multiple levels to extract features maps. In the decoder block, features maps are combined and upsampled until the output feature's dimension matches the input image's dimension. Finally, a sigmoid activation function was adopted as a classifier in the last phase. The authors in [17] suggested a dual branch CNN model with multi-scale input by choosing different kernel sizes. First, the images were resized and standardized on-the-fly; then, the images were passed to the CNN. The CNN architecture had two branches with standard input but different kernel sizes to extract distinct features maps. From the experimental result, the method was lightweight and performed high-grade prediction accuracy for the MIC-F2000 dataset.

1.2. Resampling detection

Several authors have focused their work on the identification of unnatural periodicities in digital images to detect resampling alteration. A method to detect the periodicity in images introduced by resampling and the compression of the JPEG format was implemented. To achieve this, they calculated the probability map of the image using the EM (Expectation-Maximization) algorithm [18]. Another method for resampling detection based on the EM algorithm and the probability mapping (m-map) of pixels in the frequency domain was used in [19]. This technique worked in the absence of any watermark or digital signature. In [20], a method capable of detecting traces of geometric transformation was proposed, using the periodic properties present in interpolated signals to detect whether the image has been modified or not.

1.3. Splicing detection

For this type of falsification, the paste operation alters the image’s general statistics by incorporating textures and borders that contrast with the original image [21, 22]. In recent years, some work carried out for splicing detection has focused on the extraction of the characteristics of an altered image and its original. The most relevant characteristics are used to train and validate a classifier, then the trained model is used to detect forgery on the image [23, 24]. In [25, 26], the authors presented a Local Binary Pattern (LBP) method due to its ability to represent textures with a low computational cost. LBP and the approximation coefficients of the DWT to extract the frequency characteristics were proposed in [27, 28], incorporating the histograms of the DWT detail coefficients to the vector support machine. Several authors have proposed LBP methods, such as LBP-DCT [29] and LBP-Enhanced [30]. In [31], an excellent method to highlight statistical changes from Markov's transition probability characteristics was introduced. Other authors have proposed combinations to improve results, such as Markov-DCT-DWT [32], Markov-DCT [33], Markov-QDCT [34], and Markov-Octonio DCT [35]. As copy-move detection, a deep learning approach is used for splicing detection. In [36], a two-branch CNN learns hierarchical representations from the input RGB color or grayscale test images and feeds a support vector machine as a classifier.

In this study, a hybrid algorithm for the detection of copy-move, resampling, and splicing forgery is proposed. The main contributions of this work are as follows:
• The proposed algorithm allows the identification of the three most common types of image forgery at a time. Most previous studies found in scientific literature only permitted the classification of some type of forgery per algorithm.
• Experiments were carried out with eight datasets, all with different resolutions and format features, to assess the robustness of the introduced hybrid algorithm.
• An improvement in the thresholding of the Markov-based preprocessing was achieved that allowed a significant reduction in the features used for the classifier.

2. Proposed algorithm

The general framework of the proposed algorithm methodology is presented in Figure 1. The proposed method is based on the outstanding work presented by E-Sayed [33] and on the improved thresholding method presented by Kumar et al. [37]. Unlike previous studies, we reduced the derivative matrices in space and frequency to 2 (vertical matrix and horizontal matrix), and the improved threshold was used to reduce the feature vector even more.

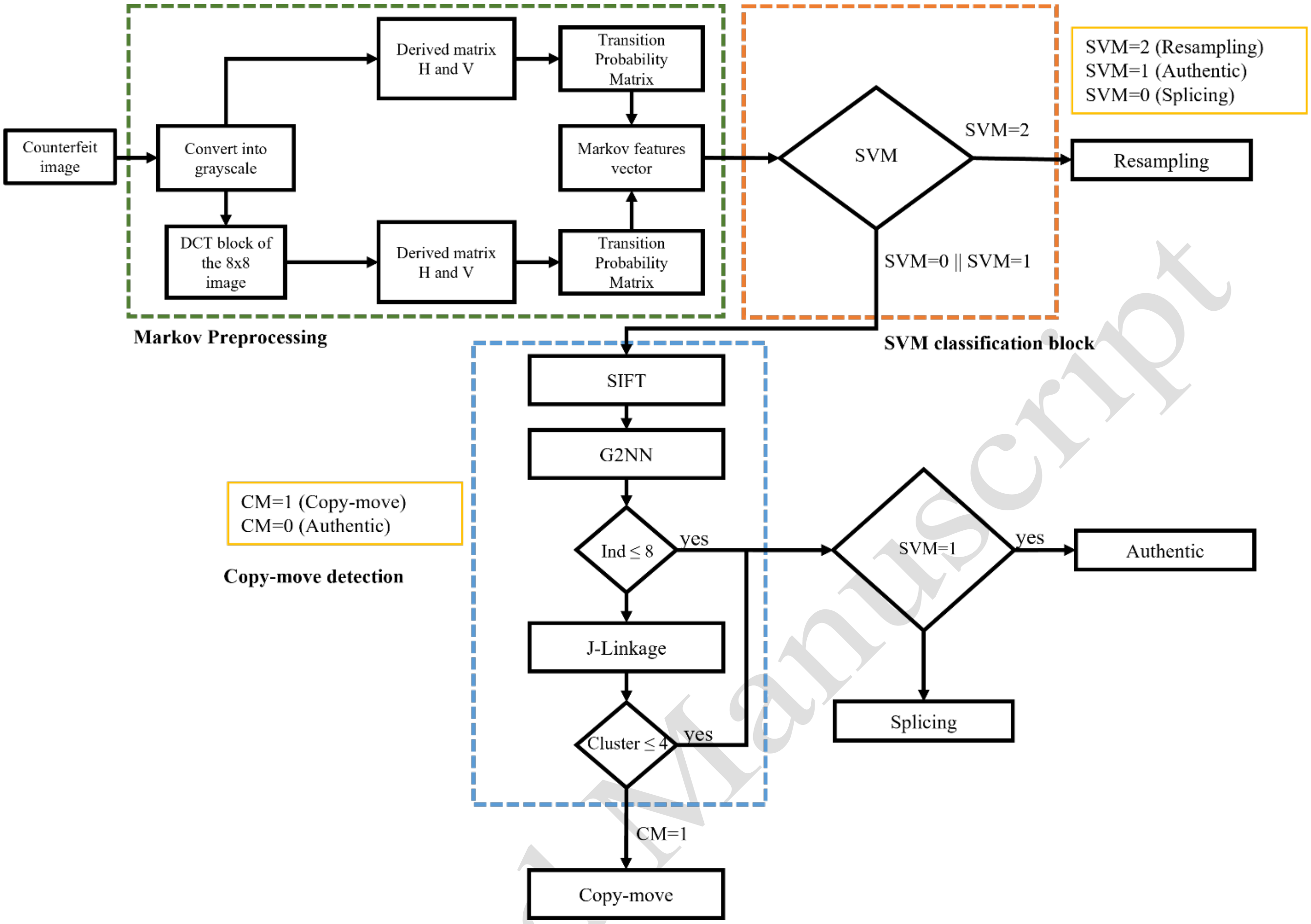


Figure 1 General flow diagram of the proposed algorithm.

This proposed algorithm is divided into two parts: copy-move detection and Markov preprocessing that is used for resampling detection. In the proposed method, the main contribution is the identification of the type of alteration of a digital image either by copy-move, resampling, or splicing.

2.1. Markov preprocessing

First, the image was converted to grayscale. Then, the horizontal and vertical derivatives of the image in space were calculated, after which the Markov process was carried out to obtain the characteristics in space. Subsequently, the image was divided into 8x8 non-overlapping blocks, and the DCT was applied to each block. After that, the horizontal and vertical derivatives were calculated for the resulting image. Next, the Markov process was performed to obtain the frequency characteristics. Finally, a vector of spatial and frequency characteristics was created to train a polynomial SVM.

DCT block

Since the altered image had changes in its local frequency distribution, it was possible to detect these changes with the DCT coefficients. First, the grayscale image was divided into 8×8 blocks without overlapping, and the DCT (version 2) was applied to each block individually. Finally, the values of the DCT coefficients were rounded, and the absolute value was calculated, as presented in Equation (1) and (2):

$$BDCT_{mk}(u,v) \ = DCT2(I_{mk}(i,j)) \qquad (1)$$

$$F(u, v) = |round(BDCT)| \qquad (2)$$

Where $BDCT_{mk}(u, v)$ represents the DCT of each 8×8 block and $F(u, v)$ is the complete image of the absolute rounded value.

Derivative matrices

It is possible to use an edge detector to observe statistical changes in the image because the operation of placing one image over another (splicing) introduces statistical alterations on the image edges [33]. To detect them, the horizontal and vertical derivatives of the image were calculated using Equation (3) and (4):

$$E_h(x, y) = I(x, y) - I(x + 1, y) \qquad (1)$$
$$1 \le x \le S_x - 1, 1 \le y \le S_y$$

$$E_v(x, y) = I(x, y) - I(x + 1, y) \qquad (2)$$
$$1 \le x \le S_x, 1 \le y \le S_y - 1$$

Where $\boldsymbol{I(x, y)}$ is the image, $\boldsymbol{S_x}$ and $\boldsymbol{S_y}$ are the image's dimensions. Likewise, the horizontal and vertical derivatives of the frequency image are calculated using Equation (5) and (6):

$$F_h(u, v) = F(u, v) - F(u + 1, v) \qquad (3)$$
$$1 \le u \le S_u - 1, 1 \le v \le S_v$$

$$F_v(u, v) = F(u, v) - F(u + 1, v) \qquad (4)$$
$$1 \le u \le S_u, 1 \le v \le S_v - 1$$

Where S_u and S_v are the image's dimensions in frequency. It should be noted that up to this point, all resulting matrices had positive integer values.

Thresholding

To reduce the feature vector dimensions and the algorithm complexity, a threshold T must be performed. If the value of the derivative matrix is higher than T, this value is replaced by T. Conversely, if the value of the matrix is less than –T, this value is replaced by –T, as shown in Equation (7):

$$T_x(u, v) = \begin{cases} +T & X(x, y) \ge +T \\ -T & X(x, y) \le -T \\ X(x, y) & otherwise \end{cases} \qquad (5)$$

Where $\boldsymbol{X(x, y)}$ are all derivative matrices $(\boldsymbol{E_h(x, y)}, \boldsymbol{E_v(x, y)}, \boldsymbol{F_h(u, v)}, \boldsymbol{F_v(u, v)})$. For this study, an adjustment was made in the range of thresholding based on the work presented by Kumar et al. [37], who stated that better results were obtained when using a range of $\boldsymbol{(i, j) = \in \{-T, -T + 2, \ldots, T + 2, T\}}$. Attention should also be given to the selection of an appropriate threshold, because a small T value may cause the transition probability matrix (TPM) not to be sensitive enough to detect alterations due to information loss. On the other hand, a high T value can cause the TPM to be extensive and therefore increase the feature vector and the complexity of the algorithm. In general, a threshold of T = 3 should be used to maintain a balance between sensitivity and complexity.

Probability transition matrix (TPM)

As previously mentioned, the splicing operation changes the correlation between the pixels of the original image. A random Markov process can be used to describe these correlation changes. In this case, the transition probability matrix was applied to each of the thresholded derivative matrices to characterize the Markov process. The horizontal and vertical transition probability matrices were calculated using Equation (8) and (9):

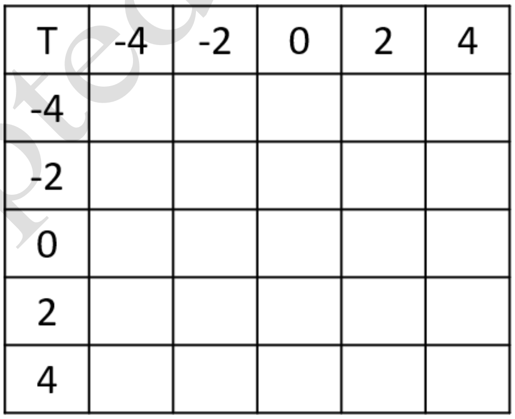
$$p\{T_h(u+1,v)=j|(T_h(u,v)=i)\}=\frac{(\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v}\delta(T_h(u,v)=i,T_h(u+1,v)=j))}{\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v}\delta(T_h(u,v)=i)} \qquad (6)$$

$$p\{T_v(u,v)=j|(T_v(u,v+1)\ =i)\}=\frac{(\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v}\delta(T_v(u,v)=i,T_v(u,v+1)=j))}{\sum_{u=1}^{S_u-2}\sum_{v=1}^{S_v}\delta(T_v(u,v)=i)} \qquad (7)$$

Where:

$$\delta(A=i,B=j)=\begin{cases}1 & A=i,B=j\\ 0 & otherwise\end{cases}$$
$$(i,j)=\in\{-T,-T+2,\dots,T+2,T\}$$

So, when applying (8) and (9) on horizontal and vertical derived matrices on space $(\boldsymbol{E_h(x,y)},\boldsymbol{E_h(x,y)})$ and the frequency $(\boldsymbol{F_h(u,v)},\boldsymbol{F_h(u,v)})$ a vector whose characteristics were $\boldsymbol{4\times(T+1)^2}$ was generated. Figure 2 shows the TPM with a set threshold T = 4. Finally, the TMP became a data vector that fed a support vector machine (SVM) to be trained.



T	-4	-2	0	2	4
-4					
-2					
0					
2					
4					

Figure 2 TPM result with a set threshold T = 4.

2.2. Copy-move detection

In case SVM has not detected any resampling alteration, the algorithm proceeds to perform copy-move detection. For this, the algorithm presented by Amerini et al. [38] was used to detect key points with the SIFT algorithm, and the library VLFeat 0.9.21 [39] was employed.

Key point detection with SIFT

The SIFT feature extraction algorithm on the altered image was used; this way, a descriptor vector of 128 features was obtained $S = \{s_i, \dots, s_n\}$.

Matching features through g2NN

Because two copied regions within the image have the same descriptors, the g2NN algorithm was used, which calculates the ratio between the Euclidean distance between a pair of candidate points and the next-nearest neighbor [40]. If the condition presented in Equation (10) is met, a pair of matching points is created $P = \{p_i, \dots, p_n\}$; where each p_i is each pair of (s_i, s_j).

Where d_i is the Euclidean distance between the candidate point and the point to verify, $d_i + 1$ is the Euclidean distance between the candidate point and the next nearest neighbor, and τ is a defined threshold that allows the rejection of pairs with very different descriptors. If two or more pairs of matching points are not found, the algorithm identifies the image as authentic.

$$\tau = \frac{d_i}{d_i+1} \tag{8}$$

J-linkage grouping

To identify the areas where the alteration occurs, the J-Linkage cluster, also called Hierarchical Agglomerative Clustering, (HAC) was used [41]. This procedure was performed with the pair of coinciding points coordinates but not with the Euclidean distance value. First, we proceeded with a random sampling of the pairs of coincident points $\boldsymbol{p}$ to generate m hypothesis of related transformations $T = \{T_1, \dots, T_m\}$. For each pair, a set of related transformations called "preference set vector": $PS = \{PS_1(p_i), \dots, PS_m(p_i)\}$ where each $PS_m(p_i)$ was defined in Equation (11).

$$PS_m(p_i) = \begin{cases} 1 & if\ \beta < 0.05 \\ 0 & otherwise \end{cases} \tag{9}$$

In other words, β is the distance between the T_m model and the pair of points p. If this value is less than 0.05, the pair of primary points and the pair of points of the copied region have similar transformations [40]. The preference set vector is used in the HAC algorithm to find the transformations of both the original points and the copied points. After establishing the preference set vector, these were assigned to a cluster; then, for each iteration, each pair of clusters was merged with the smaller distance in space. The preference set vector of a cluster was calculated as the intersection of the preference sets of the matched pair, and the distance between the paired clusters was determined with the Jaccard distance (J_δ) between the corresponding preference sets using Equation (12).

$$J_\delta(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{10}$$

The sets without similarities have a value of 1, while similar sets have a value of 0. According to these parameters, Amerini et al. [38] set a value of 1 as the cut-off grouping value. As a result, each cluster obtained at least one matching transformation among all its pairs. If more transformations are shared among all the elements of the cluster, they should be similar; for this reason, an estimation of the final transformation is determined by least-squares fitting. To dismiss outliers in cluster transformations, the fixed threshold N was used. Finally, if eight or more transformations (ind) are detected, or if there are four or more clusters (Cluster), the method alleges the image as falsified. Otherwise, it assumes that the image does not contain any modifications.

2.3. Algorithm adjustment

Given that the algorithm must identify different types of alterations, an adjustment was made for the final algorithm output. In the Markov preprocessing phase, an SVM was trained with authentic, resampling, and splicing images. If the SVM detected that the image had resampling, the algorithm indicated this. On the other hand, if the SVM identified the image as authentic or having splicing, the algorithm executed the copy-move detection process. If during this process, the copy-move block found the image to be free of copy-move alterations, the algorithm considered the image authentic. Table 1 presents the algorithm response depending on the SVM response and the copy-move algorithm. Because copy-move is a particular case of resampling, where there is no scaling or rotation, once the algorithm detects resampling, the output of the copy-move block is always resampling. Future work is expected to locate the region affected by copy-move or resampling.

TABLE 1 Truth table of algorithm output

SVM Result	Copy-move result	Algorithm output
Authentic (SVM=1)	Authentic (CM=0)	Authentic
Authentic (SVM=1)	Copy-move (CM=1)	Copy-move
Splicing (SVM=0)	Authentic (CM=0)	Splicing
Splicing (SVM=0)	Copy-move (CM=1)	Copy-move
Resampling (SVM=2)	Authentic (CM=0)	Resampling
Resampling (SVM=2)	Copy-move (CM=1)	Resampling

3. Methodology

In this section, we present a brief description of the experimentation process with which the proposed method was evaluated. The first part shows the datasets used for the algorithm evaluation. Next, the metrics used to evaluate the performance of the algorithms are presented. Then, an estimate of the optimal threshold T for the evaluation of the algorithm is shown. Finally, the methodology used to evaluate the proposed algorithm is presented. All tests were performed on MATLAB 2017b, on a 64-bit Dell computer, 8GB RAM with Windows 10, and an Intel Xenon processor.

3.1. Databases

In order to evaluate the algorithm, the most well-known forgery image evaluation datasets for splicing, resampling, and copy-move were selected. The datasets selected for evaluating splicing were CVLAB [42], CASIA V1 [43], Columbia [44] and Uncompressed Columbia [45]. Ardizzone-dataset [46], Coverage [47], MIC-F2000 [40] and CMFDdb-grip [48] datasets were used for the evaluation of copy-move and resampling. A selection of datasets rich in formats, sizes, and quantities of images with different types of alteration were used. Table 2 shows the main characteristics of all the datasets that were used to evaluate the proposed algorithm, such as format, number of original images, number of altered images, and resolution. Taking into consideration all the datasets used for the study, a total of 7100 images were evaluated, of which 3666 were unaltered, 791 had resampling, 2213 had splicing, and 430 had copy-move alterations.

TABLE 2 Databases

Database	Format	Total Images	Authentic	Altered images	Used images	Resolution
CASIA V1	jpg	1721	800	921	1721	384 × 256
COLUMBIA	bmp	1845	933	912	1845	128 × 128
UMCOMPRESSED COLUMBIA	tiff	363	183	180	363	757 × 568 and 1152 × 768
Ardizzone-dataset	bmp	1040	50	990	161	768 × 1024
COVERAGE	tif	200	100	100	200	386 × 431 and 768 × 1024
MIC-2000	jpg	2000	1300	700	2000	2048 × 1536
CMFDdb-grip	png	160	80	80	160	768 × 1024
CVLAB	jpg	650	200	450	650	720 × 480

3.2. CVLAB forgery database

For this study, a new database called CVLAB forgery database was created and published in [42]. This dataset contains 650 jpg format images that are 720 × 480. It contains 200 authentic images, 200 spliced images, and 250 copy-move images. The dataset has 50 images of animals, 40 of landscapes, 35 of flowers, 50 of buildings, 25 of people, and 50 of common objects. Figure 3 shows a sample of the images of the dataset.

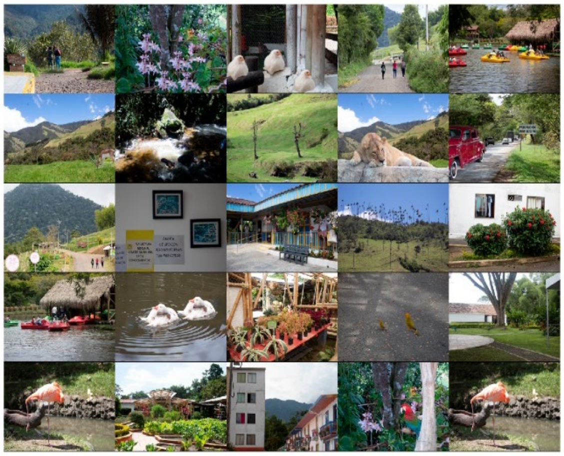


FIGURE 3 CVLAB Forgery Database images

3.3. Classifier performance

To demonstrate the detection performance, the classifier was evaluated by using the confusion matrix resulting in a 4 × 4 matrix that shows the classifier performance through the known values and predictions of the trained model. To determine the degree of reliability of the model, we calculated the accuracy (Acc), true positive rate (TPR), true negative rate (TNR), positive predictive value (PPV), the false positive rate (FPR), and the false-negative rate (FNR) using Equation (13), (14), (15), (16), (17) and (18) respectively. Alternative metrics that considered class imbalance were used, such as F1-score per class and weighted F1-score using Equation (19) and (20).

$$ACC = \frac{(TP+TN)}{(TP+TN+FN+FP)} \quad (11)$$

$$TPR = \frac{TP}{(TP+FN)} \quad (12)$$

$$TNR = \frac{TN}{(TN+FP)} \quad (13)$$

$$PPV = \frac{TP}{(TP+FP)} \quad (14)$$

$$FPR = \frac{FP}{(FP+TN)} \quad (15)$$

$$FNR = \frac{FN}{(FN+TP)} \quad (16)$$

$$F1 - score = \frac{2TP}{(2TP+FP+FN)} \quad (19)$$

$$weighted - F1 = \frac{\sum_{i=1}^{n} F1{-}score(i) * class_{inst}(i)}{total\_inst} \quad (20)$$

Where TP is the number of values predicted as positive that is indeed positive, TN is the number of values predicted as negative that is indeed negative, FP is the number of values predicted as positive that are negative, and FN is the number of values predicted as negative that are positive. $F1 - score(i)$ is F1-score of each class, $class_{inst}(i)$ are the total of instances of each class and $total\ inst$ are the total of instances used for classification.

3.4. Threshold T selection

As mentioned before, the selection of a threshold T is essential since this allows adjusting the robustness of the method together with the computational cost. At a higher value of T, a better sensitivity with a higher computational cost is obtained, and, conversely, a lower value of T loses sensitivity with a lower computational cost. Therefore, to identify the selection of the ideal threshold T, the splicing and resampling datasets mentioned above were trained with values of T equal to 4, 6, 8, and 10. Subsequently, the accuracy and precision of each model were estimated. All evaluations were performed with cross-validation of 10 folds. Finally, with the best value of the threshold T obtained, a model with an SVM with a Radial Basis Function (RBF) kernel was trained as a classifier to identify splicing and resampling alterations with the CASIA V1, Columbia, Columbia Uncompressed, CVLAB, and MICC-F2000 for a total of 7100 images. Only 161 random images of the Ardizzone-dataset were evaluated because this dataset only has 50 original images; therefore, imbalanced classes may affect the results of the evaluation.

4. Experimental results

This section presents the results of the SVM trained with different T thresholds and the results of the evaluation of the proposed algorithm.

4.1. Detection performance for SVM model

Table 3 shows the results of the evaluation carried out for the Resampling and Splicing detection with different T thresholds with multiple databases.

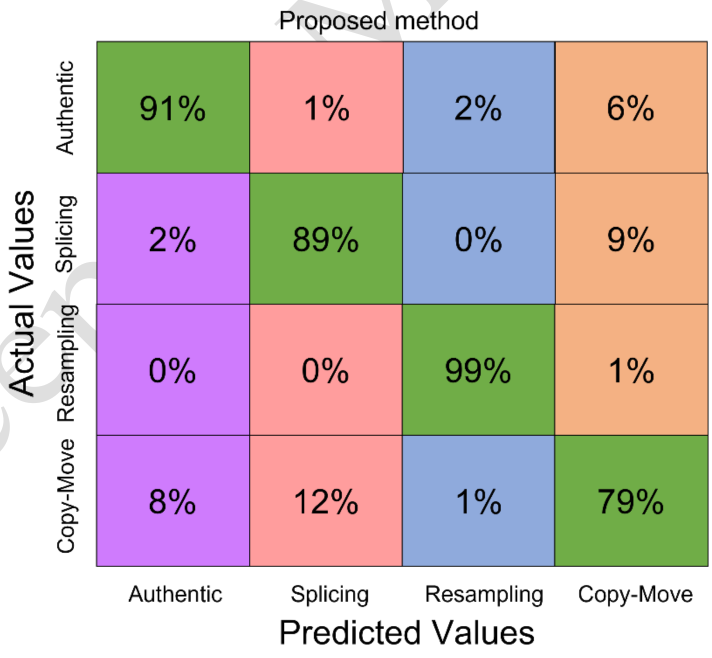
TABLE 3 Results of SVM model with different T value

THRESHOLD T/ CHARACTERISTICS	ACCURACY
T=4 (100)	0.90
T=6 (196)	0.90
T=8 (324)	0.90
T=10 (484)	0.89

As is evidenced in Table 3, a threshold of T = 4 obtained an accuracy of 90%, while T = 10 obtained an accuracy of 89%.

4.2. Performance of the proposed method

The confusion matrix that presents the result of evaluating all the datasets with the proposed methodology for the identification of splicing, resampling, and copy-move is presented in Fig. 4. For this evaluation, a trained model with an SVM was selected with an RBF kernel with a threshold of T = 4 since it obtained an outstanding result with a lower computational cost. The confusion matrix, shown below, was adjusted with the following color code: green represents true positives, purple represents false authentic positives, red represents false splicing positives, blue represents false resampling positives, and orange represents false copy-move positives.



	Authentic	Splicing	Resampling	Copy-Move
Authentic	91%	1%	2%	6%
Splicing	2%	89%	0%	9%
Resampling	0%	0%	99%	1%
Copy-Move	8%	12%	1%	79%

FIGURE 4 Confusion matrix of the proposed method.

As presented in Figure 4, the proposed method obtained a general accuracy of 91% and weighted-F1 of 92%. Where its true positive rate for authentic images was 91%, 89% for images altered by splicing, 99%

for images altered by resampling, and 79% for images altered by copy-move, Table 4 illustrates the results of the algorithm in greater detail.

TABLE 4 Performance of the proposed method

	Authentic	Splicing	Resampling	Copy-move
ACC	0.91	0.95	0.99	0.99
PPV	0.98	0.95	0.90	0.46
TPR	0.91	0.89	0.99	0.79
TNR	0.98	0.98	0.99	0.94
FPR	0.02	0.02	0.01	0.06
FNR	0.09	0.11	0.01	0.21

The high TPR values for each of the classes indicate that the algorithm is effective at identifying each type of alteration, with resampling identification being the highest with 0.99 and copy-move being the lowest with 0.79. The highest PPV value was 0.98, while the lowest accuracy was copy-move with 0.46. On the other hand, the highest accuracy corresponded to the resampling class with 0.99 and the lowest to authentic images with 0.91.

A comparison of our proposed algorithm with other techniques for the identification of various types of alterations [49, 50, 51] is presented in Table 5.

TABLE 5 Performance evaluation of different methods

	Prakash et. al [49]	Sharma and Ghanekar [50]	Hema Rajni [51]	Proposed
ACC	-	0.97	0.99	0.91
TPR Splicing	0.99	0.95	0.98	0.89
TNR Splicing	0.99	-	0.99	0.98
TPR Resampling	-	0.95		0.99
TNR Resampling	-	-		0.99
TPR Copy-move	0.93	0.99	0.98	0.79
TNR Copy-move	0.64	-	0.99	0.94
Evaluated Databases	3	4	3	8

Table 5 shows a 97% and 99% overall accuracy for the algorithm proposed by Sharma and Hema [50, 51], respectively, while the proposed algorithm's accuracy in this study is 91%. It is important to note that Prakash [49] only used three, Sharma, and Ghanekar [50] four, Hema Rajni [51] three, and the proposed technique eight datasets.

5. Discussion

In [49], splicing and copy-move alterations from Markov and the Zernike moment were identified. However, a resampling alteration can be considered a case of copy-move when there is scaling and rotation. This might have increased the computational cost of the results of the proposed algorithm since the estimation of both forgeries is more challenging. Additionally, the datasets used by the researchers have the same size and format; therefore, there are no considerable variations in the evaluation. [50] used a camera-based method in which they employed the Bayer matrix of the camera sensor to identify whether the image contained splicing, so they did not use a pixel-based methodology for the estimation. Additionally, the authors proposed a different equation from the one presented in Equation (14) to estimate the sensitivity of the copy-move alterations, which may vary the results of the comparison. [51] estimated the copy-move and resampling forgery using neural networks. The Markov process was used initially to feed a CNN that detected whether the image had been doctored. If it had been altered, the algorithm would transfer the image to a new CNN that classified whether it was a copy-move or splicing forgery. If the image had been altered using copy-move, the combined process of the circular harmonic transform (CHT) and the Zernike moments would be adopted to locate the copied region. Like Prakash, the author only classified copy-move and splicing alterations, while our proposed algorithm also recognized resampling forgery, making the challenge even greater. Further, it is well known that one of the main drawbacks of CNNs is their high computational cost due to their complex models and the limitations of datasets quantity for copy-move, resampling, and splicing due to deep learning requires much data, especially for training and testing [12]. In our proposal, we estimated the most remarkable types of forgery in the state of the art. Additionally, we evaluated our algorithm with eight different datasets to assess the robustness to the format and the resolution, in contrast with the authors previously introduced.

6. Conclusion

In this study, a hybrid algorithm was presented for the identification of copy-move, splicing, and resampling alterations from the Markov and SIFT process, which we call HA-MS (hybrid Algorithm - Markov and SIFT). Initially, a grayscale image was converted, then Markov features in space and frequency were extracted to train a model with an RBF-kernel SVM. The SVM was used to classify whether the image was altered by splicing or resampling or if the image was authentic. If it was suspected that the image was authentic or had copy-move alterations, SIFT was carried out to verify if the image had similar regions. The algorithm can identify several types of alterations with a general accuracy of 91. Some of the principal novelties of this proposal are the reduced number of features needed to carry out detection, in contrast to the method, and the algorithm robustness to variations in image format and resolution. [52].

REFERENCES

[1] H. Farid, "Image forgery detection," IEEE Signal processing magazine, vol. 26, no. 2, pp. 16–25, 2009.
[2] M. A. Qureshi and M. Deriche, "A bibliography of pixel-based blind image forgery detection techniques," Signal Processing: Image Communication, vol. 39, pp. 46–74, 2015.
[3] H. Farid, Photo forensics. MIT Press, 2016
[4] Y. Zhang, C. Zhao, Y. Pi, and S. Li, "Revealing image splicing forgery using local binary patterns of dct coefficients," in Communications, Signal Processing, and Systems. Springer, 2012, pp. 181-189.
[5] N. B. A. Warif, A. W. A. Wahab, M. Y. I. Idris, R. Ramli, R. Salleh, S. Shamshirband, and K.-K. R. Choo, "Copy-move forgery detection: survey, challenges and future directions," Journal of Network and Computer Applications, vol. 75, pp. 259–278, 2016.
[6] S. S. Mangat and H. Kaur, "Improved copy-move forgery detection in image by feature extraction with kpca and adaptive method," in 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 694–703, IEEE, 2016.

[7] F. Yang, J. Li, W. Lu, and J. Weng, "Copy-move forgery detection based on hybrid features," Engineering Applications of Artificial Intelligence, vol. 59, pp. 73–83, 2017.

[8] M. F. Hashmi, A. R. Hambarde, and A. G. Keskar, "Copy move forgery detection using dwt and sift features," in 2013 13th International conference on intellient systems design and applications, pp. 188–193, IEEE, 2013.

[9] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in European conference on computer vision, pp. 404–417, Springer, 2006.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," Computer vision and image understanding, vol. 110, no. 3, pp. 346–359, 2008.

[11] B. Shivakumar and S. S. Baboo, "Detection of region duplication forgery in digital images using surf," International Journal of Computer Science Issues (IJCSI), vol. 8, no. 4, p. 199, 2011.

[12] L. W. Hernández-González, D. A. Curra-Sosa, R. Pérez-Rodríguez, and P. D. Zambrano-Robledo, "Modeling Cutting Forces in High-Speed Turning using Artificial Neural Networks", TecnoL., vol. 24, no. 51, p. e1671, Apr. 2021. doi: 10.22430/22565337.1671.

[13] L. Mera-Jiménez and J. F. Ochoa-Gómez, "Convolutional Neural Network for the Classification of Independent Components of rs-fMRI", TecnoL., vol. 24, no. 50, p. e1626, Jan. 2021. doi: 10.22430/22565337.1626.

[14] A. B. Z. Abidin, H. B. A. Majid, A. B. A. Samah and H. B. Hashim, "Copy-Move Image Forgery Detection Using Deep Learning Methods: A Review," 2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS), 2019, pp. 1-6, doi: 10.1109/ICRIIS48246.2019.9073569.

[15] Y. Rodriguez-Ortega, D. Ballesteros and D. Renza, "Copy-Move Forgery Detection (CMFD) Using Deep Learning for Image and Video Forensics", Journal of Imaging, vol. 7, no. 3, p. 59, 2021. doi: 10.3390/jimaging7030059.

[16] A. Jaiswal and R. Srivastava, "Detection of Copy-Move Forgery in Digital Image Using Multi-scale, Multi-stage Deep Learning Model", Neural Processing Letters, 2021. doi: 10.1007/s11063-021-10620-9.

[17] 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, 2020. doi: 10.1049/ipr2.12051.

[18] S.-p. Li, Z. Han, Y.-z. Chen, B. Fu, C. Lu, and X. Yao, "Resampling forgery detection in jpeg-compressed images," in 2010 3rd International Congress on Image and Signal Processing, vol. 3, pp. 1166–1170, IEEE, 2010.

[19] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," IEEE Transactions on signal processing, vol. 53, no. 2, pp. 758–767, 2005.

[20] B. Mahdian and S. Saic, "On periodic properties of interpolation and their application to image authentication," in Third International Symposium on Information Assurance and Security, pp. 439–446, IEEE, 2007.

[21] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in Proceedings of the 9th workshop on Multimedia & security, pp. 51–62, ACM, 2007.

[22] Y. Zhang, C. Zhao, Y. Pi, and S. Li, "Revealing image splicing forgery using local binary patterns of dct coefficients," in Communications, Signal Processing, and Systems, pp. 181–189, Springer, 2012.

[23] M. F. Jwaid and T. N. Baraskar, "Study and analysis of copy-move & splicing image forgery detection techniques," in I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017 International Conference on, pp. 697–702, IEEE, 2017.

[24] N. K. Gill, R. Garg, and E. A. Doegar, "A review paper on digital image forgery detection techniques," in 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7, IEEE, 2017.

[25] A. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, and G. Bebis, "Splicing image forgery detection based on dct and local binary pattern," in Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE, pp. 253– 256, IEEE, 2013.

[26] A. Shah and E.-S. El-Alfy, "Image splicing forgery detection using dct coefficients with multi-scale lbp," in Computing Sciences and Engineering (ICCSE), 2018 International Conference on, pp. 1–6, IEEE, 2018.

[27] F. Hakimi, M. Hariri, and F. GharehBaghi, "Image splicing forgery detection using local binary pattern and discrete wavelet transform," in Knowledge-Based Engineering and Innovation (KBEI), 2015 2nd International Conference on, pp. 1074– 1077, IEEE, 2015.

[28] M. Kaur and S. Gupta, "A passive blind approach for image splicing detection based on dwt and lbp histograms," in International Symposium on Security in Computing and Communication, pp. 318–327, Springer, 2016.

[29] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, "Passive detection of image forgery using dct and local binary pattern," Signal, Image and Video Processing, vol. 11, no. 1, pp. 81–88, 2016.

[30] F. Hakimi and I. M. H. Zanjan, "Image-splicing forgery detection based on improved lbp and k-nearest neighbors algorithm," International Journal Of Electronics Information & Planning, 2015.

[31] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in Proceedings of the 9th workshop on Multimedia & security, pp. 51–62, ACM, 2007.

[32] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on markov features in dct and dwt domain," Pattern Recognition, vol. 45, no. 12, pp. 4292–4299, 2012.

[33] E.-S. M. El-Alfy and M. A. Qureshi, "Combining spatial and dct based markov features for enhanced blind detection of image splicing," Pattern Analysis and Applications, vol. 18, pp. 713–723, Aug 2015.

[34] C. Li, Q. Ma, L. Xiao, M. Li, and A. Zhang, "Image splicing detection based on markov features in qdct domain," Neurocomputing, vol. 228, pp. 29–36, 2017.

[35] H. Sheng, X. Shen, Y. Lyu, Z. Shi, and S. Ma, "Image splicing detection based on markov features in discrete octonion cosine transform domain," IET Image Processing, vol. 12, no. 10, pp. 1815–1823, 2018.

[36] Y. Rao, J. Ni and H. Zhao, "Deep Learning Local Descriptor for Image Splicing Detection and Localization," in IEEE Access, vol. 8, pp. 25611-25625, 2020, doi: 10.1109/ACCESS.2020.2970735.

[37] A. Kumar, C. S. Prakash, S. Maheshkar, and V. Maheshkar, "Markov feature extraction using enhanced threshold method for image splicing forgery detection," in Smart Innovations in Communication and Computational Sciences, pp. 17–27, Springer, 2019.

[38] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, L. Del Tongo, and G. Serra, "Copy-move forgery detection and localization by means of robust clustering with j-linkage," Signal Processing: Image Communication, vol. 28, no. 6, pp. 659–669, 2013.

[39] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in Proceedings of the 18th ACM international conference on Multimedia, pp. 1469–1472, ACM, 2010.

[40] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo and G. Serra, "A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery", IEEE Transactions on Information Forensics and Security, vol. 6, no. 3, pp. 1099-1110, 2011. doi: 10.1109/tifs.2011.2129512.

[41] R. Toldo and A. Fusiello, "Robust Multiple Structures Estimation with J-Linkage", Lecture Notes in Computer Science, pp. 537-547, 2008. doi: 10.1007/978-3-540-88682-2_41.

[42] J. Cortes Osorio, C. Lopez Robayo and N. Hernandez Betancourt, "Computer Vision and Machine Learning Lab.", Academia.utp.edu.co, 2020. [Online]. Available: http://academia.utp.edu.co/jacoper/forgery/. [Accessed: 21- Jan- 2020].

[43] J. Dong and W. Wang, "Casia image tampering detection evaluation database (caisa tide) v1.0." [Online]. Available: http://forensics.idealtest.org/].

[44] T.-T. Ng, S.-F. Chang, and Q. Sun, "A data set of authentic and spliced image blocks," Columbia University, ADVENT Technical Report, pp. 203–2004, 2004.

[45] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in 2006 IEEE International Conference on Multimedia and Expo, pp. 549–552, IEEE, 2006.

[46] E. Ardizzone, A. Bruno, and G. Mazzola, "Copy–move forgery detection by matching triangles of keypoints," IEEE Transactions on Information Forensics and Security, vol. 10, no. 10, pp. 2084–2094, 2015.

[47] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, “Coverage a novel database for copy-move forgery detection,” in 2016 IEEE International Conference on Image Processing (ICIP), pp. 161–165, IEEE, 2016.

[48] D. Cozzolino, G. Poggi, and L. Verdoliva, “Copy-move forgery detection based on patchmatch,” in 2014 IEEE International Conference on Image Processing (ICIP), pp. 5312–5316, IEEE, 2014.

[49] C. S. Prakash, A. Kumar, S. Maheshkar, and V. Maheshkar, “An integrated method of copy-move and splicing for image forgery detection,” Multimedia Tools and Applications, pp. 1–25, 2018.

[50] S. Sharma and U. Ghanekar, “A hybrid technique to discriminate natural images, computer generated graphics images, spliced, copy move tampered images and authentic images by using features and elm classifier,” Optik, vol. 172, pp. 470–483, 2018.

[51] Hema Rajini, “Image forgery identification using convolution neural network”. International Journal of Recent Technology and Engineering, 8 (1 Special Issue 4), pp. 311–320, 2019.

[52] C. Lopez, “Algoritmo híbrido para la identificación de falsificaciones de tipo de copy-move, splicing y resampling en imágenes digitales usando markov y sift”, M.S thesis, Universidad Tecnológica de Pereira, Pereira, Colombia, 2019.