# Passive Detection of Splicing and Copy-Move Attacks in Image Forgery

Mohammad Manzurul Islam (✉), Joarder Kamruzzaman, Gour Karmakar,
Manzur Murshed, and Gayan Kahandawa

School of Science, Engineering and Information Technology, Federation University Australia
{mm.islam, joarder, gour, manzur.murshed,
g.appuhamillage}@federation.edu.au

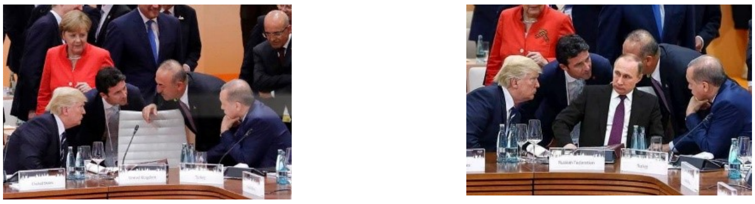
**Abstract.** Internet of Things (IoT) image sensors for surveillance and monitoring, digital cameras, smart phones and social media generate huge volume of digital images every day. Image splicing and copy-move attacks are the most common types of image forgery that can be done very easily using modern photo editing software. Recently, digital forensics has drawn much attention to detect such tampering on images. In this paper, we introduce a novel feature extraction technique, namely Sum of Relevant Inter-Cell Values (SRIV) using which we propose a passive (blind) image forgery detection method based on Discrete Cosine Transformation (DCT) and Local Binary Pattern (LBP). First, the input image is divided into non-overlapping blocks and 2D block DCT is applied to capture the changes of a tampered image in the frequency domain. Then LBP operator is applied to enhance the local changes among the neighbouring DCT coefficients, magnifying the changes in high frequency components resulting from splicing and copy-move attacks. The resulting LBP image is again divided into non-overlapping blocks. Finally, SRIV is applied on the LBP image blocks to extract features which are then fed into a Support Vector Machine (SVM) classifier to identify forged images from authentic ones. Extensive experiment on four well-known benchmark datasets of tampered images reveal the superiority of our method over recent state-of-the-art methods.

**Keywords:** Digital forensics, splicing attack, copy-move attack, Discrete Cosine Transformation, Local Binary Pattern, Support Vector Machine.

## 1 Introduction

Today, Internet of Things (IoT) has emerged as an integrated technology in our daily life. According to Business Insider Intelligence [1], there will be more than 24 billion IoT devices by 2020 which results in approximately four devices per person living on earth. Our everyday essential devices such as wearable sensors, visual sensors, home appliances, security devices, etc. are increasingly being connected to the Internet. Among them, visual sensors play a vital role in physical and cyberspace security and surveillance. Digital social media platforms like Facebook and Instagram are being flooded with millions of images each day. For many cutting-edge applications, people rely on image data more than any other form of data. However, sophisticated digital image editing tools and software have become available. They are very easy to use and they can generate fake images that appear to be very natural. The forged images generated by these tools do not leave any trace for human visual system. Hiding facts, spreading negative propaganda, disrupting operational and decision-making processes have become very common in today's online media. Among all the possible image tampering operations, splicing and copy-move are the most notorious and commonly used attacks on digital images [2]. Image splicing forgery is done by copying one or more portion of an image and pasting it on another image, while in copy-move forgery, one or more objects of an image is copied and then are pasted on the other part of the same image.

As we know that 'a picture is worth a thousand words', an artificially altered image can have devastating consequences. During the 2017 G-20 summit in Germany, AP photojournalist Markus Schreiber captured the image in Fig. 1a prior to the first working session on the very first day of the summit. Later this picture was most likely edited and uploaded in social media as Fig. 1b by a Russian journalist and Putin loyalist Vladimir Soloviev [3]. Although he soon deleted the post from Facebook, it already spread all over the world and introduced new debate and confusion in world politics. In the same way, an altered image can mislead the world leaders in making business decision, taking political steps or even starting a nuclear war.



(a) Authentic image (b) Spliced image

**Fig. 1.** Image splicing example

Modern photomontage does not leave any trace for naked eyes, yet they can be identified through digital forensics. The existing methods for identifying image forgery can be roughly divided into two categories: active and passive. Active methods (e.g., [4]) rely on injecting digital watermark or signature into the original image. To verify the authenticity of an image, the receiver checks if the digital watermark or signature is unchanged or not. Unfortunately, most of the image sensors do not have the capability to integrate complex digital watermarking functionalities because of high cost and resource requirements. As a result, active techniques are not commonly observed and practised in today's data driven IoT network. On the other hand, passive approaches (e.g., [5, 6]) do not need such prior knowledge, require less resources, and hence have drawn much attentions in digital forensics in recent years. The main idea behind passive (blind) detection is that an altered image might not be visibly identifiable as tampered, but tampering obviously introduces disturbance in the structural and statistical characteristics of an image. To be more specific, image tampering introduces new micro-patterns and sharp edges along the boundary of the pasted area. From signal processing's point of view, splicing and copy-move artifacts are the ‘noise’ inserted into a clear signal.

A major portion of images that are targeted for tampering are security sensitive images captured through security and surveillance cameras installed in factory warehouses, shops, financial institutes, military installations, government vaults, border defence etc. These images are mostly in gray scale due to the nature of their applications, lighting condition and recording time (e.g., night time). Again, color images can also be converted into gray scale images. All these justify the advancement of detecting attacks on gray scale images as the attack detection methods for gray scale images can be used in both gray and color images.

Although many researchers have proposed different approaches to image forgery detection with promising accuracy, there are still scopes for the advancement of these techniques using innovative features that are more discriminative and sensitive to the tampering artifacts produced by splicing and copy-move attacks. To achieve this, in this paper, firstly, we introduce a novel feature extraction technique, namely Sum of Relevant Inter-Cell Values (SRIV) for propagating the effects of splicing and copy-move attacks into all features more explicitly than representing it using typical features such as histogram or higher order statistical moments based features. Secondly, using SRIV features, we then propose a passive (blind) detection method using Discrete Cosine Transformation (DCT) and Local Binary Pattern (LBP) for detecting splicing attacks on image. Since LBP can enhance the local changes among the neighbouring DCT coefficient values, first we identify the micro-patterns introduced by splicing operation applying 2D block DCT transformation on image and then, apply LBP in those DCT coefficients. For propagating the effects of the changes into all features, we then extract the features using our proposed SRIV technique applied to the LBP image 2D array. Finally, we feed these features to support vector machine (SVM) for learning and classification. Improved classification accuracy over recent methods described in [5] and [6] using four benchmark datasets substantiate the efficacy of our proposed SRIV technique and image forgery detection approach.

## 2 Related Works

A number of approaches have been proposed in recent years to detect image tampering. They differ mainly on the techniques they adopt to model the structural and statistical changes in forged images. The works reported below utilized SVM for classification once features have been extracted from an image. Among them, the authors who implemented their work based on gray scale image used Columbia dataset [7] while others used different color datasets [8-10].

In [11], Ng et al. proposed bicoherence features to detect image splicing and suggested several methods to improve the capabilities of bicoherence features for splicing detection. They achieved as high as 72% detection accuracy over their own gray image dataset named Columbia [7]. Later, this dataset turned into one of the most popular benchmark datasets for gray scale image splicing detection. Hilbert-Huang transform

(HHT) and moments of characteristics function of wavelet sub-band were used to extract features in [12]. It was the first work to utilize HHT to identify image splicing. The authors reported 80.15% detection accuracy. Chen et al. in [13] adopted statistical moments of characteristics functions of wavelet sub-band and 2D phase congruency to identify splicing artifacts and achieved 82.32% detection accuracy.

A few researchers adopted run-length based approach to identify image splicing. Dong et al. [14] investigated the disturbance of pixel correlation and rationality introduced by image splicing operation . They proposed a run-length and edge statistics based approach to identify spliced images from authentic ones and attained 76.52% accuracy. Later, this method was improved by He et al. [15] in terms of accuracy (80.58%), computational cost and feature dimensionality.

Shi et al. [16] proposed a method based on a natural image model where statistical moment features and Markov features are extracted from a given image as well as from multi block DCT of the same image. He et al. [17] expanded the original Markov features by Shi et al. and modelled the splicing artifacts based on Markov features in DCT and DWT domains. Unlike [16], they considered both intra-block and inter-block correlation among DCT coefficients. Although methods in [16] and [17] achieved satisfactory result on Columbia dataset, the detection accuracy was reduced to 84.86% and 89.76%, respectively when applied on CASIA 2 dataset [10] which is a more challenging dataset in nature [17]. In [18], Wang et al. proposed a method to identify splicing attacks by modelling the edge information of image in chroma space as a finite-state Markov chain and considered its stationary distribution as features. This method achieved 95.6% accuracy on CASIA 2 dataset.

Zhang et al. [5] and Alahmadi et al. [6] proposed their methods utilizing both DCT and LBP. They mainly differ based on the order of DCT and LBP application on image blocks and feature extraction technique. Zhang et al. applied LBP operator on the magnitude component of 2D-DCT coefficients of the gray scale input image. They extracted features by calculating the histogram of the resultant LBP 2D array. In contrast, Alahmadi et al. divided the chrominance channels of the input image into blocks. Then LBP is applied and the resultant LBP 2D array of each block is transformed into frequency domain using 2D-DCT. Finally, features were extracted by calculating the standard deviation of the corresponding inter-cell DCT coefficients. Both the methods are promising in terms of detection accuracy. Inspired by the ability of DCT and LBP to generate discriminative features of authentic and spliced images, we propose a new feature extraction technique and using it, an image forgery detection approach, which is described in the following section.

## 3 Proposed Method

Image splicing and copy-move attacks are very widespread attacks on images. The detection mechanism is a binary decision problem – whether an image is forged or not. These attacks introduce structural and statistical changes in the host image which, in turn, affect features that can be extracted to describe the image. Therefore, a number of techniques need to be applied on the images before final features can be derived to feed into a chosen classifier. Figure 2 depicts the overall mechanism in our proposed method and its key components are described in the following sections.

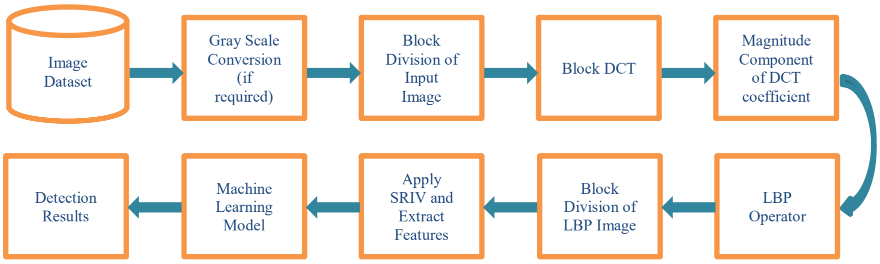


Image Dataset
Gray Scale Conversion (if required)
Block Division of Input Image
Block DCT
Magnitude Component of DCT coefficient
LBP Operator
Block Division of LBP Image
Apply SRIV and Extract Features
Machine Learning Model
Detection Results

**Fig. 2.** Proposed image splicing and copy-move detection system

### 3.1 Converting images into gray scale

We have implemented our system using four benchmark datasets commonly used for image splicing and copy-move detection. Among them, one dataset is already in gray scale and remaining datasets are in color space. As a result, we converted color datasets into gray scale. It is worth noting that many applications in surveillance and security system rely on gray scale images that are collected in night time environment.

### 3.2 Block division of input image

Splicing and copy-move operation can be applied in different ways on host images. Again, different image fragments may be pasted into different parts of the host image. It is not expected to be able to identify the splicing artifacts by one single block size. Hence, for different types of images, different sized block divisions are essential to identify discriminative features of the forged images. Our proposed method performs block divisions in two phases. In the first phase, we divide an input image into square-sized blocks. The second phase is explained later in Section 3.5. We have tested our system with different block sizes: 4x4, 8x8 and 16x16 as well as combining features from all three mentioned blocks. The following procedure divides an image into blocks. Let $I^{wb \times hb}$ be a gray scale image of size $wb \times hb$ pixels. We divide $I^{wb \times hb}$ into $w \times h$ non-overlapping blocks of size $b \times b$ pixels. The resultant image block 2D array is,

$$I^{wb \times hb} = \begin{bmatrix} I_{1,1}^{b \times b} & \cdots & I_{1,w}^{b \times b} \\ \vdots & \ddots & \vdots \\ I_{h,1}^{b \times b} & \cdots & I_{h,w}^{b \times b} \end{bmatrix}. \tag{1}$$

### 3.3 Block discrete cosine transformation (BDCT)

Image tampering introduces new micro patterns and sharp edges along the affected regions. It changes the local frequency distribution by altering regularity, smoothness, continuity of the tampered image and thus it disturbs the natural correlation between image pixels [16]. It is essential to reduce the diversity of image content and magnify the effects of image splicing and copy-move attack before final feature extraction. To represent the degree of content change of an image, it is converted into frequency domain. BDCT has shown promising result in representing pixel domain changes in local frequency distribution as it exhibits excellent decorrelation and energy compaction properties [19]. We apply 2D-DCT on the blocks of $I^{wb\times hb}$ to generate DCT coefficients. Let $Y^{wb\times hb}$ be the resultant transform domain coefficient after applying 2D-DCT on each block and it is given by,

$$Y^{wb\times hb} = \begin{bmatrix} Y_{1,1}^{b\times b} & \cdots & Y_{1,w}^{b\times b} \\ \vdots & \ddots & \vdots \\ Y_{h,1}^{b\times b} & \cdots & Y_{h,w}^{b\times b} \end{bmatrix} , \tag{2}$$

where $Y_{\mathrm{i},j}^{b\times b} = 2D\text{-}DCT\left(I_{\mathrm{i,j}}^{b\times b}\right), 1 \le i \le w, 1 \le j \le h$. The 2D-DCT of an input block $I_{\mathrm{i,j}}^{b\times b}$ produces the output block $Y_{\mathrm{i,j}}^{b\times b}$ as,

$$Y_{\mathrm{i,j}}^{b\times b}(\mathrm{p,q}) = \alpha_p \alpha_q \sum_{m=0}^{b-1} \sum_{n=0}^{b-1} I_{\mathrm{i,j}}^{b\times b}(m,n) \cos\frac{\pi(2m+1)p}{2b} \cos\frac{\pi(2n+1)q}{2b} , \tag{3}$$

where $0 \le p \le b-1, 0 \le q \le b-1$ and

$$\alpha_p = \begin{cases} \sqrt{\frac{1}{b}}, & \text{if } p = 0 \\ \sqrt{\frac{2}{b}}, & \text{otherwise} \end{cases} , \quad \alpha_q = \begin{cases} \sqrt{\frac{1}{b}}, & \text{if } q = 0 \\ \sqrt{\frac{2}{b}}, & \text{otherwise} \end{cases} . \tag{4}$$

### 3.4 Local binary pattern (LBP) operator

To identify and enhance different splicing artifacts, we employ LBP operator on the magnitude component of $Y^{wb\times hb}$. LBP is a computationally inexpensive yet robust texture descriptor. The main idea for adopting LBP in our system is to enhance the local changes among the neighbouring DCT coefficient values because of the occurrences of micro-patterns and sharp edges that are introduced by splicing and copy-move attacks. LBP can effectively highlight these tampering artifacts and enhance them in the host images. In LBP, each pixel of a given 2D array is compared with its neighbouring pixels and an LBP code is generated for that pixel. It is computed as below:

Let $L^{wb\times hb}$ be the resultant LBP array generated by applying LBP operator on the magnitude components of $Y^{wb\times hb}$ and is given by,

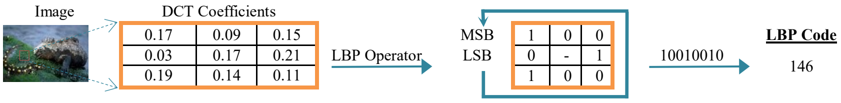
$$L^{wb\times hb} = LBP_{N,R}(|Y^{wb\times hb}|) , \tag{5}$$

$$LBP_{N,R} = \sum_{n=0}^{N-1} g(p_n - p_c)2^n \ . \tag{6}$$

Here, $N$ is the number of neighbor pixels; $R$ is the radius and $p_c$ is the central pixel which is compared with each neighbouring pixel $p_n(n = 0,1, \ldots, N-1)$. In our proposed method, we use $N = 8$ and $R = 1$. The function $g(p_n - p_c)$ is given by:

$$g(p_n - p_c) = \begin{cases} 1, & p_n - p_c \geq 0 \\ 0, & p_n - p_c < 0 \end{cases} . \tag{7}$$

For $N = 8$ and $R = 1$, the central pixel $p_c$ compares its own value with neighbouring 8 pixels. If the neighbor pixel's value is greater than or equal to the central pixel value, then 1 is recorded; otherwise 0. Based on these comparisons, central pixel $p_c$ stores it's LBP code. Figure 3 explains the procedure with an example. Here, the binary values are obtained after comparison between central pixel $p_c$ and the 8 neighboring pixels. Then the 8-bit binary digit is formed starting from Least Significant Bit (LSB) to Most Significant Bit (MSB). Finally, the binary digit is converted into decimal and the LBP code is stored in place of central pixel $p_c$.



| | | |
|---|---|---|
| 0.17 | 0.09 | 0.15 |
| 0.03 | 0.17 | 0.21 |
| 0.19 | 0.14 | 0.11 |

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | - | 1 |
| 1 | 0 | 0 |

**Fig. 3.** LBP code generation procedure

### 3.5 Block division of LBP image

In the second phase of block division, we divide the LBP image 2D array $L^{wb\times hb}$ into same size of blocks similar to the block division done in Section 3.2. We divide $L^{wb\times hb}$ into $w \times h$ non-overlapping blocks of size $b \times b$ pixels. The resultant LBP image block 2D array is given by,

$$L^{wb\times hb} = \begin{bmatrix} L_{1,1}^{b\times b} & \cdots & L_{1,w}^{b\times b} \\ \vdots & \ddots & \vdots \\ L_{h,1}^{b\times b} & \cdots & L_{h,w}^{b\times b} \end{bmatrix} . \tag{8}$$

### 3.6 Apply SRIV and feature generation

As shown in Fig. 2, in our proposed method, the SRIV features are derived from LBP codes generated using DCT coefficients. The main reason for adopting such approach in a specific order is that DCT coefficients represent the pixel value variations in the spatial domain, while LBP enhances the local changes among the neighboring DCT coefficient values, magnifying the changes of splicing and copy-move attacks in higher frequency components. To make the detection system more accurate, we need to preserve the local changes captured by LBP as much as possible. Since splicing attacks usually make subtle changes in an image, these local changes can be regarded as outliers. Mean is most affected by outliers than other statistical measures. The SRIV features in our proposed method are based on an aggregation operator (sum). These features are similar to the mean based features as the number of blocks having a particular size (e.g., 8x8) in a specific image remains always the same. Therefore, this vindicates the SRIV features can represent the local changes because of splicing and copy-move attacks more accurately than the standard deviation based features used in [6] and the histogram-based features applied in [5]. We experimented our method with different block sizes as mentioned in Section 3.2 and 3.5. Consequently, we have varying dimensionality of features as listed in Table 2. The SRIV features are computed as below:

Let $Z_k^{w \times h}$ be the $k$-th LBP code values of all blocks in $L^{wb \times hb}$. Therefore,

$$Z_k^{w \times h} = \begin{bmatrix} L_{1,1}^{b \times b}(k) & \cdots & L_{1,w}^{b \times b}(k) \\ \vdots & \ddots & \vdots \\ L_{h,1}^{b \times b}(k) & \cdots & L_{h,w}^{b \times b}(k) \end{bmatrix}, \; 1 \leq k \leq b^2 , \tag{9}$$

where $L_{u,v}^{b \times b}(k)$ is the $k$-th LBP code of that block. Then the $k$-th feature $F_k$ on the whole image is calculated as,

$$F_k = \sum_{u=1}^{w} \sum_{v=1}^{h} L_{u,v}^{b \times b}(k) \; . \tag{10}$$

To justify our argument as mentioned before that the SRIV features are more discriminative and more effective than the standard deviation based features, we extracted features by our approach using both SRIV and standard deviation. For a representative sample, we selected one authentic image (Fig. 4a) and its spliced version (Fig. 4b) from CASIA 2 dataset. We then plotted the extracted features in a graph (Fig. 5) where x-axis represents feature number and y-axis represent the feature values. From Fig. 5, it is clearly visible that the SRIV feature values vary more sharply than those of standard deviation for both the original and its spliced image. This is also evidenced by the fact that the standard deviation of SRIV feature values for both the original and its spliced image (0.18, 0.17) are higher than those of standard deviation based feature values (0.16, 0.15). All of these evidences show the SRIV features are more discriminating and hence more effective than those for standard deviation.

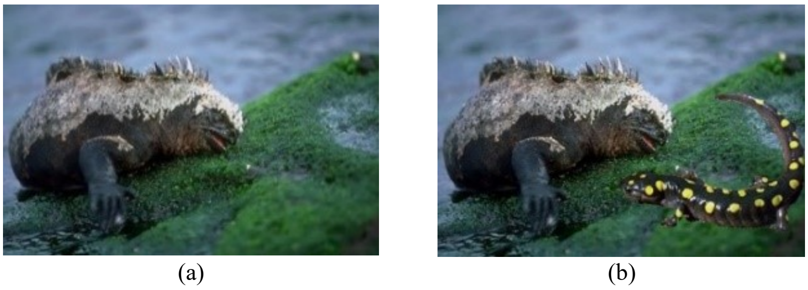

(a) (b)

**Fig. 4.** Authentic image (a) and its spliced image (b) from CASIA 2

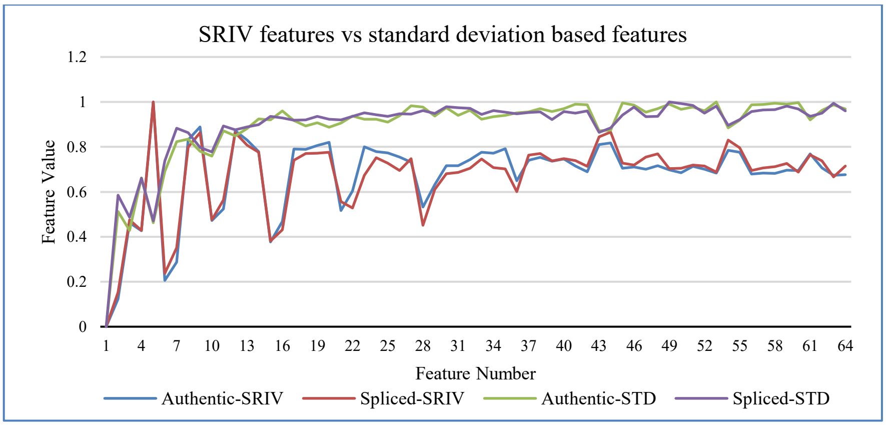


**Fig. 5.** Comparing the SRIV features with the standard deviation based features

# 4 Experiments and results

## 4.1 Description of datasets

We have evaluated our proposed system using four publicly available and well recognized benchmark datasets for image splicing detection: (i) Columbia gray [7], (ii) Columbia Uncompressed [8], (iii) CASIA 1 [9] and (iv) CASIA 2 [10]. We have summarized the datasets used to evaluate our method in Table 1.

**Table 1.** Summary of the datasets

| Dataset | Image Size | Image Type | No. of Images | | | Tampering Method |
|---|---|---|---|---|---|---|
| | | | Authentic | Tampered | Total | |
| **Columbia** | 128 x 128 | JPG | 933 | 912 | 1845 | Simple crop-and-paste |
| **Columbia Uncomp.** | 757 x 568 - 1152 x 768 | TIF, BMP | 183 | 180 | 363 | Simple crop-and-paste, spliced image from exactly 2 cameras |
| **CASIA 1** | 384 x 256, 256 x 384 | JPG | 800 | 921 | 1721 | Photoshop with pre-processing; No post-processing |
| **CASIA 2** | 240 x 160 - 900 x 600 | JPG, TIF, BMP | 7491 | 5123 | 12614 | Photoshop with pre-processing and/or post -processing |

## 4.2 SVM Classifier and model validation

We adopted SVM as classifier (LIBSVM [20]) as it shows promising performance in many application domains including splicing detection. Radial Basis Function (RBF) kernel was selected for this work. The regularisation parameter ($C$) and variance of RBF kernel ($\gamma$) were chosen through grid-search method and sixfold cross-validation was used for model evaluation. For every experiment, similar to [5], we picked 5/6th of the tampered images and 5/6th of the authentic images to train the SVM classifier. The remaining 1/6th tampered images and 1/6th authentic images were used to test the trained classifier. MATLAB was used for feature extraction and data pre-processing.

### 4.3 Results and discussion

We summarise the detection accuracy for features derived from block size of 4x4, 8x8, 16x16 individually as well as their combined features (4x4 + 8x8 + 16x16) in Table 2. The effect of different sized block DCT varies from dataset to dataset. Our proposed method achieves detection accuracy of 85.64%, 94.49%, 95.40% and 99.76% over Columbia gray, Columbia Uncompressed, CASIA 1 and CASIA 2 datasets respectively. Additionally, the precision, recall and AUC (Area Under ROC curve) of our system is also reported in Table 2.

Columbia gray dataset is a popular but older dataset with low resolution fixed dimension (128 x 128) JPG images. Our method performs best (85.64%) for block size of 8x8 on this dataset while block size 4x4 and 16x16 reduces detection accuracy by 7% and 5%, respectively. Combined features from all three blocks provides 84.34% detection accuracy. Similar trend is observed for Columbia Uncompressed dataset where combining features from all blocks does not yield the best result. However, we achieved the best results by combining features for CASIA 1 (95.40%) and CASIA 2 (99.76%) datasets. Our method has produced quite encouraging result in these datasets, which demonstrates the strength of the feature extraction and overall techniques used in our approach.

**Table 2.** Overall detection accuracy in our proposed method with varying block size. Note that image in Columbia Uncomp., CASAI 1 and CASIA 2 are converted into gray scale

| Block Size | Feature Dimensionality | Evaluation | Columbia | Columbia Uncomp. | CASIA 1 | CASIA 2 |
|---|---|---|---|---|---|---|
| 4x4 | 16 | Accuracy (%) | 78.5908 | 87.6033 | 72.8438 | 95.0844 |
| | | Precision | 0.789 | 0.857 | 0.761 | 0.935 |
| | | Recall | 0.773 | 0.900 | 0.718 | 0.945 |
| | | AUC | 0.786 | 0.876 | 0.729 | 0.950 |
| 8x8 | 64 | Accuracy (%) | **85.6369** | 92.2865 | 93.5897 | 97.6453 |
| | | Precision | 0.852 | 0.942 | 0.934 | 0.968 |
| | | Recall | 0.859 | 0.900 | 0.947 | 0.975 |
| | | AUC | 0.856 | 0.923 | 0.935 | 0.976 |
| 16x16 | 256 | Accuracy (%) | 80.8672 | **94.4904** | 87.7622 | 99.231 |
| | | Precision | 0.803 | 0.944 | 0.875 | 0.988 |
| | | Recall | 0.813 | 0.944 | 0.900 | 0.993 |
| | | AUC | 0.809 | 0.945 | 0.876 | 0.992 |
| 4x4 + 8x8 + 16x16 | 336 | Accuracy (%) | 84.336 | 94.2149 | **95.3963** | **99.7622** |
| | | Precision | 0.830 | 0.944 | 0.946 | 0.996 |
| | | Recall | 0.860 | 0.939 | 0.970 | 0.998 |
| | | AUC | 0.844 | 0.942 | 0.953 | 0.998 |

### 4.4 Comparison with recent methods

Among various methods for detecting splicing and copy-move attacks (Section 2), two existing ones adopt both DCT and LBP in their systems and report good detection accuracy. Since they have not reported results with all four datasets, to make a fair comparison, we implemented those two methods to get their detection capability for each dataset. The basic experimental setup remains the same as mentioned in Section 4.2.

In [5], Zhang et al. found best accuracy for combined features extracted from block size 4x4, 8x8 and 16x16. They identified best parameters for SVM and RBF kernel through grid-search method. In [6], Alahmadi et al. attained the best accuracy with 16x16 blocks and LBP parameter P(neighbour) = 8, R(radius) = 1, SVM parameter $C$ = $2^5$ with RBF kernel $\gamma$ = $2^{-5}$. We implemented their methods using their reported parameters. Table 3 depicts the comparison of detection accuracy among different methods across different datasets. It is clearly visible that our method's overall accuracy is higher (up to 5%) than two existing state-of-the-art methods in all four benchmark datasets. To the best of our knowledge, detection accuracy of 99.76% is the highest among all other methods available in the literature that deal with gray scale images. Our method outperforms others in terms of precision, recall and AUC in all cases except for recall in Columbia Uncomp. and precision in CASIA 1. Specially, our method attains better AUC, which is a more accepted performance metric, for all four benchmark datasets.

**Table 3.** Comparison of detection accuracies of the proposed method with [5] and [6]

| Dataset | Evaluation | Proposed Method | Method in [5] | Method in [6] |
|---|---|---|---|---|
| **Columbia** | Accuracy (%) | **85.6369** | 81.1924 | 77.1816 |
| | Precision | 0.852 | 0.806 | 0.768 |
| | Recall | 0.859 | 0.827 | 0.772 |
| | AUC | 0.856 | 0.816 | 0.772 |
| **Columbia Uncomp.** | Accuracy (%) | **94.4904** | 92.8375 | 93.3884 |
| | Precision | 0.944 | 0.910 | 0.994 |
| | Recall | 0.944 | 0.950 | 0.872 |
| | AUC | 0.945 | 0.929 | 0.933 |
| **CASIA 1** | Accuracy (%) | **95.3963** | 92.5991 | 78.0886 |
| | Precision | 0.946 | 0.951 | 0.821 |
| | Recall | 0.970 | 0.909 | 0.755 |
| | AUC | 0.953 | 0.927 | 0.783 |
| **CASIA 2** | Accuracy (%) | **99.7622** | 84.1433 | 94.1965 |
| | Precision | 0.996 | 0.812 | 0.918 |
| | Recall | 0.998 | 0.793 | 0.935 |
| | AUC | 0.998 | 0.834 | 0.939 |

## 5 Conclusion

In this paper, we introduced SRIV, a novel feature extraction technique using which we proposed a robust model for detecting splicing and copy-move attacks on image data adopting both DCT and LBP in the mentioned order. These attacks change the pixel values in the spatial domain by introducing sharp edges, alien micro-patterns and so on. DCT shows excellent image pixel decorrelation and energy compaction properties which is used to capture the change in the spatial domain. Then, LBP is applied on the magnitude component of the 2D array returned by DCT to enhance the local changes among the neighbouring DCT coefficient values. Finally, SRIV is applied on the LBP image blocks to extract features. These features are used to train an SVM with RBF kernel to detect the tampered images. Experimental results confirm that our method outperforms other methods across four benchmark image forgery detection datasets. Future work will target detection of splicing and copy-move attacks on color images.

**Acknowledgement.** This work is supported by the Research Priority Area (RPA) scholarship of Federation University Australia.

## References

1. Meola, A.: The Internet of Things: Meaning & Definition. Business Insider (2018)
2. Redi, J.A., Taktak, W., Dugelay, J.-L.: Digital image forensics: a booklet for beginners. Multimedia Tools and Applications **51**, 133-162 (2011)
3. Novak, M.: That Viral Photo of Putin and Trump is Totally Fake. gizmodo.com (2017)
4. Kwitt, R., Meerwald, P., Uhl, A.: Lightweight Detection of Additive Watermarking in the DWT-Domain. IEEE Transactions on Image Processing **20**, 474-484 (2011)
5. Zhang, Y., Zhao, C., Pi, Y., Li, S., Wang, S.: Image-splicing forgery detection based on local binary patterns of DCT coefficients. Security and Communication Networks **8**, 2386-2395 (2015)
6. Alahmadi, A.A., Hussain, M., Aboalsamh, H.A., Ghulam, M., Bebis, G., Mathkour, H.: Passive detection of image forgery using DCT and local binary pattern. Signal, Image and Video Processing **11**, 81-88 (2017)
7. Ng, T.-T., Chang, S.-F.: A Model for Image Splicing. In: IEEE Int. Conf. Img. Proc. (2004)
8. Hsu, Y.-F., Chang, S.-F.: Detecting Image Splicing Using Geometry Invariants And Camera Characteristics Consistency. Int. Conf. on Multimedia and Expo, Canada (2006)
9. Dong, J., Wang, W., Tan, T.: CASIA Image Tampering Detection Evaluation Database. In: IEEE International Conference on Signal and Information Processing, pp. 422-426 (2013)
10. Dong, J., Wang, W.: CASIA Tampered Imaged Detection Evaluation Database (CASIA TIDE v2.0). National Laboratory of Pattern Recognition, Chinese Academy of Science (2009-2016)
11. Ng, T.-T., Chang, S.-F., Sun, Q.: Blind detection of photomontage using higher order statistics. In: IEEE International Symposium on Circuits and Systems, pp. 688-691 (2004)
12. Fu, D., Shi, Y.Q., Su, W.: Detection of Image Splicing Based on Hilbert-Huang Transform and Moments of Characteristic Functions with Wavelet Decomposition. pp. 177-187. Springer Berlin Heidelberg (2006)
13. Chen, W., Shi, Y.Q., Su, W.: Image splicing detection using 2-D phase congruency and statistical moments of characteristic function. Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505. SPIE, Washington (2007)
14. Dong, J., Wang, W., Tan, T., Shi, Y.Q.: Run-Length and Edge Statistics Based Approach for Image Splicing Detection. In: Kim, H.-J., Katzenbeisser, S., Ho, A.T.S. (eds.) Digital Watermarking, pp. 76-87. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
15. He, Z., Sun, W., Lu, W., Lu, H.: Digital image splicing detection based on approximate run length. Pattern Recognition Letters **32**, 1591-1597 (2011)
16. Shi, Y.Q., Chen, C., Chen, W.: A natural image model approach to splicing detection. Proceedings of the 9th workshop on Multimedia & security, pp. 51-62. ACM, USA (2007)
17. He, Z., Lu, W., Sun, W., Huang, J.: Digital image splicing detection based on Markov features in DCT and DWT domain. Pattern Recogn. **45**, 4292-4299 (2012)
18. Wang, W., Dong, J., Tan, T.: Image tampering detection based on stationary distribution of Markov chain. In: IEEE Int Conf on Image Processing, pp. 2101-2104 (2010)
19. Khayam, S.A.: The discrete cosine transform (DCT): theory and application. Michigan State University (2003)
20. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:21-27:27 (2011)