

Color Image Splicing Localization Based On Block Classification Using Transition Probability Matrix

ESRA ODABAS YILDIRIM (✉ esra.odabas@atauni.edu.tr)

Ataturk University: Ataturk Universitesi <https://orcid.org/0000-0002-0936-1342>

Gul TAHAOGLU

Karadeniz Technical University: Karadeniz Teknik Universitesi

Guzin ULUTAS

Karadeniz Technical University: Karadeniz Teknik Universitesi

Beste USTUBIOGLU

Karadeniz Technical University: Karadeniz Teknik Universitesi

Vasif NABIYEV

Karadeniz Technical University: Karadeniz Teknik Universitesi

Research Article

Keywords: Image Splicing Detection, Image Splicing Localization, Transition Probability Matrices, Image Forgery Detection

Posted Date: August 8th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1901009/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Title: Color Image Splicing Localization Based On Block Classification Using Transition Probability Matrix

Authors:

Esra ODABAS YILDIRIM ^A

Gul TAHAOGLU^B

Guzin ULUTAS^B

Beste USTUBIOGLU^B

Vasif NABIYEV^B

^A :

Department of Software Engineering

Faculty of Engineering

Ataturk University

25240, Erzurum

^B :

Department of Computer Engineering

Karadeniz Technical University

61080, Trabzon

Corresponding Author:

Esra ODABAS YILDIRIM

esra.odabas@atauni.edu.tr, +90 505 8274685

Funding:

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) with Project No: 119E045

** Corresponding author. Mobile: +90 505 827 46 85;*

E-mail addresses: esra.odabas@atauni.edu.tr (E. Odabas Yildirim), gultahaoglu@ktu.edu.tr (G. Tahaoglu), guzin@ieee.org (G. Ulutas), bgencturk@ktu.edu.tr (B.Ustubioglu), vasif@ktu.edu.tr (V. Nabiyev)

COLOR IMAGE SPLICING LOCALIZATION BASED ON BLOCK CLASSIFICATION USING TRANSITION PROBABILITY MATRIX

Esra Odabas Yildirim^{1*}, Gul Tahaoglu², Guzin Ulutas², Beste Ustubioglu², Vasif Nabiye²

¹Department of Software Engineering, Faculty of Engineering, Ataturk University, Erzurum, Turkey

²Department of Computer Engineering, Faculty of Engineering, Karadeniz Technical University, Trabzon, Turkey

gultahaoglu@ktu.edu.tr, guzin@ieee.org, bgencturk@ktu.edu.tr

Abstract. With the increasing technology, digital images have become a widely used data type in crucial areas such as medical journalism and law. Since it is used in such important areas, it has become questionable whether digital images are original or not. Image splicing forgery is one of the most common forgery types applied to digital images. This work proposes a new image splicing detection and localization method. Our motivation is to reveal the boundaries of forgery by using statistical features of the image blocks. The proposed method has two main stages: training and localizing. In both phases, image blocks that contain edge information are used because the splicing operation causes some inconsistency on the edges. In the training stage, original blocks are selected from the regions that include original boundaries, and forged blocks are selected from the areas that contain splicing operation-induced edges. Transition probability matrices are calculated in eight directions to obtain the correlation of the borders between the neighbor blocks on original and splicing edges. These matrices are used as a feature for each block. The blocks are classified as authentic and spliced using SVM. A new post-processing step has been proposed to eliminate the false positives that may occur due to the presence of original regions that are likely to be detected as spliced edges in the image. The publicly available Columbia dataset has been used to show the effectiveness of the state-of-the-art and proposed method. The results indicate that the proposed method has performed well even under JPEG compression and Gaussian blurring attacks.

Keywords: Image Splicing Detection, Image Splicing Localization, Transition Probability Matrices, Image Forgery Detection.

1. Introduction

Thanks to technological developments, digital images are frequently created using cameras and smart mobile phones in daily life. In addition, they also are produced with special machines used in fields such as medicine and engineering, and their usage areas are becoming widespread. Digital images can be used when making an important decision and even as evidence in law. The advancement of technology has also enabled the easy use of image editing software and the creation of forged images without leaving any traces. For these reasons, the authenticity of the digital image has become questionable in recent years. Thus, proving the authenticity of digital images has also become an important research area under “Image forgery detection”. The methods to detect digital image forgeries are grouped into two categories: Passive (blind) and active (Lian and Kanellopoulos 2009). While blind techniques do not require any additional information to authenticate an image, active methods require prior embedded information called a watermark to authenticate the image. The need for extra details has made these methods disadvantageous (Rey and Dugelay 2002).

Moreover, any image without signature/watermark information may need authentication. Although passive methods are enough to use the statistical features extracted from the image, they do

not need any extra information or data (Kundur and Hatzinakos 1999). Passive forgery detection methods deal with two forgery operations; Copy-move forgery and image splicing forgery. Copy move forgery is created by copying a region from the image and then pasting it to another area of the same image. Image splicing forgery is more challenging to detect and is created by splicing a region copied from another image into the image. An example of image splicing forgery is given in Fig. 1. A part from image (a) is copied and pasted onto image (b) to obtain a forged image.

In literature, the image splicing detection methods can be divided into two groups according to system outputs. As seen in Table 1, one of them puts forward whether the input image is just forged or not. The other type of method also performs pixel-level localization of the duplicated region.

The works in the first group use image feature such as statistical and textural to classify them. Image features are evaluated in different frequency domains also. Muhammad et al. (2014) used Steerable Pyramid Transform (SPT) and Local Binary Pattern (LBP) to obtain textural features of the image. According to study, the input image has been decomposed different scale sub-bands using SPT and LBP histograms extracted from all those sub-bands. For the final feature vector, these LBP histograms are concatenated. Using SVM, forged and original images are classified—experiments for the method realized on CASIA v1.0, CASIA v2.0, and Color Columbia datasets. The technique has achieved an average 96.20% accuracy ratio.

Textural features are used by Shen et al. (2017) to detect forged images. They obtained features using the Discrete Cosine Transform (DCT) domain. Then they constructed Gray level co-occurrence matrices in the DCT domain and obtained textural features from those GLCMs. The method achieved 98.54% and 97.73 accuracy rates on the CASIA v1.0 and CASIA v2.0 datasets. LBP, Local Phase Quantization (LPQ), and Weber Local Descriptor (WLD) are used as texture descriptors for feature extracting from the images (Shen et al. 2017, Agarwal and Chand 2015, Hussain et al. 2013, Alahmadi et al. 2017). Agarwal et al. (2015) proposed a method based on Entropy Filter and LPQ. An entropy filter is applied to the image at various scales, and LPQ is applied to extract texture features. Histograms obtained from the LPQ on different scales are used as texture features. Hussain et al. (2013) used the multi-scale Weber local descriptor in the color channels of the YCbCr space and compared it with LBP. WLD gave better results than LBP, as reported in the study. Another study on extracting textural features used LBP by Alahmadi et al. (2017). The method divides the images into blocks, and LBP transformation is applied in each block. LBP transformed image is transformed into the DCT domain to track the changes in the local frequency distribution.

Markov Model is used by He et al. (2012) to obtain the image's statistical features in DCT and Discrete Wavelet transform (DWT) domain. The method tested on Columbia, and CASIA v2.0 datasets, 93.55% and 89.76% accuracy rates were achieved. Li et al. (2017) extracted Markov features on the Quaternion Discrete Cosine Transform (QDCT) domain and had a 92.38% accuracy ratio on the CASIA v2.0 dataset. Markov, based on another method proposed by Odabas Yildirim and Ulutas (2018), motivated by the studies presented by He et al. (2012) and Li et al. (2017), used a low-frequency subband from the DWT domain. Computational complexity is reduced via this method, and it achieved approximately 98.93%, 99.77%, and 97.24% accuracy rates on CASIA v1.0, CASIA v2.0, and Columbia (color) datasets by using SVM. Odabas Yildirim and Ulutas (2018) used Markov transition probability matrices on the high-frequency subbands of DCT in another study. Block DCT (BDCT) is applied to the YCbCr transformed image, and 16 high frequency (HF) DCT coefficients are selected by zigzag scanning for every block. Quantization is realized to clarify the importance of HF DCT coefficients. These 16 HF coefficients are grouped to form a matrix of size 4x4, and a four-directional Markov model is applied to these matrices to obtain statistical transitions on HF coefficients. The method used SVM for classification and tested on CASIA v1.0, CAISA v2.0, and Color Columbia datasets, and it gives approximately 99.98%, 99.89%, and 98.92% accuracy rates, respectively. Odabas Yildirim and Ulutas (2019) proposed an image forgery detection method using statistical and textural features on the SWT domain. Textural features are obtained from five Haralic

texture descriptors from GLCMs, and statistical features are obtained from the four directional Markov model. Statistical and textural features are concatenated to form the final feature vector. The method has 99.29% accuracy on CASIA v1.0, 99.58% on CAISA v2.0, and 97.99% on Color Columbia datasets.

The works in the second group localize the spliced region in the forged image. Some of these studies assume that the boundary between the spliced and the original region has been processed, such as median filtered (Bianchi and Piva 2012) or double JPEG compressed (Popescu and Farid 2005), or blurred (Bahrami et al. 2015), etc. This way, the localization problem has become detecting the specific operation locally. The noise-based methods are common studies in this group because patches from different images have generally different noise levels in spliced images. Zoran and Weiss (2009) proposed a statistical model for estimating noise variance and demonstrated its effectiveness in low-level noise images. The method assumes that kurtosis values in different filters and scales will not change due to scaling, and any changes in these values will be caused by additional noise. The algorithm proposed is the first kurtosis-based method that does not require knowing kurtosis values. The basis of the noise estimation step is the assumption that the original, unforged image has unchanged statistics. That is, it assumes that the kurtosis of the responses of the authentic images to different filters is an unknown constant and that the spliced region and the external noise added to the image will change the kurtosis values along with the scales. Pan et al. (2011) used the noise level estimation method proposed by Zoran and Weiss (2009) to local image blocks, and the technique clustered blocks via coarse to refined clustering to localize the spliced area. Mahdian and Saic (2009) estimated the noise level in each block with a median-based noise variance estimator from each image block (not overlapping) in the high-frequency subband after the wavelet transformation of the image. Then, these blocks were combined by examining the noise differences between neighboring blocks to create a homogeneous area. However, the method's main disadvantage is determining the correct threshold value. If the threshold value is not chosen carefully, the combined area cannot be distinguished from the image. Popescu and Farid (2014) used noise variance in the study to detect the spliced region. The second and fourth moments were calculated from each image block to estimate noise variance. The method requires knowing the kurtosis of the original signal.

Lyu et al. (2014) extended the noise level estimation method and localized the spliced area with pixel level. The noise-based methods perform better when the noise differences between the original and spliced region are significant—the weakness of these methods is having a negligible noise level between the original image and the spliced patch. In practice, however, this difference in spliced images is relatively tiny, and current noise-based methods provide less accurate performance in this case. Zeng et al. (2017) proposed a new approach to eliminate this performance gap and detect forged patches in the images with minor noise differences. The study suggested PCA-based localization. The image is divided into blocks, and the local noise level was estimated using PCA. The method used the K-means algorithm to cluster blocks as original and spliced via their noise level. Pun et al. (2016) used multi-level noise estimation. The study presents visually distinguishable results utilizing multi-scale analysis. Yao et al. (2016) proposed to estimate the noise level function (NLF) based on Taylor expansion. By exploring the relationship between NLF and the camera response function (CRF), the method fits the NLF curve under the CRF. It then formulates a Bayesian maximum a posteriori (MAP) framework to optimize the NLF estimation.

Zhang et al. (2019) suggested an image splicing detection technique to detect manipulations made without changing the content. First, the test image is divided into non-overlapping blocks using Simple Linear Iterative Clustering (SLIC). It then performs a block-level local noise estimate in the image and uses fuzzy c-averages aggregation to identify the spliced patch regions. It has higher

detection accuracy and robustness than existing methods, especially when the noise difference between the spliced region and the original image is small. Dua et al. (2020) proposed a technique in which features are obtained by modeling the quantified DCT coefficients with a Double Stochastic Model (DSM) parameter of the DSM for the feature vector to classify images as forged and original with the using SVM. The localization process is carried out in two approaches, depending on the type of forgery. The localization of splicing forgery is performed using the correlation of the inter-block relationships of the dequantized DCT coefficients. In the localization of copy-paste forgery, moments of phase-matching properties are used for each block of the forged image.

Another splicing localization method using noise inconsistency is proposed by Zeng et al. (2020). The study is based on the approach that, unlike the existing noise-based methods, the spliced patch will have different sensor noise due to different ISO settings in other machines. A weighted function is proposed to model the relationship between scene brightness and sensor noise level, independent of image content, to analyze the character of the sensor noise. First, the noise level of the test image is estimated locally by the Principal Component Analysis (PCA) based algorithm. Before grouping noise levels, the k-means weighting operation is applied to those levels. Wang et al. (2020) also used noise inconsistency to detect spliced regions. The Laplace operator was used to extract local noise in the image. The Fuzzy C-means (FCM) clustering algorithm and adaptive thresholding were used to obtain the suspicious regions. The image is divided into K non-overlapping blocks using the Simple Linear Iterative Clustering (SLIC) algorithm, and each block's Laplace noise is extracted. It has been reported that the method is resistant to post-processing.

Table 1. Different studies on image splicing in the literature

Method	Features	Classifier	Image authentication	Localization
Muhammed et al. (2014)	SPT-LBP	SVM	✓	—
Shen et al. (2017)	DCT-GLCM	SVM	✓	—
Agarwal et al. (2015)	Entropy Filter- LPQ	SVM	✓	—
Hussain et al. (2013)	WLD-LBP	SVM	✓	—
Alahmadi et al. (2017)	LBP-DCT	SVM	✓	—
He et al. (2012)	DCT-DWT-Markov	SVM	✓	—
Li et al. (2017)	QDCT-Markov	SVM	✓	—
Odabas Yildirim and Ulutas (2019)	SWT-Markov, SWT-TF	SVM	✓	—
Zoran and Weis (2009)	Noise estimation	—	—	✓
Pan et al. (2011)	Noise estimation (clustered blocks via coarse to fine)	—	—	✓
Mahdian and Saic (2009)	Noise estimation (median based)	—	—	✓
Poposecu and Farid (2014)	Noise variance (2nd and 4th moments)	—	—	✓
Lyu et al. (2014)	Noise level estimation	—	—	✓
Zeng et al. (2017)	Noise estimation (PCA based)	K-means		
Pun et al (2016)	Multi-level noise estimation	—	—	✓
Yao et al. (2016)	Noise estimation, NFL and CRF (based on Taylor expansion)	—	—	✓
Zhang et al. (2019)	Noise estimation (SLIC)	—	—	✓

Dua et al. (2020)	DCT-DSM	SVM	✓	✓
Zeng et al. (2020)	Noise estimation (sensor noise-PCA)	—	—	✓
Wang et al. (2020)	Noise estimation (Laplace operator-SLIC)	FCM	—	✓

In summary, the methods in the literature use several features or techniques to determine the spliced region, as seen in Table 1. Some studies only classify the images, and others localize the spliced region on forged images. The studies that classify the images use a classifier, and generally, SVM is chosen, as can be seen in Table 1. When looking at the second column in Table 1, studies on splicing localization are generally based on noise estimation can be seen. Image patches from different image sources have different noise levels, which is proved by the noise-based methods. However, differences in the noise level must be significant enough to detect the forgery. The most critical shortcoming of these methods is that they are not resistant to JPEG compression attacks because the compression eliminates the correlation of noise. We aim to propose a new approach to overcome this weakness. As seen in Table 1, the studies focus on either the image's authenticity or the spliced region's localization. The study in the current paper proposed to achieve both authenticity and localization. Using transitions of edges, a system is designed to classify image blocks as forged or original; with this, the boundary of the spliced region has been located via the forged blocks. After marking the boundary of the spliced area, the next step is marking the whole patch of the spliced region. Mainly, the proposed method consists of two phases; the marking phase and the localization phase. We selected original and forged blocks from the newly generated dataset in the marking stage. Original blocks are selected from original images, and forged blocks are obtained using the boundary of the spliced region located in forged images. Transition probability matrices (TPMs) are calculated from each block to obtain feature vectors. SVM is used for training the system with features obtained from those original and forged blocks. SVM is chosen as the classifier because SVM can handle non-linear solutions using different kernels. The advantages of SVM with other machine learning methods are: SVM supports both linear and non-linear solutions.

In contrast, linear and logistic regression support only linear solutions, and outliers are better-taken care of by SVM than KNN (k-nearest neighbor). Studies on the splicing forgery in the literature such as Muhammed et al. 2014, Shen et al. 2017, Agarwal and Chand 2015, Hussain et al. 2013, Alahmadi et al. 2017, He et al. 2012, Li et al. 2017, Odabas Yildirim and Ulutas 2019, are also used SVM to authenticate the images as can be seen in Table 1. So, in this study, the performance of the SVM in classifying the blocks is researched. After classifying image blocks as forged or original, a boundary of the spliced region has been located via the forged blocks. After determining the spliced blocks, the tamper localization step is generated. First, this step aims to eliminate blocks that are incorrectly marked as forged blocks even though they are original. For this purpose, the connected components by the forged blocks are determined, and small-sized components are eliminated. After that, the spliced region is represented via alpha shapes associated with the remaining connected components. Compared to the existing image splicing methods, the advantages of the proposed method are listed below:

- Most of the methods in the literature assume that the forgery boundary of the spliced image will be subject to distortions such as JPEG compression or blurring etc., and detect forgery by investigating these distortions. In the proposed method, it is possible to detect and localize spliced images that do not have these distortions since forgery can be detected without assuming these distortions.
- The input image is evaluated whether forged or not, and the spliced region is localized if the image is revealed to be forged. Thus, the proposed method gathers the primary operations of the referenced methods collected in two separate classes above in a single framework.
- Designing a learning-based approach with image blocks is the first in the literature. Using image blocks taken from different images makes the system more trustworthy.

- It is aimed to have higher detection accuracy even in JPEG compression and Gaussian blurring attacks.

The rest of the paper is organized as follows. The proposed method's details with subtitles (Marking Phase, Localization Phase) are given in Section 2, Section 3 offers Experimental results and discussion, and the paper concludes with Section 4.

2. Proposed Method

Image splicing operation causes some inconsistencies at the boundaries of the spliced region in a forged image. The method utilized from these inconsistencies to obtain the boundaries of the spliced regions. For this purpose, we proposed a learning-based approach to differentiate the boundaries of the spliced regions from the normal regions. The method consists of two phases: The marking phase and the localization phase, as seen in Fig. 2. While the first phase of the method marks the boundaries of the spliced region (the general diagram of this phase is given in Fig. 3 and Fig. 4), the second phase localizes exact spliced regions (the steps of this phase are given in Fig. 5). The details of these two phases will be given in the following sections.

2.1. Marking Phase

In this algorithm phase, the method aims to mark the boundary of the spliced region. For this purpose, the algorithm divides the image into overlapping blocks and uses extracted features from these blocks to mark them as spliced or normal. The method uses a learning-based approach to differentiate spliced blocks from normal ones. The marking phase consists of two steps. The system is trained using extracted features from the blocks of images in the dataset in the Training and Testing steps. The method marks the overlapping blocks of the test image as spliced or normal. Afterward, we call the blocks at the boundary of the spliced region “boundary block” and other blocks of the image will be called “normal block”.

Training step: We create a new dataset (available at: https://drive.google.com/drive/folders/1K9wTcFrSqT_hD57HS3a0HKcgWLzeDQM_?usp=sharing) to train the system to differentiate standard blocks from boundary blocks. The details about the creation of a new dataset will be given in section 3.1. This dataset contains spliced images with a sharp transition at the boundary of spliced regions. All images in the dataset are used to train the system.

Images are converted into YCbCr color space to obtain detailed information. Each color channel is evaluated separately, and the color channel Cr with the highest detection rate is chosen empirically by the method. The details of this experiment are given in Section 3.3. Cr channel of each image in the dataset is divided into overlapping blocks; some are normal, and some are boundary blocks. While boundary blocks (red-painted blocks in Fig. 3 indicate them) are used to introduce forged blocks to the system, regular blocks (green-painted blocks in Fig. 3 show them) are also used to introduce original blocks to the system. The training phase will be realized using the feature vectors extracted from regular and boundary blocks of all the images in the dataset. Fig.4 shows the general diagram of the feature extraction procedure.

Our method utilizes a feature extraction procedure during the training and testing phases. In the training phase, feature vectors are obtained from all overlapping blocks of images in the newly created dataset, and the dataset also contains mask images. Thus, we can classify the blocks as normal or boundary during the training phase. SVM is used to train the system, and the following operations are applied to each block to extract features.

Eight difference matrices through the horizontal, vertical, diagonal directions denoted by $F_{h1}, F_{h2}, F_{v1}, F_{v2}, F_{d1}, F_{d2}, F_{d3}, F_{d4}$ are constructed for each block. Equations given in (1) are used to construct difference matrices from each block. Let current block denoted by B .

$$\begin{aligned} F_{h1} &= B(u, v) - B(u + 1, v) & F_{h2} &= B(u, v) - B(u - 1, v) & F_{v1} &= B(u, v) - B(u, v + 1) \\ F_{v2} &= B(u, v) - B(u, v - 1) & F_{d1} &= B(u, v) - B(u - 1, v - 1) & F_{d2} &= B(u, v) - B(u - 1, v + 1) \\ F_{d3} &= B(u, v) - B(u + 1, v - 1) & F_{d4} &= B(u, v) - B(u + 1, v + 1) \end{aligned} \quad (1)$$

Then transition probability matrices denoted by $P_1 \dots P_8$ for eight difference matrices are calculated. These matrices give relative information about the current block with its neighboring blocks. Markov chains are used to model the states in the probabilistic system. A Markov chain's state Transition Probability matrix gives the probabilities of transitioning from one state to another (Pishro-Nik 2016). TPM provides the possibilities of transition from one state to another state. In the splicing localization problem, every block indicates a form, and its TPM is calculated with neighbored blocks (states). In general, Transition Probabilities are calculated using (2) in one-dimensional states.

$$T_{i,j} = P(S_{m+1} = j | S_m = i) \quad (2)$$

where $(S_m = i)$ denotes “process is in state i ” and the next state $(S_{m+1} = j)$ represents “process is in state j ”. Transition probability from state i to state j can be calculated using $P(S_{m+1} = j | S_m = i)$ and denoted by $T_{i,j}$. Transition probabilities for all states are generally given in a matrix form as in (3).

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \vdots & \vdots & & \vdots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{bmatrix}$$

$$p_{i,j} \geq 0 \text{ and } \sum_{k=1}^n p_{i,k} = \sum_{k=1}^n P(S_{m+1} = k | S_m = i) \quad (3)$$

In the transition probability matrix, when we are in state i , the next state should be one of the possible states. Therefore, when we add all possible values of j , we should get the value 1. So, the sum of any row values of the matrix must be 1. Each image block is evaluated as a state, and correlation with its neighbor blocks is calculated with TPM. Eight TPMs are calculated for each block using (4) as indicated above.

$$P_m = \frac{\sum_{x=1}^{S_x} \sum_{y=1}^{S_y} \begin{cases} \text{if } (F_n(x, y) = i \text{ and } F_n(x+k, y+l) = j), & 1 \\ \text{else} & 0 \end{cases}}{\sum_{x=1}^{S_x} \sum_{y=1}^{S_y} \begin{cases} \text{if } F_n(x, y) = i, & 1 \\ \text{else} & 0 \end{cases}} \quad (4)$$

where $S_x - S_y$ denote width and height of block, F_n denote the current difference matrices, $k, l \in \{+1, -1\}$ and $m \in \{1 \dots 8\}$.

After obtaining eight transition probability matrices $P_1 \dots P_8$ from the corresponding difference matrices ($F_{h1}, F_{h2}, F_{v1}, F_{v2}, F_{d1}, F_{d2}, F_{d3}, F_{d4}$), we concatenate them to construct corresponding feature vector for the current block.

Testing Step:

The method gets the test image and converts it into YCbCr space, and it also creates a mask image M , which consists of black pixels of the same size as the test image. Cr channel of the test image is divided into overlapping blocks. Feature extraction algorithm defined at the “Training step” is also applied to these blocks. And obtained feature vectors from them are given to SVM to mark their boundary or regular blocks at the mask image M . Mask image consists of black/white pixels, and white pixels denote boundary blocks. However, in the first phase of the algorithm (the Marking step), we mark the boundary of the spliced region. The second stage of the proposed method, called the “Localization Phase,” aims to localize the spliced area and eliminate false classification results.

2. 2. Localization Phase

After the abovementioned steps, white pixels at the mask image M represent the boundary block. However, some standard blocks on the mask image were also mistakenly labeled as boundary blocks due to false classification.

At this stage, blocks that are mistakenly marked as boundary blocks are eliminated, and then the spliced region is revealed. The block representation of this stage is given in Fig.5. As seen in the figure, our localization phase consists of two steps: Small components elimination, Boundary Determination, and Localization of the spliced area. We give details of these steps below.

Small components elimination

According to the experimental studies, it has been observed that the incorrect labels have a small size and are not continuous. On the other hand, it has been perceived that the correctly labeled boundary blocks are highly connected. Based on these observations, we proposed to use Connected Component Labeling (CCL) to connect district components (Haralick and Shapiro 1992). It is an algorithmic application of graph theory that subsets of connected components are uniquely labeled. In the proposed method, the dimensions of connected components are represented by S_i , i is the number of connected components. If S_i is smaller than the predefined edge threshold ∂ , i . component is eliminated. (We set $\partial = 75$ experimentally).

Boundary determination and localization the spliced area

The spliced region is determined in the next step through the remaining connected components. Similar to the importance of revealing the object's shape in image processing-based systems, it is important here to accurately determine the spliced region. Just as the bounding hull via the convex hull, the concave hull, alpha shape (-shape), plays an important role in achieving this goal in image processing. The convex hull of a set of points is a convex polygon with a minimum area that includes all these points. The convex hull does not always specify the points region accurately. However, the alpha shape represents the points region more accurately than the convex hull (Edelsbrunner 1992). The proposed method uses the boundary of the alpha shapes to determine the closed region to complete spliced edges. To reveal the spliced edge more accurately, the boundaries are determined with shrink factor (s , $[0,1]$), to shrink the inside of the border to enfold the points via alpha shape. (we set $s=0.7$ experimentally). This parameter determines which alpha shape with an alpha radius will be selected according to an ordered alpha radius. Once the boundary of forgery is determined, the region within this boundary is considered the spliced region. The pseudocode of this approach is also given in Algorithm 1.

Algorithm 1. Determine boundary (x, y, s) //boundary around (x, y)

1. Find alphashapes

2. Find critical alpha radius (alpha_crit) // it is needed to create a single region for alphashape
3. Extract all alphavalues (alphavalues) //to create unique shapes above this critical value
4. Select single alpha value according to shrink factor (s, [0, 1])

```
(1-s)*numberofelements(alphavalues>=alpha_crit)
```

5. Return the indices of the vertices of determined alphashape with specified alpha value

We give some experiments to show the effectiveness of the localization phase. Some example images, given in Fig. 6 and 7, were produced during the localization phase of the two spliced images from the Columbia dataset. (The image names are canon3_nikond70_sub_20 and canonxt_kodakdcs330_sub_12 respectively).

Fig.6(a) and 7(a) show mask images generated by the marking phase. We can perceive that falsely marked blocks in Fig. 6(a) are less than Fig. 7(a) because some regular blocks in Fig. 7(a) can accommodate abrupt color transition effects with neighboring blocks like boundary blocks. Thus, the classifier can label such blocks as boundary blocks. Images are given in Fig.6(b) and 7(b) are obtained after eliminating the false positives via Connected Component Labeling (Small components elimination phase). The pixels representing the spliced region shown in Fig.7 (b) are more continuous than in Fig.6 (b). Fig.6(c) and 7(c) show the determined boundary to represent the splice region, and (d)s show the determined final spliced areas.

The figure shows that the Localization phase eliminates falsely labeled blocks and connects distinct, marked blocks via CCL. And then, the method creates the region using marked blocks by boundary determination algorithm. Various experiments are realized to show the effectiveness of the proposed method in the following section.

3. Experimental Results and Discussion

In this section, we evaluated the proposed method experimentally. First, the proposed method is compared with the state-of-the-art image splicing localization methods, and numerical results are given with pixel-level detection accuracy. Second, we tested the robustness of the proposed method against JPEG compression. Matlab R2016b is used to develop the technique and to evaluate its performance. The hardware platform used during development and testing is a PC with Intel(R) Core (TM) i7, 2.67 GHz processor, and 8 GB RAM.

3.1. Datasets

In this study, two datasets are used. One of them is created by us (available at: https://drive.google.com/drive/folders/1K9wTcFrsqT_hD57HS3a0HKcgWLzeDQM_?usp=sharing) for the training stage and the second dataset is the Columbia dataset (Hsu and Chang 2006) which can be used commonly.

3.1.1. Design details of the generated forged image dataset

We created a small data set to obtain original and forged blocks for the training phase of the proposed method. The dataset contains 89 authentic images selected from different commonly used data sets

(Casia (Dong and Wang 2013) and Grip (Cozzolino et al. 2015)) and their forged versions created using GIMP (GNU image manipulation program). Original and spliced edges obtained from forged images that are available at (https://drive.google.com/drive/folders/1K9wTcFrSqT_hD57HS3a0HKcgWLzeDQM_?usp=sharing). We have created forged images so that the human eye can detect the spliced regions because the blocks at the border of the areas spliced must include sharp transitions to train the system more accurately. There are some examples of our original and forged images in Fig.8.

Forged regions can be detected by the human eyes, as can be seen in Fig.7. (a, b, e, f) are the forged images have sharply spliced regions. Their masks are given c, d, g, and h, respectively. The proposed system determines boundary blocks with the help of those forged image masks shown in Fig.9. For this purpose, the mask image is scanned from the top left to the bottom right. If a white pixel is encountered after the black pixel during scanning, the boundary block is created by taking the pixels at one distance from the white pixel. Then the block corresponding to the boundary block obtained from the mask image in the YCbCr representation of the same image is taken as a forged block. In Fig 9 (b), the red blocks show boundary blocks, while the orange blocks in the YCbCr image Fig. 9 (b) show splice blocks. Other edge blocks, except for the splice boundary blocks in the image, were taken as the original block. The classification model is trained with forged and original edge blocks that can be sharply distinguished from each other, and then all the blocks of the test image are questioned to label authentic or forged.

3. 1. 2. Columbia Dataset

We tested the proposed method on the Columbia uncompressed image splicing detecting evaluating dataset for the more realistic test (Hsu and Chang 2006). Four cameras (Canon G3, Canon 350D Rebel XT, Nikon D70, and Kodak DCS330) are used for original images in this dataset, and 180 spliced images are created with these authentic images by using Adobe Photoshop. Image splicing detection methods based on noise level inconsistency are commonly used in this dataset because images have been taken from four cameras and different cameras mean different noise levels.

3. 2. Evaluating Metrics

The evaluation metrics are used to quantify the classifier's performance. In this process, boundary blocks from input test images are classified as forged or authentic, so binary classification is used to label the blocks as genuine or forged. Spliced boundary blocks have been labeled "0", and the original blocks marked "1" by the method. SVM and radial basis function kernel are used to classify the boundary blocks. 10-fold cross-validation is used to get a more accurate estimation of the performance. During the training phase, 22716 spliced and 22631 original blocks were used. The training dataset is partitioned into ten equal subsets where the numbers of authentic and spliced blocks in each subset are the same. There are ten folds, and each fold is a test in which a subset is selected as the testing set; meanwhile, nine remaining subsets are combined with the training set. After classifying the sub-blocks, the post-processing operation is generated for the elimination of false-positive classification and painting the pixels in the spliced region. The final spliced part is revealed with the localization of the boundary block. The performance of the proposed method is analyzed at the pixel level. In this analysis, it is checked whether each pixel of the test image is correctly labeled. In the experimental study, we used TPR (True Positive Rate) and FPR (False Positive Rate) as defined in (5) for evaluating the performance of the proposed method for pixel-level evaluation. In this study, TPR is used for the probability of pixels correctly detected in the spliced region, and FPR is used for the possibility of pixels seen as spliced. However, they become original pixels (falsely detected). Destruel and others (2018) used F-measure and Matthews Correlation Coefficient (MCC) metrics for a full comparison of the popular studies for pixel-level evaluation on the Columbia dataset. We also

obtained our experimental results for this study and other popular reference studies using the same metrics for performance evaluation. These metrics are given in (6).

$$\begin{aligned} \text{TPR} &= \frac{TP}{TP+FN} \cdot 100, \quad \text{FPR} = \frac{FP}{FP+TN} \cdot 100 \\ \text{F-measure} &= \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \quad \text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{P \cdot (TP+FP) \cdot N \cdot (TN+FN)}}, \quad P = TP + FN, \quad N = TN + FP \end{aligned} \quad (5)$$

(6)

where TP (True Positive) is the number of correctly detected ‘the forged pixels’ in the spliced region, TN (True Negative) is the number of correctly detected ‘the original pixels’ in the non-spliced region, FP (False Positive) is the number of falsely labeled as ‘the original pixels’ in the spliced region and FN (False Negative) is the number of spliced pixels that are falsely detected as ‘the original pixels’.

Precision, Recall, and F-measure metrics were used in the literature to evaluate the performance at the image level, that is, whether the image is forged or not. We also used these metrics for a fair comparison (7).

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

Where, for image-level evaluation, TP represents the number of correctly detected forged images, FP represents mistakenly labeled forged images, and FN represents the forged images that are falsely detected as the original.

3.3. Experimental Analyses

The proposed method is used YCbCr color space, and each of the channels (Y, Cb, Cr) is evaluated separately. Firstly, the training stage has been processed on the Y channel, and the test image is samely converted to the Y channel, then the localization stage has been processed on this channel. Then Cb and Cr channels were evaluated, respectively. An example result of the test images can be seen in Fig.10. The first row shows the impact of the first image (Image 1), and the second row shows the result of another image (Image 2) on different color channels. From left to right, each row shows the Y, Cb, and Cr results, respectively. As can be seen in Fig.10 (a) and (d) Y channel has no good results in both detection accuracy and falsely detected block on the two images. Fig.10 (c) and Fig. 10 (f) show that the best results are obtained on Cr. The Cr channel has low falsely detected blocks and better continuity of boundary blocks. With those results, the Cr channel is selected for the other experimental results details below.

We tested the proposed method on the Columbia dataset (Ng et al. 2004) to evaluate the detection accuracy of spliced regions. We took four splicing images from Columbia and tested the detection accuracy rate of our method on these images and the methods proposed by Pan et al. (2011), Lyu et al. (2014), Zeng et al. (2017), Mahdian and Saic (2009) and Zhang et al. (2019). Fig.11 shows four examples of spliced images, and their detection results and pixel-level comparison are given in Fig. 12. We also compared the related methods to the whole Columbia dataset in Fig. 13. The proposed method was also tested against post-processing operations; JPEG-compression and Gaussian blur (Fig. 13 (b,c)).

In Fig.11, the first row shows the spliced images chosen from Columbia dataset, and the second row shows the detection results of the proposed method. The spliced regions are marked with a green, and pixels incorrectly labeled as spliced are marked red. The third row shows the detection results of the method proposed by Pan et al. (2011). While green grids are used to mark the spliced regions that

are detected correctly, red grids represent blocks that are detected falsely. The fourth row shows the detection result of the method proposed by Lyu et al. (2014). Spliced regions are marked green, and red represents the falsely detected blocks. The fifth row shows the detection result of the method proposed by Zeng et al. (2017), and the sixth row shows the detection of the results of the method proposed by Mahdian and Saic (2009), the spliced regions are marked in the green grid, and falsely detections are marked with a red grid. The seventh row shows the detection result of the method proposed by Zhang et al. (2019), and spliced regions are marked with white.

We aim to obtain higher TPR and lower FPR results than other works. Fig.11 shows that the proposed method detects spliced regions more accurately when compared to similar works. In Fig.11, images can be numbered from left to right, No.1-4. The method proposed by Lyu et al. (2014) detects some original blocks as spliced blocks in Image 1, and it also does not capture the spliced blocks correctly, as seen in the figure. The same situation also occurs for the method proposed by Mahdian and Saic (2009). The proposed method has high detection accuracy in Image 1. The method proposed by Pan et al. (2011) does not capture the spliced blocks in Image 2. The methods offered by Pan et al. (2011), Lyu et al. (2014), and Mahdian and Saic (2009) have falsely detected blocks in Image 3, and *the proposed method has the highest detection accuracy in this image.*

Detection results of Fig.11 are given in Fig.12 with TPR and FPR values. The proposed method is superior to other methods. Only the method proposed by Zhang et al. (2019) differs less from the TPR proposed in this paper on Image 1 and 2. The proposed method has low FPR generally only Zhang et al. (2019) have lower FPR in Image 1 and Image 4. Zeng et al. (2017) also have lower FPR than the proposed method but have lower TPR than the proposed methods, as seen in Fig. 12 (b). Having lower FPR is not meaningly. FPR and TPR results should be evaluated together. So the proposed method has good results among the other results in Fig. 12.

The robustness of the image splicing detection algorithm is an essential factor in evaluating the algorithm's performance. Splice methods suggested in the literature seem insufficient, especially against jpeg compression. For this reason, as another experimental study, we evaluated the robustness of the proposed approach to jpeg attacks. We compared it with related studies' overall Columbia dataset before and after post-processing (jpeg and blur attack). The average pixel-level comparison results are reported in Fig.13. As seen in Fig.13 (a), the proposed method has a higher TPR on the whole dataset. The false-positive rate of the proposed method is the lowest value among the related techniques. The method is robust against JPEG compression compared with the other methods, as can be seen in Fig.13 (b). The proposed method is also robust Gaussian blur, as can be seen in Fig. 13 (c). FPR values of the methods proposed by Zeng et al. (2017), Mahdian and Saic (2009), and Zhang et al. (2019) are lower than the proposed method, but these methods are not robust enough for JPEG compression. This is because compression destroys the noise correlation. Compression artifacts on the compressed images with QF=95 cannot be perceived. Correlation between spliced edges with their neighbor pixel is not changing so much, and edge information on the blocks, which are at the boundary of spliced regions, does not decrease gradually with such a quality factor value. The proposed method is also robust Gaussian blur, as shown in Fig. 13 (c). The technique has the highest TPR and the lowest FPR among the related processes, and the second high TPR is obtained by Zhang et al. (2019).

In another experimental study at the pixel level, a comparative analysis of the proposed method with other image splicing detection methods in the literature that have localization stage (Krawetz 2007, Ferrara et al. 2012, Wandji et al. 2014, Lyu et al. 2014, Cozzoloni et al. 2015, Salloum et al. 2017, Destrueel et al. 2018). Since the performances of the reference studies were presented using the F-measure and MCC metrics, this experiment was also evaluated using them. In Table 2, the average results on the Columbia dataset are given. The results of the reference studies were collected from the literature (Destrueel et al., 2018). As can be seen from the table, the highest average result obtained

with F-measure and MCC metrics belongs to the proposed method. According to these values obtained from experimental studies, Salloum et al. (2017) and the proposed method have the same average F-measure values. According to evaluation with the MCC metric, the proposed method has the highest performance in labeling forged pixels.

Table 2. The average results of pixel level classification

Methods	F-measure	MCC
Krawetz (2007)	0.47	0.23
Mahdian and Saic (2009)	0.57	0.41
Ferrara et al. (2012)	0.47	0.23
Wandji et al. (2014)	0.50	0.33
Lyu et al. (2014)	0.53	0.35
Cozzoloni et al. (2015)	0.45	0.21
Salloum et al. (2017)	0.61	0.48
Destruel et al. (2018)	0.5	0.39
Proposed method	0.61	0.55

Finally, the performance of the proposed method at the image level was evaluated to assess the performance of correct detection of the test image as forged or original. For this purpose, the forged images (number of 180 forged images) and authentic images (number of 180 original images) in the Columbia dataset are used. Destruel et al. (2018) performed the comparison with the method proposed by Hsu and Chang (2010) using Precision and Recall metrics. To make a fair comparison with these studies, we also used the Precision and Recall metrics, and also the F-measure values, the harmonic mean of these two metrics, were also evaluated. The average results of image level classification are given in Table 3. As can be seen from the table, the highest average result was obtained by the proposed method according to three metrics. According to these values obtained from experimental studies, it has been proven that the proposed method has the highest performance in labeling forged images.

Table 3. The average results of image level classification

Methods	Precision	Recall	F-measure
Hsu and Chang (2010)	0.70	0.70	0.70
Destruel and others (2018)	0.78	0.78	0.78
Proposed method	0.81	0.99	0.89

4. Conclusion

In this work, we have proposed to design a system that can detect image splicing clues. Based on the assumption that spliced edges have different features than the original edges, we exploited edge statistics via Transition Probability Matrices between the boundary of spliced regions and original edge blocks to reveal edge types. Firstly, in the training stage, we created a dataset for selecting boundary blocks of the spliced area and original blocks. 22716 spliced and 22631 original blocks are obtained from this dataset, and the system is trained with these blocks. The second stage of the method is the localization of the spliced region of the test image. The test image is divided into overlapping blocks, and each block is classified using SVM with obtained feature vectors. After labeling image blocks according to the results of SVM, spliced edges are revealed. The post-processing stage comes into play to eliminate pixels that are mistakenly labeled as spliced edges and then accurately display the spliced region. Experimental results showed that the designed system has a higher accuracy detection rate than the related methods and is robust against even under JPEG compression and Gaussian blur attacks.

5. References

- Agarwal S, Chand S (2015) Image forgery detection using multi scale entropy filter and local phase quantization. *Graph. Signal Process.*, 10, pp. 78–85. doi:10.5815/ijigsp.2015.10.08
- Alahmadi A, Hussain M, Aboalsamh H, Muhammad G, Bebis G. and Mathkour H, (2017) Passive detection of image forgery using DCT and local binary pattern, *Signal, Image Video Processing*, 11,1 81-88. doi:10.1007/s11760-016-0899-0
- Bahrami K, Kot AC, Li L, Li H (2015) Blurred image splicing localization by exposing blur type inconsistency, *IEEE Trans Inf Forensics Secur* 10(5):999–1009. doi: [10.1109/TIFS.2015.2394231](https://doi.org/10.1109/TIFS.2015.2394231)
- Bianchi T, Piva A, (2012) Image forgery localization via block-grained analysis of JPEG artifacts, *IEEE Trans Inf Forensics Secur* 7(3):1003–1017. doi: 10.1109/TIFS.2012.2187516
- Cozzolino D, Poggi G, Verdoliva L,(2015) Efficient dense-field copy-move forgery detection, *IEEE Trans. on Information Forensics and Security*, 10(11), 2284-2297. doi: [10.1109/TIFS.2015.2455334](https://doi.org/10.1109/TIFS.2015.2455334).
- Cozzolino D, Poggi G, and Verdoliva L, (2015) Splicebuster: A new blind image splicing detector, 2015 IEEE International Work- shop on Information Forensics and Security, Rome, Italy, November, pp. 1–6. doi: 10.1109/WIFS.2015.7368565
- Destruel C, Itier V, Strauss O and Puech W (2018) Color Noise-Based Feature for Splicing Detection and Localization, 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP), 2018, pp. 1-6, doi: 10.1109/MMSP.2018.8547093.
- Dong J, Wang W (2013) CASIA tampered image detection evaluation database, IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 422–426. doi: 10.1109/ChinaSIP.2013.6625374
- Dua S, Singh J, Parthasarathy H, (2020) Detection and localization of forgery using statistics of DCT and Fourier components, *Signal Processing:Image Communication* 82, 115778. doi: <https://doi.org/10.1016/j.image.2020.115778>
- Edelsbrunger, H. (1992). Weighted alpha shapes. Technical Report UIUCDCS-R-92-1760, Dept. Comput. Sci., Univ. Illinois, Urbana, IL.
- Ferrara P, Bianchi T, De Rosa A, and Piva A (2012) Image forgery localization via fine-grained analysis of CFA artifacts, *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 1566–1577, 10 2012. doi: 10.1109/TIFS.2012.2202227
- Haralick RM, Shapiro L, (1992) Connected component labeling, *Computer and Robot Vision*, Vol. 1, Addison-Wesley, New York.
- He Z, Lu W, Sun W et al (2012) Digital image splicing detection based on markov features in DCT and DWT domain, *Pattern Recognit.*, 45, (12), pp. 4292–4299. doi: [10.1016/j.patcog.2012.05.014](https://doi.org/10.1016/j.patcog.2012.05.014)
- Hsu YF and Chang SF (2006). Detecting image splicing using geometry invariants and camera characteristics consistency. In 2006 IEEE International Conference on Multimedia and Expo (pp. 549-552). IEEE. doi: [10.1109/ICME.2006.262447](https://doi.org/10.1109/ICME.2006.262447)
- Hsu YF and Chang SF, “Camera response functions for image forensics: an automatic algorithm for splicing detection,” *IEEE Transac- tions on Information Forensics and Security*, vol. 5, no. 4, pp. 816–825, December 2010.

Hussain M, Muhammad G, Saleh SQ, Mirza AM and Bebis G, (2013) Image forgery detection using multi-resolution Weber local descriptors, Eurocon, July, IEEE,1570-1577. doi: <https://doi.org/10.1007/s12530-019-09309-1>.

Krawetz N (2007) A picture's worth: Digital image analysis and forensics, in Black Hat Briefings, USA, November, pp. 1–31, Online article on: <http://www.hackerfactor.com/papers/bh-usa-07-krawetz-wp.pdf>.

Kundur, D. and Hatzinakos, D. (1999) Digital Watermarking for Telltale Tamper Proofing and Authentication. In: Proceedings of the IEEE, 87, 1167-1180

Li C, Ma Q, Xiao L, Li M, Zhang A (2017) Image splicing detection based on Markov features in QDCT Domain. Neurocomputing, 228, pp. 29–36. doi: [10.1016/j.neucom.2016.04.068](https://doi.org/10.1016/j.neucom.2016.04.068)

Lian, S., Kanellopoulos D. 2009. "Recent advances in multimedia information system security", Informatica, 33, 3–24.

Lyu S, Pan X, Zhang X, (2014) Exposing region splicing forgeries with blind local noise estimation, Int J Comput Vis 110(2):202–221. doi: <https://doi.org/10.1007/s11263-013-0688-y>

Mahdian B, Saic S, (2009) Using noise inconsistencies for blind image forensics, Image Vision Computing, 27,10,1497-1503. doi: <https://doi.org/10.1016/j.imavis.2009.02.001>

Muhammad G, Al-Hammadi MH, Hussain M, Bebis G (2014) Image forgery detection using steerable pyramid transform and local binary pattern" Mach. Vis. Appl., 25 (4) pp. 985–995. doi: [10.1007/s00138-013-0547-4](https://doi.org/10.1007/s00138-013-0547-4)

Odabas Yildirim E, Ulutas G (2018) Image splicing detection with DWT domain extended Markov features, 26th Signal Processing and Communications Applications Conference (SIU), Izmir, pp.1-4. doi: [10.1109/SIU.2018.8404325](https://doi.org/10.1109/SIU.2018.8404325).

Odabas Yildirim E, Ulutas G (2018) Markov-Based Image Splicing Detection in the DCT High Frequency Region, International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, pp. 1-4. doi: [10.1109/idap.2018.8620870](https://doi.org/10.1109/idap.2018.8620870)

Odabas Yildirim E, Ulutas G (2019) Augmented Features to Detect Image Splicing on SWT domain, Expert Systems with Applications, Volume 131, pp 81-93. doi: [10.1016/j.eswa.2019.04.036](https://doi.org/10.1016/j.eswa.2019.04.036)

Pan X, Zhang X, Lyu, S, (2011) Exposing image forgery with blind noise estimation, In the 13th ACM Workshop on Multimedia and Security, pp. 15–20. doi:

Pishro-Nik H, (2016) Introduction to probability, statistics, and random processes.

Popescu C, Farid H, (2004) Statistical tools for digital forensics, in 6th International Workshop on Information Hiding. Toronto, Canada, pp. 128–147.

Popescu C, Farid H, (2005) Exposing digital forgeries by detecting traces of resampling. IEEE Trans Signal Process 53(2):758–767. doi: <https://doi.org/10.1145/2037252.2037256>

Pun C M, Liu B, Yuan X C (2016) Multi-scale noise estimation for image splicing forgery detection. Academic press, Inc 38 (C) :195-206 . doi: <https://doi.org/10.1016/j.jvcir.2016.03.005>

Rey, C., Dugelay, J. L. 2002. "A survey of watermarking algorithms for image authentication", EURASIP Journal on Applied Signal Processing, 613–621

Salloum R, Ren Y, and Kuo C-C-J (2017) Image splicing localization using a multi-task fully convolutional network (MFCN), ArXiv e-prints, September. doi: [10.1016/j.jvcir.2018.01.010](https://doi.org/10.1016/j.jvcir.2018.01.010)

Shen X, Shi Z and Chen H, (2016) Splicing Image Forgery Detection Using Textural Features Based on the Gray Level Co-occurrence Matrices, IET Image Process, 11, (1), pp. 44–53. doi: [10.1049/iet-ipr.2016.0238](https://doi.org/10.1049/iet-ipr.2016.0238).

Wandji NND, Sun X, and Fah KM (2014) A survey of partition- based techniques for copy-move forgery detection, The Scientific World Journal, vol. 2014, pp. 1–13. doi: <https://doi.org/10.1155/2014/975456>

Wang X, Zhang Q, Jiang C, Zhang Y, (2020) Coarse-to-fine Grained Image Splicing Localization Method Based on Noise Level Inconsistency, In 2020 International Conference on Computing, Networking and Communications (ICNC) February, IEEE (pp. 79-83). doi: [10.1109/ICNC47757.2020.9049720](https://doi.org/10.1109/ICNC47757.2020.9049720)

Yao H, Wang S, Zhang X, Qin C, Wang J (2017) Detecting image splicing based on noise level inconsistency. Multimed Tools Appl 6(10):1–23. doi: [10.1007/s11042-016-3660-3](https://doi.org/10.1007/s11042-016-3660-3).

Zeng H, Peng A, Lin X, (2020) Exposing image splicing with inconsistent sensor noise levels, Multimedia Tools and Applications, 1-16. doi: <https://doi.org/10.1007/s11042-020-09280-z>.

Zeng, H., Zhan, Y., Kang, X. et al. (2016) Image splicing localization using PCA-based noise level estimation. Multimed Tools Appl 76, 4783–4799. <https://doi.org/10.1007/s11042-016-3712-8>

Zhang D, Wang X, Zhang M, Hu J, (2019) Image splicing localization using noise distribution characteristic, Multimed Tools Applications 78, 22223–22247. doi: [10.1007/s11042-019-7408-8](https://doi.org/10.1007/s11042-019-7408-8).

Zoran D, Weiss Y, (2009) Scale invariance and noise in nature image, In Proc. of IEEE International Conference on Computer Vision, pp. 2209–2216. doi: [10.1109/ICCV.2009.5459476](https://doi.org/10.1109/ICCV.2009.5459476)

Declarations

Funding This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) with Project No: 119E045

Conflicts of interest/Competing interests The authors declare that this article is original, has not been published before, is not currently being considered for publication elsewhere and there is no conflict of interest regarding the publication. The authors confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. The authors further confirm that the order of authors listed in the manuscript has been approved by all of them.

Availability of data and material The datasets generated during and analysed during the current study are available in the [dataset created by us and Columbia dataset] repository, [https://drive.google.com/drive/folders/1K9wTcFrSqT_hD57HS3a0HKcgWLzeDQM_?usp=sharing , <https://www.ee.columbia.edu/ln/dvmm/downloads/authsplcuncmp/>].

Code availability Not applicable

Ethics approval Not applicable

Consent to participate Not applicable

Consent for publication Not applicable

Figures

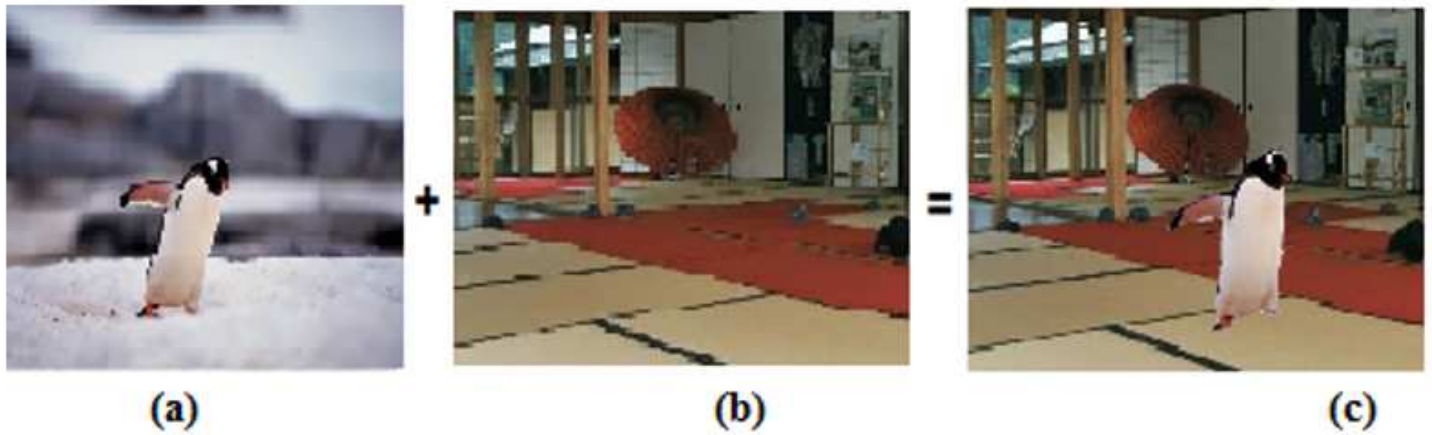


Figure 1

An Example for Image Splicing (a, b) original images (c) spliced image

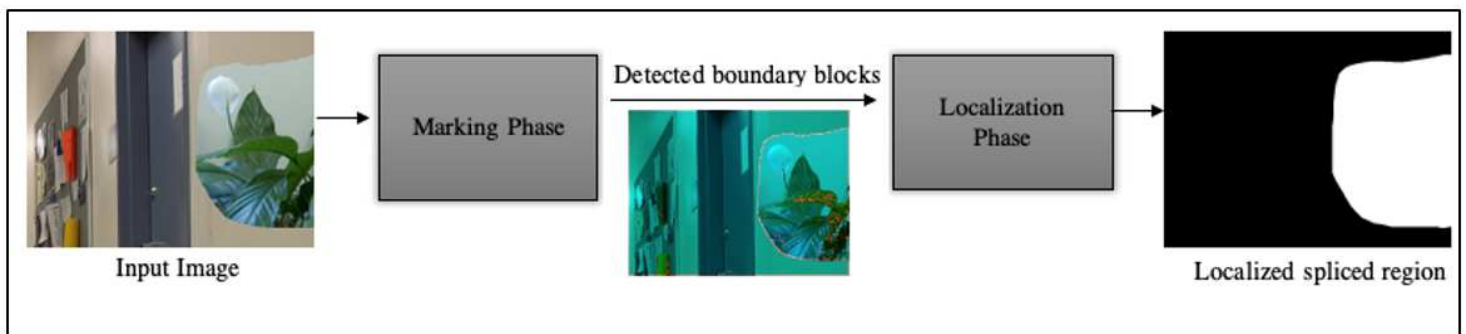


Figure 2

Main steps of the proposed method

Figure 3

The flowchart of marking phase

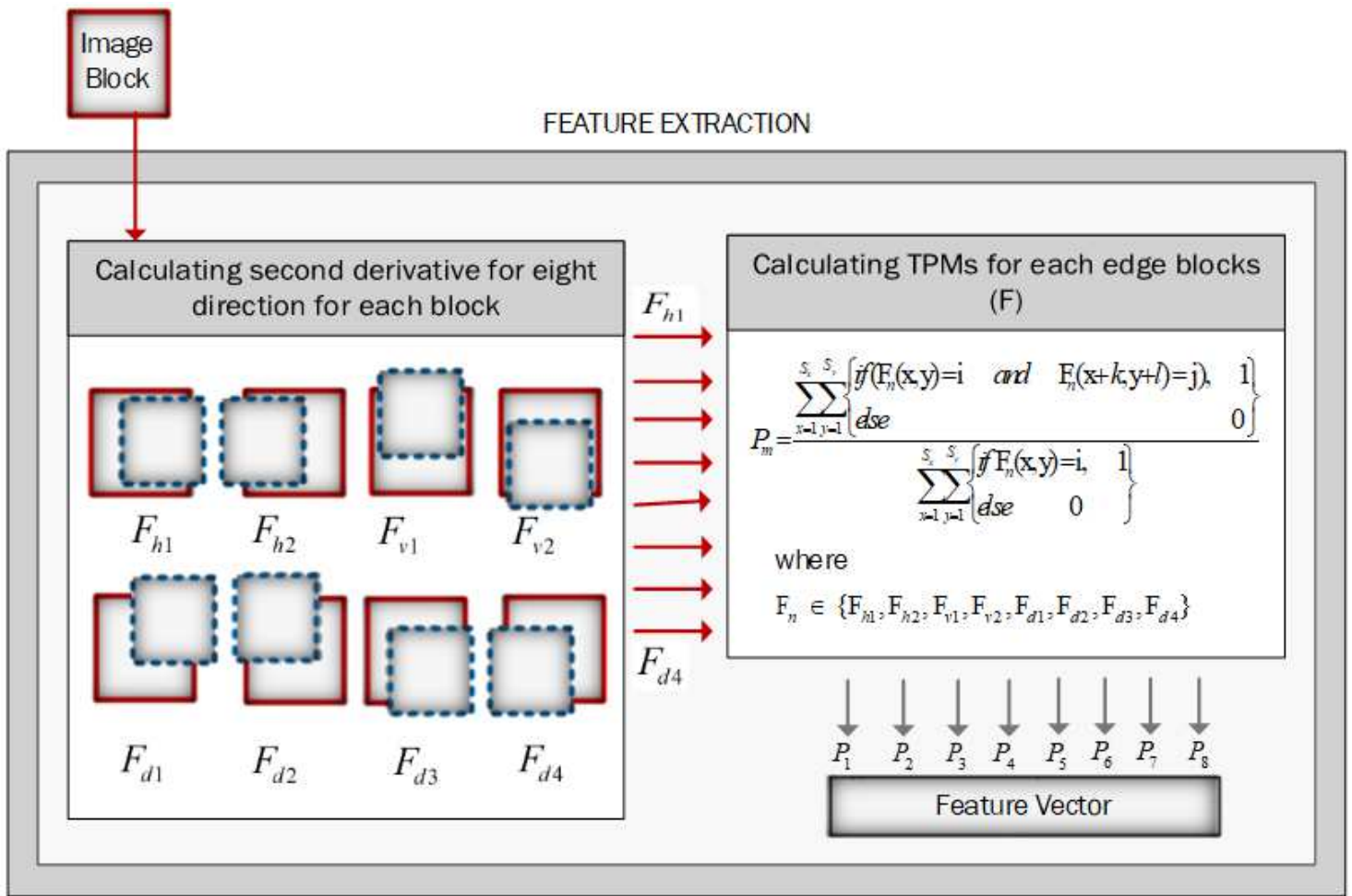


Figure 4

Detail of feature extraction procedure in marking phase

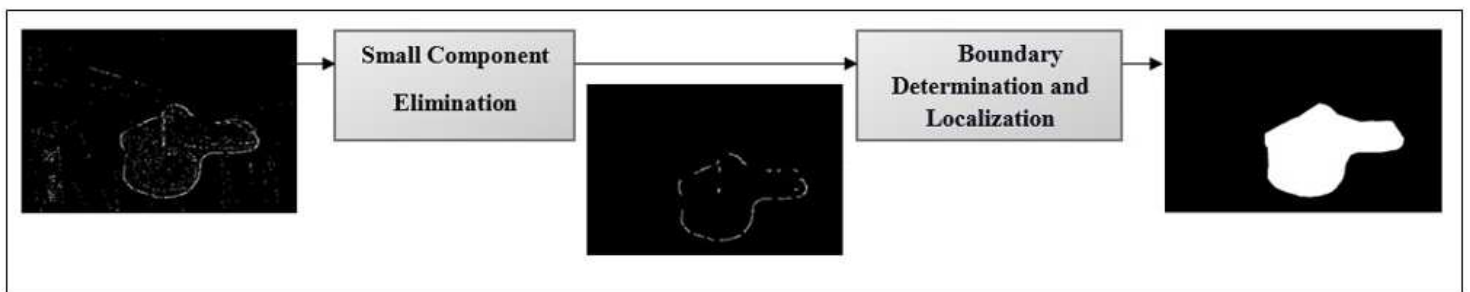
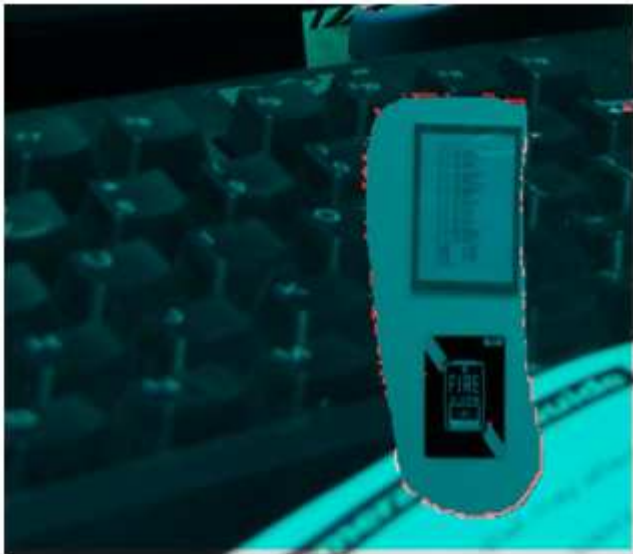
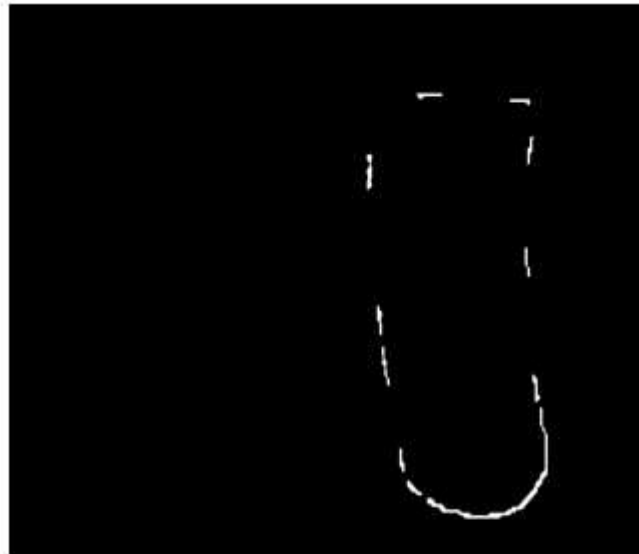


Figure 5

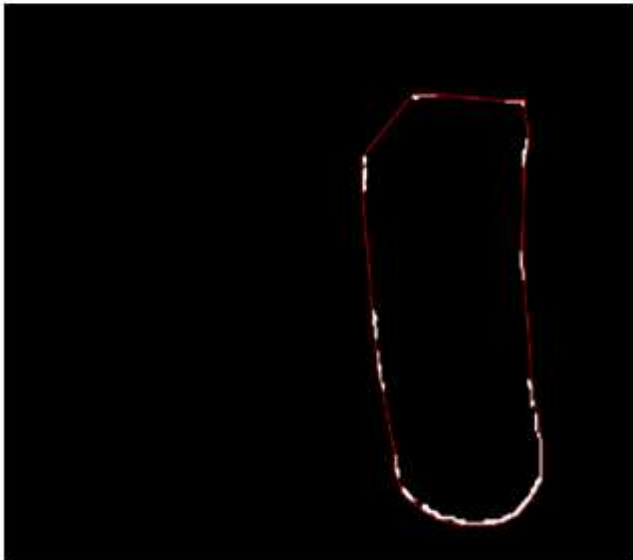
The steps of the localization phase



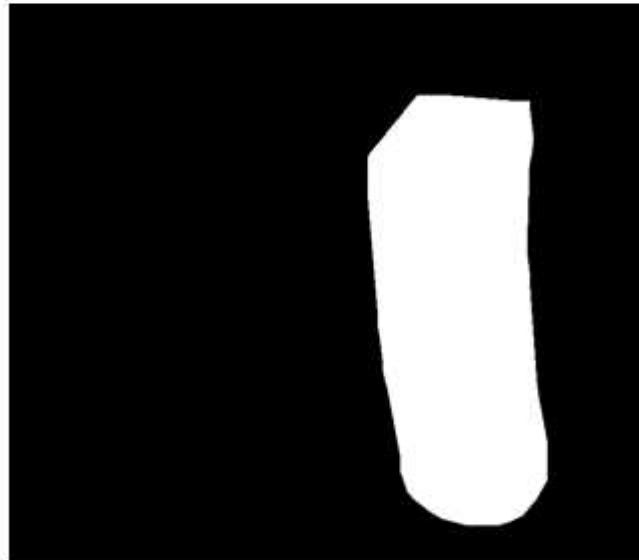
(a)



(b)



(c)



(d)

Figure 6

Obtained images of each step of the localization phase (Image name: canong3_nikond70_sub_20) (a) Labeled forged blocks (b) After small component elimination (c) Determined boundary (d) Final output

Figure 7

Obtained images of each step of the localization (Image name: canonxt_kodakdcs330_sub_12) (a) Labeled forged blocks (b) After small component elimination (c) Determined boundary (d) Final output

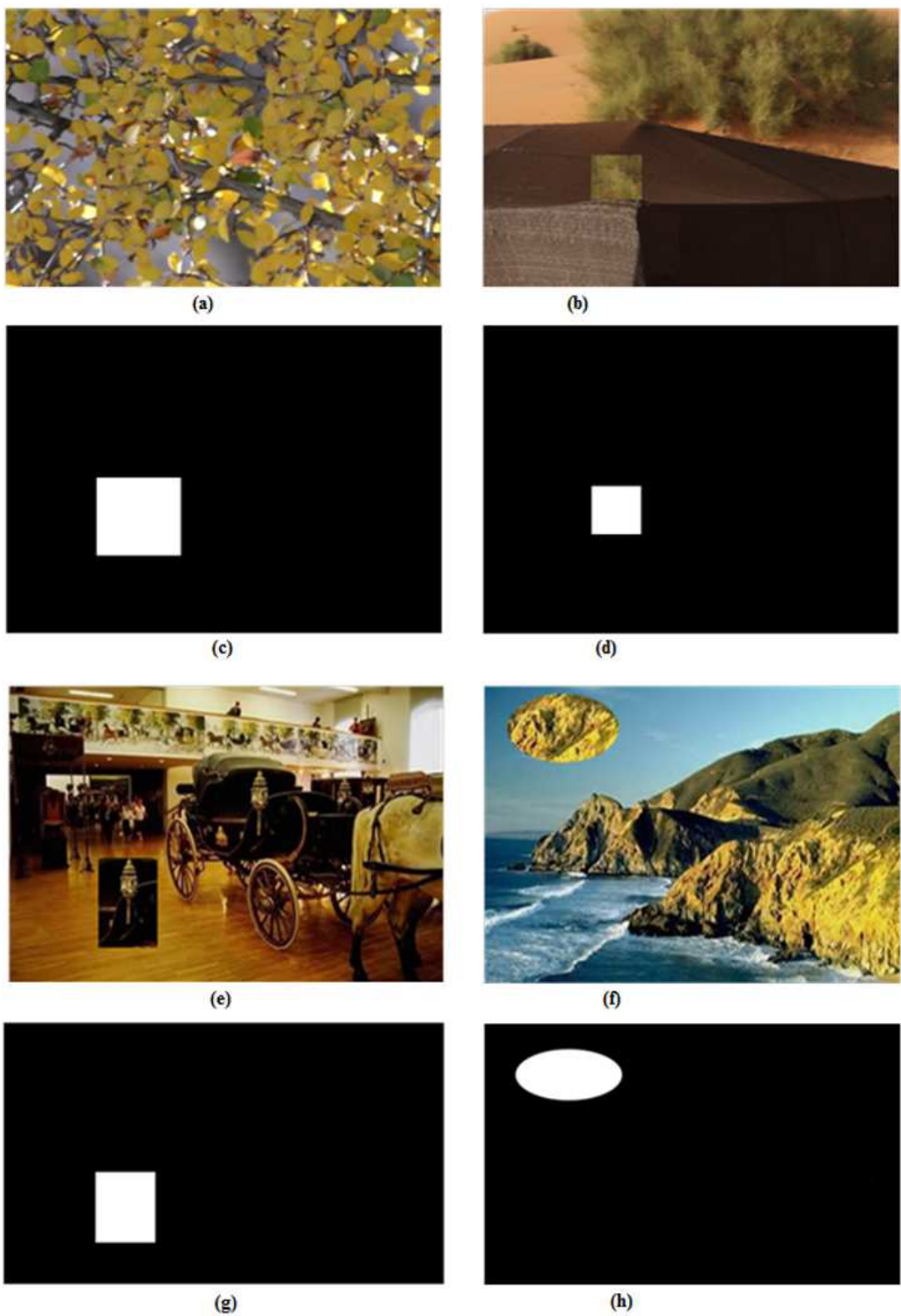


Figure 8

Examples of forged images in created data set. (a, b, e, f) Forged images, (c, d, g, h) Masks respectively.

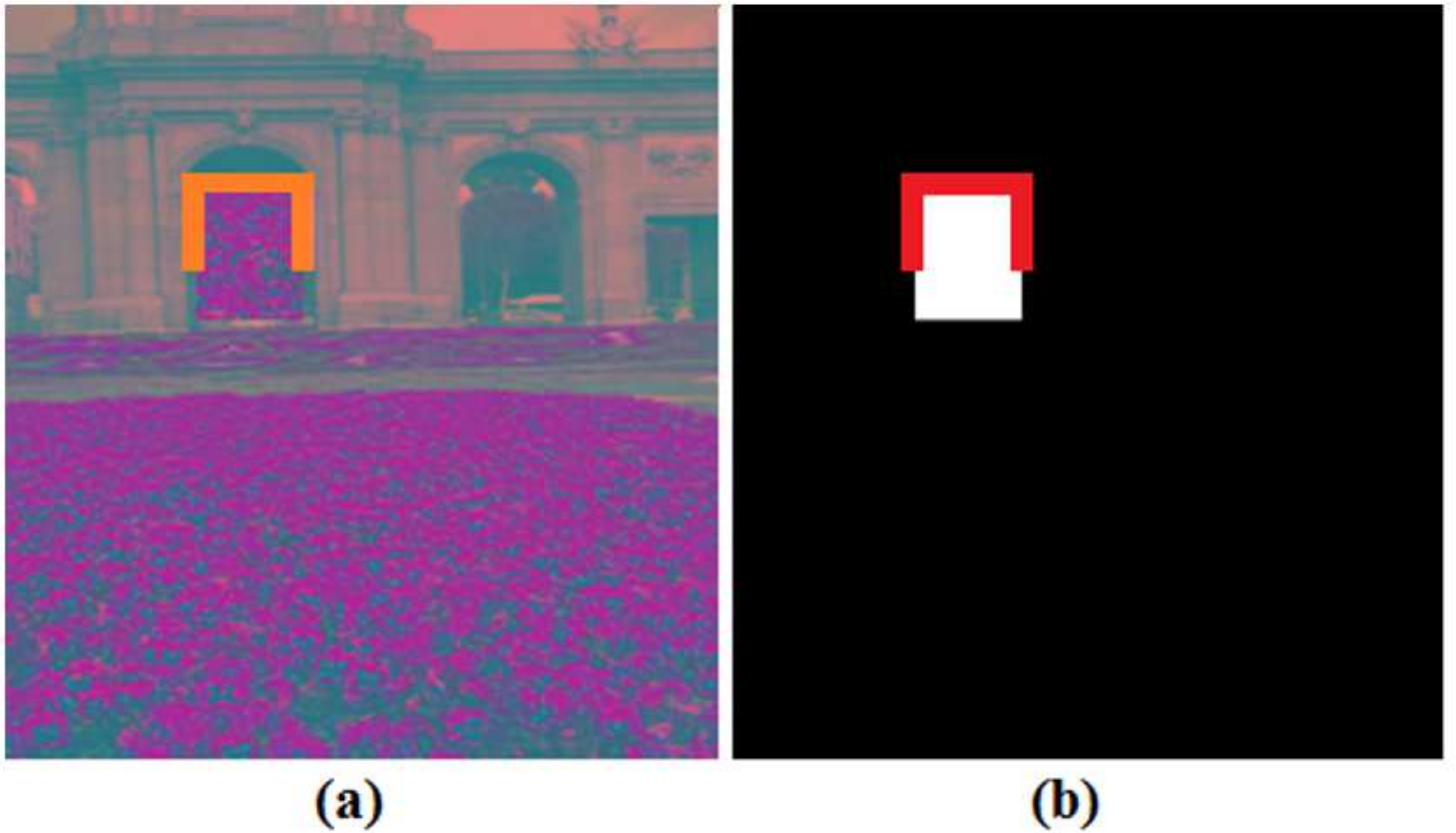


Figure 9

Selecting boundary blocks of spliced region (a) spliced image (b) mask image

Figure 10

Experimental results of images on different color channel (a,d) results of Y channel (b,e) results of Cb channel (c,f) results of Cr channel

Figure 11

Detection results of the splicing images



Figure 12

Pixel-level performance comparison (%) for the four test images in Fig.11 (a) TPR comparison (b) FPR comparison

Figure 13

Pixel-level performance comparison (%) for the Columbia dataset. (a) before post processing results (b) after post-processing (JPEG compression) results (c) after post-processing (Gaussian blur) results