Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-09707-7

Image splicing detection technique based on Illumination-Reflectance model and LBP

Patrick Niyishaka[1] · **Chakravarthy Bhagvati**[1]

Received: 24 April 2020 / Revised: 22 July 2020 / Accepted: 25 August 2020 /
Published online: 11 September 2020

Abstract
A Copy-create digital image forgery is image tampering that merges two or more areas of images from different sources into one composite image; it is also known as image splicing. Excellent forgeries are so tricky that they are not noticeable to the naked eye and don't reveal traces of tampering to traditional image tamper detection techniques. To tackle this image splicing detection problem, machines learning-based techniques are used to instantly discriminate between the authentic and forged image. Numerous image forgery detection methods to detect and localize spliced areas in the composite image have been proposed. However, the existing methods with high detection accuracy are computationally expensive since most of them are based on hybrid feature set or rely on the complex deep learning models, which are very expensive to train, run on expensive GPUs, and require a very large amount of data to perform better. In this paper, we propose a simple and computationally efficient image splicing forgery detection that considers a trade-off between performance and the cost to the users. Our method involves the following steps: first, luminance and chrominance are found from the input image; second, illumination is estimated from Luminance using Illumination-Reflectance model; third, Local Binary Patterns normalized histogram for illumination and Chrominance is computed and used as the feature vector for classification using the following machine learning algorithms: Support Vector Machine, Linear Discriminant Analysis, Logistic Regression, K-Nearest Neighbors, Decision Tree, and Naive Bayes. Extensive experiments on the public dataset CASIA v2.0 show that the new algorithm is computationally efficient and effective for image splicing tampering detection.

Keywords Splicing · Illumination · Reflectance · LBP · Luminance · Chrominance

1 Introduction

Image tampering aims to modify the semantic meaning of the visual message to deceive the viewers by adding, removing, or changing, some major areas of image [27]. Digital

✉ Patrick Niyishaka
niyishakapatrick@gmail.com

[1] University of Hyderabad, Hyderabad, India

images play a vital role in different fields including, crime scene investigation, images are used in courts of law as evidence, medical images are used for detecting and diagnosis of diseases like brain tumors, and several million images are uploaded to the social media platforms per day. However, digital image editing tools like GIMP or Photoshop have made effortless the process of editing and doctoring images. Those fake images mislead the public and to discern which is the authentic image or forged is a great challenge. Therefore, the image forgery detection techniques to ensure the authenticity of digital images are extremely needed. The field of *Digital Image Forensics* (DIf) aims at validating the authenticity of images [27]. Generally, the DIf categorises the digital image tampering detection algorithms into two categories named *Active* and *Passive*. Active detection algorithm embeds digital watermark or digital signature in the image. Passive or blind detection algorithm does not consider any prior information to be embedded in the image. Since most digital images on social media, internet, etc, do not have digital watermarks/signatures embedded, the blind image forensics has attracted more attention. The *passive approach* categorizes the digital image forgery in three main categories based on the process they are created. These categories are *Image Retouching*, *Image Splicing*, and the *Copy-Move Forgery* [27].

Image splicing or Composite or Copy-create forgery: involves the composition or merging of two or more images changing the original image significantly to produce a forged image as shown in Fig. 1 [9].

Image retouching or Airbrushing: is carried out mostly to enhance or reduce the image features but the subject of the image remains the same [3].

Copy-move forgery or Cloning: a region in an image is copied and pasted to another part of the same image [19, 20].

This article focuses on image splicing forgery and its detection methods. Researchers have been working to tackle the image splicing forgery, most of the state-of-the-art methods consider the image splicing problem as a binary classification problem [6, 35], that is to classify an image as authentic(original) or tampered(forged). However, this approach has also raised the following question: what are the appropriate features that discriminate an authentic region from tampered that can be used for the image splicing classification task? Existing techniques are very computationally costly in terms of hardware, feature extraction, training time, running time, and feature vector size. To expose the splicing forgery, we propose to use a simple model based on illumination and chrominance features, our method is very computationally efficient in terms of feature extraction, training and running time, and uses a very small feature vector size to instantly discriminate between the authentic and forged image.

Illumination and chrominance are good features for splicing detection because image splicing involves the merging of two or more images to fabricate a new composite image. Since those images involved in splicing forgery are taken from different cameras with different lighting conditions, it is tough to achieve proper illuminant conditions for the entire image into the newly created image. Therefore, image splicing introduces illumination inconsistency in the fabricated image. The authors in [11, 29], showed that the traces of tampering that can not be detected by the naked eye could be hidden in the chrominance channel

Fig. 1 Image splicing model

because the human eye is more sensitive to luminance than it is to chrominance. Therefore, we propose to use the illumination and chrominance features to expose the splicing forgery.

Our primary contributions are summarized as follows: we use a max/mini-filters to approximate edge-preserving filter [5], then we apply this filter to the luminance channel to extract the illumination component based on Illumination-reflectance model. To construct the feature vector, we use Local Binary Patterns [18] normalized histogram computed from the illumination component and chrominance channel. Our method is flexible and can take in various machine learning classifiers including, Support Vector Machine, Linear Discriminant Analysis, Logistic Regression, K-Nearest Neighbors, Decision Tree, and Naive Bayes, for intensive testing, effective modeling and discrimination.

The rest of the paper is structured as follows: in Section 2 the related works, which are state-of-art forgery detection techniques are discussed. The proposed technique is discussed in Section 3, and the Section 4 describes the experiments and reports the detection results.

2 Related work

Numerous detection techniques have been proposed to tackle the splicing forgery problem, [33] proposed a technique for splicing detection based on the Gray Level Co-occurrence Matrix(GLCM) features. First, the image is converted into YCbCr color space and they showed that the human eye is more sensitive to luminance than to chrominance. Then the GLCM of the chrominance channel is used as features. To reduce the dimension of the feature vector and increase the accuracy of the classifier, a Boost Feature Selection (BFS) method is used. Finally, the feature vectors are fed to the LIBSVM classifier to detect the forged image. [25] proposed a detection method based on the Markov chain, DCT, and DWT. Grey data is generated from blocked pictures to build DCT and DWT, and the DCT and DWT coefficients of blocked pictures are computed. Then Markov features are created from the Transition probability matrix in spatial, DCT, and DWT areas. PCA is applied to reduce the dimensionality of pictures and enhancing the correlation among pixels. Finally, the Markov feature vector is fed to the ENSEMBLE classifier to detect the Authentic and forged image. [9] proposed an algorithm based on a deep learning approach and wavelet transform to detect the spliced image. In the deep learning approach, Convolution Neural Network (CNN) is used to automatically extract features from the considered image. After applying CNN, Haar Wavelet Transform (HWT) is also applied to build the feature vector. Then a Support Vector Machine (SVM) is used to classify the considered image as authentic or forged. [35] proposed an image splicing detection and localization technique based on the local feature descriptor which is learned by a deep convolutional neural network. They used a two-branch CNN to automatically learn hierarchical representations from the input image. The first layer of CNN is used to suppress the effects of image contents and extract features designed for image splicing detection applications. The pre-trained CNN-based local descriptor is used to extract the block-wise dense features and the block pooling feature fusion strategy is adopted to obtain the final discriminative features for image splicing detection with Support Vector Machine (SVM). Based on the pre-trained CNN model and the fully connected conditional random field (CRF), they developed the image splicing localization scheme. [6] proposed a method that extracts coefficient-wise Markov features and block-wise Markov features in the discrete cosine transform (DCT) domain. Then, a feature vector is obtained by combining these two Markov features and SVM is used to classify images as authentic or spliced. [36] used Fast Shallow Convolution Neural Network

(SCNN) to distinguish the boundaries of forged regions from original edges, their model is trained on 60671 authentic patches and 43166 tampered patches, and they localized the forged areas using a probability map.

[15] used a hybrid CNN-LSTM model to learn the discriminative features since the tampered regions exhibit discriminative features in boundaries shared with neighboring authentic pixels. The technique proposed by [2] uses a new form of a convolutional layer that is specifically designed to suppress the content of images and adaptively learn manipulation detection features. [14] combined region and texture features for Image splicing detection. The edge-based fetaure, Saliency-based feature, and Wavelet-based feature are captured on regions, and on these regions, the texture feature- rotation invariant co-occurrences among adjacent LBP (RiCoLBP) is applied. The features thus obtained are optimized using Non-Negative Matrix Factorization and fed to SVM for classification.

Table 1 outlines various image splicing forgery detection methods.

The handcrafted features-based methods [14, 18] require an optimization or feature selection step like the Non-Negative Matrix Factorization or the local learning based (LLB) to reduce the number of features, as well as to enhance the performance of the system. Our method avoids the need of this step and uses a very small size of feature vector.

Deep learning models used in this field endure complex modifications to perform better, they train on massive data and run on expensive GPUs, e.g., in their design form, convolutional neural networks (CNN) will learn features related to image's content not to tampering features [2], this implies that simple and computationally efficient forgery detection algorithms are still needed in the field.

3 Proposed method

The methods [26] and [18] exhibit a high accuracy in splicing forgery detection. However, they are computationally expensive, the former is based on CNN with eight hidden layers and requires a massive amount of data samples to train the discriminate CNN model. The latter uses the steerable pyramid transform which involves multi-scale, multi-orientation

Table 1 Various image splicing forgery detection models

Methods	Authors
Using Handcrafted Features	
Using Markov Features	[6]
Adjacent LBPs	[14]
Using SPT and LBP	[18]
Using Deep Learning	
Using shallow CNN	[36]
Using hybrid CNN-LSTM	[15]
Learning Rich Features	[24]
Using Deep learning	[26]
Using CNN + New CL	[2]
Using Deep Learning Local Descriptor	[35]

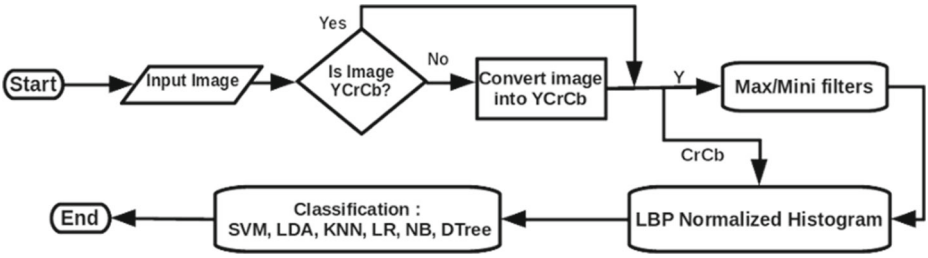

Fig. 2 Flowchart for the proposed method

image decomposition [8] where the given image is decomposed into subbands having different resolutions (scales) and frequencies, and various orientations with the length of the feature vector is more than 3,500. Therefore we propose a technique that is very computationally efficient for splicing forgery detection. The proposed technique involves the following steps: first, the input image is converted to Y and $CrCb$ color space; Second, uses Illumination-Reflectance model to extract illumination component from Y; Finally, Local Binary Patterns normalized histogram extracted from illumination and $CbCr$ is used as a feature vector for classification using different machine learning algorithms. The diagram of the proposed method is shown in Fig. 2.

3.1 Convert input image to YCbCr

$YCbCr$ isolates luma/luminance, the intensity information, from chroma/chrominance, the colour information. Y is the luminance component, Cb, and Cr, are the blue-difference and red-difference chroma components. The authors in [1, 33] showed that the chrominance channel is suitable for splicing forgery detection because most tampering clues which can not be detected by the naked eye, are hidden in the chromatic channel since the human visual system is more sensitive to overall intansity Y changes than to colour $CbCr$ changes [34]. As shown in Fig. 3 the input image is first converted from the RGB to $YCbCr$ color space by:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \tag{1}$$

$$Cb = 0.492(B - Y) \tag{2}$$

$$Cr = 0.877(R - Y) \tag{3}$$

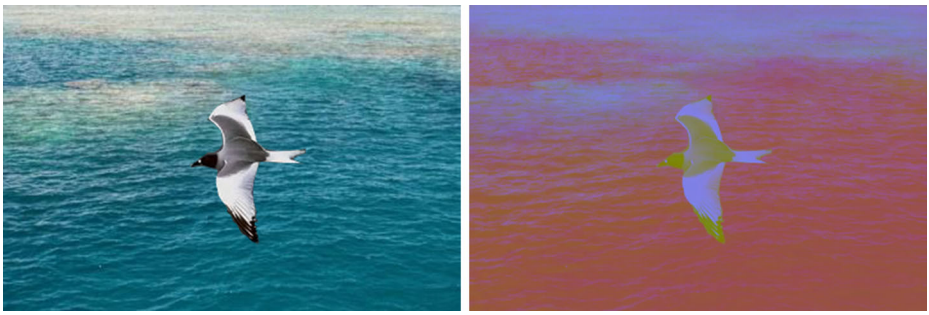

Fig. 3 Convert input image from RGB to $YCbCr$

3.2 Extracting illumination component from luma/luminance based on illumination-reflectance model

The illumination-reflectance model of image formation describes how the object surfaces interact with light [5]. It is used in numerous image enhancement applications based on Homomorphic filter [28] or Retinex [21]. This model assumes that the intensity at any pixel, which is the amount of light reflected by a point on the object, is the product of the illumination of the scene and the reflectance of the object(s).

$$F(x, y) = L(x, y)R(x, y) \tag{4}$$

F is the image, L is scene illumination, and R is the scene reflectance. R emerges from the properties of the scene objects, but L occurs from the lighting conditions at the time of image capture [28]. Illumination varies slowly (Low-frequency) across the image as compared to reflectance which can change considerably at object edges (High-frequency). This difference enables the split-up of the illumination component from the reflectance component. The first step is to transform the multiplicative components into additive components by moving into the log domain (ln) by:

$$F(x, y) = L(x, y)R(x, y) \tag{5}$$

$$ln(F(x, y)) = ln(L(x, y)R(x, y)) \tag{6}$$

$$ln(F(x, y)) = ln(L(x, y)) + ln(R(x, y)) \tag{7}$$

Then a low-pass filter (LPF) can be used to remove high frequencies (reflectance) and keep the low frequencies (illumination). However, LPF tends to blur edges, and the use of Homomorphic filtering and Retinex leads to the "halo" effects [5] because these techniques assume that the illumination component is globally smooth, and this assumption is not always correct.

Due to the above-mentioned limitations, Illumination-Reflectance model used in image enhancement can not be used directly in image splicing detection to achieve efficiency. Therefore, we consider to filter Y from YCbCr or V from HSV color space with an edge-preserving filter and take the resulting image as the estimation of the illumination component. It is reasonable because Y component is related to intensity information in image and V component is directly related to the brightness information. In our experiments, using Y gave a slightly better accuracy than using V.

To tackle the use of LPF, it is reasonable to adopt an edge-preserving filter such as Bilateral filter [32] or Guided Filter [13] instead of LPF such that clear edges are kept between surfaces under different lighting conditions, and details within a single surface are blurred. To make a trade-off between the "halo" effect and the detail removal ability, it is ideally suited to construct a fast, approximate edge-preserving filter with the max/mini-filters since Bilateral and Guided filters exhibit some limitations when dealing with real-time applications and the "halo" effect [5]. Max/mini-filters are given by:

$$f(x, y) = max_{(i,j)\in S_{xy}}\{K(i, j)\}, \quad f(x, y) = min_{(i,j)\in S_{xy}}\{K(i, j)\} \tag{8}$$

To achieve a high running speed, there are very fast implementations of max/mini-filters [5]. Thus, we used Y with the max/mini-filters to extract the illumination component as shown in the algorithm 1.

Fig. 4 Illumination component estimation on authentic(not tampered) image

Algorithm 1 the pseudocode of illumination estimation.

Result: Estimated Illumination
Input I:(R,G,B) ← Y CbCr ;
foreach *pixel* $x \in Y$ **do**
    $J(x) \leftarrow GetMaxValueInTheMask$
foreach *pixel* $x \in J$ **do**
    $L(x) \leftarrow GetMinValueInTheMask$
$L' \leftarrow a * (L + t)$;
$R \leftarrow I/L'$

L' is estimated illumination, R is reflectance, $a = 1.1$ is a constant slightly larger than 1 used to avoid the resulting image being too bright. $t = 0.05$ is a small positive number used to prevent zeros in division [5]. Figure 4 illustrates the illumination component from authentic image, whereas Fig. 5 shows the illumination component from tampered (spliced) image.

Fig. 5 Illumination component estimation on tampered (spliced) image

3.3 Local binary patterns (LBP)

LBP is a powerful feature for texture classification [11]. Image texture gives us information about the spatial arrangement of color or intensities in an image or selected region of an image. Given p_c as a central pixel value, P number of neighborhood pixels and r the radius of the neighborhood, LBP is calculated by:

$$LBP_{p,r} = \sum_{i=1}^{p-1} S(p_i - p_c).2^i \tag{9}$$

$$S(p_i - p_c) = \begin{cases} 1 : p_i \geq p_c \\ 0 : p_i < p_c \end{cases} \tag{10}$$

3.4 Tampering detection

To categorize an input image as authentic or spliced, we used the LBP normalized histogram as the feature vector for classification. A standard approach is to run multiple classifiers and compare their performance against one another and select the classifier which exhibits better performance. The authors in [22] provide easy access to numerous different classification algorithms such as K-Nearest Neighbors (KNN) [16], Support Vector Machine(SVM) [10], Linear Discriminant Analysis(LDA) [30], Logistic Regression (LR) [23], Decision Tree Classifiers (Dtree) [12], and Naive Bayes (NB) [22]. Figure 6 shows the LBP histogram from an input image.

4 Experiments and results

In this section, the dataset used is described in Section 4.1, evaluation metrics are presented in Section 4.2, platform and the analysis of running time are discussed in Section 4.3. The experiment results are reported in Section 4.4, misclassification cases are discussed in Section 4.5, and comparative results are reported in Section 4.6.

4.1 Dataset and evaluation metrics

The images used in the experiments are from the dataset CASIA v2.0 [7]. The total number of images is 12614 of which 7491 are authentic colored images and 5123 are forged colored

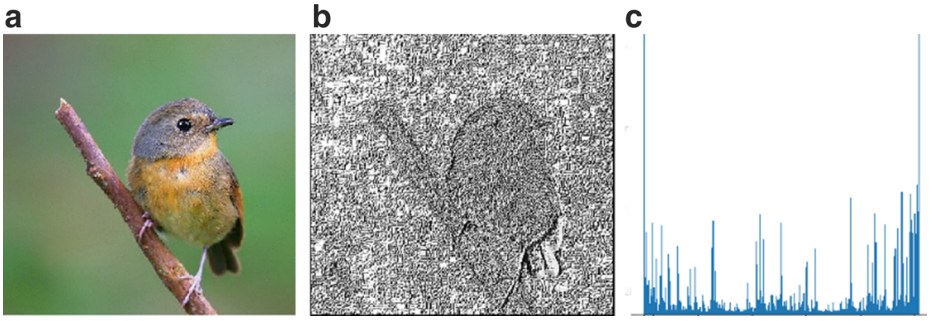

Fig. 6 (**a**) Input Image. (**b**) LBP. (**c**) LBP histogram

images in various kinds of formats that include JPEG, BMP, and TIFF. Images sizes (width, height) vary from 240 × 160 to 900 × 600. This dataset is more challenging because it contains mostly low-resolution images and introduces post-processing on the boundary area of tampered regions. Figure 7 plots the distribution of the image in the authentic and forged categories.

4.2 Evaluation metrics

The metrics to assess the performance of an image forgery detection technique include: forged images (f_i) correctly classified are t_p. Images wrongly classified as forged being authentic (A_i) are termed as f_p. Tampered images that are falsely missed are termed as f_n and authentic images correctly classified are t_n. tp_r is true positive rate, fp_r is false positive rate. The Receiver Operating Characteristic (ROC) curve [4] is created by plotting the true positive rate against the false-positive rate at various threshold settings. Accuracy acc is the combined fraction of true positives and negatives in the entire test set Precision (p_r) denotes the probability that a detected forgery is truly a forgery, while Recall (r_c) shows the probability that a forged image is detected as being forged. Finally, f_1 score combines p_r and r_c in a single value [20].

$$tp_r = \frac{\#\, t_p}{\#\, f_i}, \quad fp_r = \frac{\#\, f_p}{\#\, A_i}, \quad acc = \frac{tp + tn}{t_p + t_n + f_p + f_n} \tag{11}$$

$$p_r = \frac{tp}{t_p + f_p}, \quad r_c = \frac{t_p}{t_p + f_n}, \quad f_1 = 2\frac{p_r r_c}{p_r + r_c} \tag{12}$$

4.3 Experimental platform and the analysis of running time

The hardware used during the experiments is a Desktop with Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz, 64-bit processor with 8GB RAM. The software environments are python 3.6, Scikit-learn 0.21.2, skimage 0.16.2, and Ubuntu 18.04.3 LTS OS. The experiments are performed on 4000 images, and full dataset 12614 images. Table 2 reports the running time on 12614 images from dataset CASIA v2.0. The training size is 60% and the test size is 40%. We used various classifiers such as LR, SVM, KNN, LDA, Dtree, and NB because we tested three different feature vector size: 256, 512, and 768. The LBP and Maxi/Mini-filters parameters are found empirically by experiments on the CASIA v2.0 dataset, LPB parameters are neighbour set points = 32, radius= 3, method = uniform, and Maxi/Mini-filters windows size = 9.

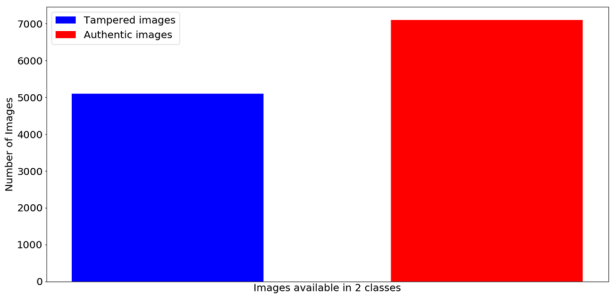

Fig. 7 Data distribution in dataset CASIA v2.0

Table 2 Analysis of running time. M is minutes, and S is seconds. 12614 images from dataset CASIA v2.0 are used

Feature vector size	Feature extraction (M)	Training time (M)	Prediction time (S)
256	42.18	1.50	13.42
512	55.0	2.0	20.44
768	76.10	3.30	52.61

As we mentioned earlier, we used various classifiers. Thus, the reported training time, and prediction time, are for all classifiers combined.

4.4 Experiments

Different machine learning algorithms such as SVM, KNN, LDA, Decision Tree, and Naive Bayes are used with our model for training and testing. All experiments are performed on images from CASIA v2.0 public benchmark dataset for forgery detection. Table 3 reports the accuracy of the model when the feature vector size is 256 and Fig. 8(a) is the graphical plot of its ROC curve. Table 3 reports the accuracy of the model when the feature vector size is 512 and Fig. 8(b) is the graphical plot of its ROC curve. Table 3 reports the accuracy of the model when the feature vector size is 768 and the ROC curve is shown in Fig. 8(c).

Table 3 Accuracy when the feature vector size is 256, 512, and 768, on 12614 images from CASIA v2.0. Testsize=0.4

Classifiers	Accuracy (%)
Accuracy when the feature vector size is 256	
Logistic Regression	62.20
Decision Tree	58.30
K Nearest Nearbor	60.89
Linear Discriminant Analysis	**65.83**
Naive Bayes	55.52)
Support Vector Machines	59.27
Accuracy when the feature vector size is 512	
Logistic Regression	89.13
Decision Tree	88.58
K Nearest Nearbor	86.08
Linear Discriminant Analysis	**92.98**
Naive Bayes	76.55
Support Vector Machines	90.22
Accuracy when the feature vector size is 768	
Logistic Regression	90.82
Decision Tree	91.07
K Nearest Nearbor	86.08
Linear Discriminant Analysis	**93.79**
Naive Bayes	76.93
Support Vector Machines	91.45

Bold entries are used to highlight the best results

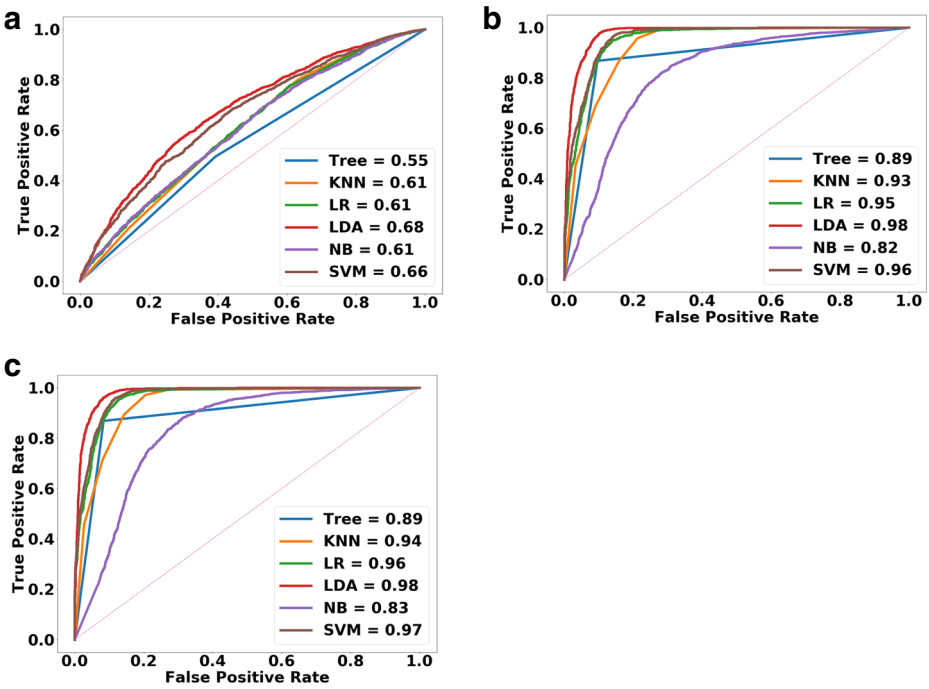

Fig. 8 ROC curves: (**a**) Feature vector size = 256. (**b**) Feature vector size = 512. (**c**) Feature vector size = 768

Fig. 9 illustrates some spliced forged image successfully classified as forged.

4.5 Misclassification cases

Many cases of misclassification are authentic images categorized as tampered images as shown in Fig. 10 and reported in Table 4.

Misclassification causes include, small size images, photographs taken with background blur effect, and uniform background. Also, spliced images are subjected to post-processing operations such as blurring, filtering, scaling, rotation, and different image file formats.

Fig. 9 Spliced images successfully classified as forged

Fig. 10 Authentic images misclassified as tampered images

4.6 Comparative results

From the results obtained in Section 4.4 it is noticeable that our method performs better when the classifier used is LDA because image splicing datasets are very complex distributions and LDA as a generative approach to model input distribution it does model complexities in distribution. Therefore, it gives a slight better performance than discriminant model such as SVM for this case.

The accuracy obtained are **94.59** on 4117 images, and **93.79** on 12614 images. We also compare the performance of our method against other methods known in the literature and the results are reported in Table 5.

Table 4 reports the tn, fp, fn, tp, p_r, r_c, and f_1 score detection results obtained on 12614 images from dataset CASIA v2.0.

The results from Table. 4 show that the method [9] presents better f_1 score. However, this method is very computationally expensive than our method (see Table 5), it uses deep learning (6 convolution layers with 600 kernels) and exhibits the deep learning drawbacks discussed in Section 2.

The results reported in Table 5 indicate that our method is more computationally efficient in term of feature vector size. The methods [26], [18] exhibit better accuracy. However, they are computationally expensive than our method since the former requires a massive

Table 4 tn, fp, fn, tp, p_r, r_c, and f_1 score, on 12614 images from CASIA v2.0. Testsize=0.4

Classifiers	tn	fp	fn	tp	precision%	Recall%	F1 score
Logistic Regression	2734	275	192	1845	93	91	92
Decision Tree	2757	252	251	1786	92	92	92
K Nearest Nearbor	2582	427	199	1838	93	86	89
Linear Discriminant Analysis	**2789**	**220**	**96**	**1941**	**97**	93	95
Naive Bayes	2268	741	438	1599	84	75	79
Support Vector Machines	2734	275	191	1846	**97**	90	93
Deep Learning + HWT [9]	-	-	-	-	**97**	**99**	**98**
DWT + LBP [17]	-	-	-	-	-	91.87	-
Markov +DCT+DWT [37]	-	-	-	-	-	92.5	-

Bold entries are used to highlight the best results

Table 5 Performance comparison (feature vector size and accuracy) against other methods known in the literature on dataset CASIA v2.0

Models	Features extraction	Vector size	*acc* (%)
CNN+RGB [26]	Pre-trained CNN	16384	**97.83**
	Feature fusion	400	
Markov+ DCT + DWT [37]	-	-	89.76
Fast SCNN[36]	SCNN	25088	85.83
Improved double quantization [31]	-	-	79.72
STP + LBP [18]	LBP Histogram	3,584 to 480	97.33
Deep Learning + HWT [9]	HWT	4,096 to 1,024.	96.36
Proposed Method	LBP Histogram	**256**	65.83
		512	92.98
		768	94.59

Bold entries are used to highlight the best results

data samples to train the discriminate CNN model, and the latter uses the steerable pyramid transform which involves multi-scale, multi-orientation image decomposition with the number of pixels in the pyramid being much greater than the number of pixels in the input image and requires a combination of two different data reduction techniques to reduce the complexity of the model.

5 Conclusion

In this paper, we presented an efficient image splicing detection method. The proposed method uses the illumination component, chroma channel, and Local Binary Patterns to distinguish authentic images from forged. The input image is first transformed in luminance and Chroma, we extract the illumination component from luminance using an approximate edge-preserving filter with a max/mini-filters through Illumination-reflectance model. Finally,   we compute the LBP normalized histogram from illumination and chroma and we take it as a feature vector for classification using different machine learning algorithms. Extensive experiments show that the proposed model is computationally efficient and effective for splicing forgery detection. Our future work will be to localize the doctored regions/areas in the spliced image.

References

1. Alahmadi A, Hussain M, Aboalsamh H, Muhammad G, Bebis G (2013) Splicing image forgery detection based on dct and local binary pattern, 253–256 https://doi.org/10.1109/GlobalSIP.2013.6736863
2. Bayar B, C. Stamm M (2016) A deep learning approach to universal image manipulation detection using a new convolutional layer, 5–10, https://doi.org/10.1145/2909827.2930786
3. Bharati A, Singh R, Vatsa M, Bowyer KW (2016) Detecting facial retouching using supervised deep learning. IEEE Transactions on Information Forensics and Security 11:1903–1913. https://doi.org/10.1109/TIFS.2016.2561898

4. Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recogn 30:1145–1159. http://www.sciencedirect.com/science/article/pii/S0031320396001422
5. Chen D, Lan S, Xu P, Zhang Y (2016) "Illumination-Reflectance Based Image Enhancement Method for Character Recognition." 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) (2016): 207–211.
6. Chun-su, Nam T, Jong-weon L, Goo-Rak K (2019) Efficient image splicing detection algorithm based on markov features. Multimedia Tools and Applications 78:12405–12419. https://doi.org/https://doi.org/10.1007/s11042-018-6792-9
7. Dong J, Wang W, Tan T (2013) Casia image tampering detection evaluation database, pp 422–426
8. Drira F, Denis F, Baskurt A (2004) Image watermarking technique based on the steerable pyramid transform, 165–176, https://doi.org/10.1117/12.578741
9. Eman IAE-L, Ahmed T, Hala HZ (2019) A passive approach for detecting image splicing using deep learning and haar wavelet transform. I. J. Computer Network and Information Security 5:28–35. https://doi.org/10.5815/ijcnis.2019.05.04
10. Evgeniou T, Pontil M (2001) Support vector machines: Theory and applications. In: Paliouras G, Karkaletsis V, Spyropoulos CD (eds) Machine Learning and Its Applications. ACAI 1999. Lecture Notes in Computer Science, vol 2049. Springer, Berlin, Heidelberg, pp 249–257. https://doi.org/10.1007/3-540-44673-7_12
11. Fahime H, Mahdi H, Farhad G (2015) Image splicing forgery detection using local binary pattern and discrete wavelet transform. In: 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, pp 1074–107
12. Gupta B, Rawat A, Jain A, Arora A, Dhami N (2017) Analysis of various decision tree algorithms for classification in data mining. International Journal of Computer Applications 163:15–19. https://doi.org/10.5120/ijca2017913660
13. He K, Sun J, Tang X (2013) Guided image filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(6):1397–1409. https://doi.org/10.1109/TPAMI.2012.213
14. Isaac M, Wilscy M (2018) Image forgery detection using region-based rotation invariant co-occurrences among adjacent lbps. Journal of Intelligent and Fuzzy Systems 34:1679–1690. https://doi.org/10.3233/JIFS-169461
15. Jawadul HB, Amit KR-C, Jason B, Lakshmanan N, BS M (2017) Exploiting spatial structure for localizing manipulated image regions. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 4970–4979
16. Laaksonen J, Oja E (1996) Classification with learning k-nearest neighbors. In: Proceedings of International Conference on Neural Networks (ICNN'96), vol 3, pp 1480–1483
17. Mandeep K, Savita G (2016) A passive blind approach for image splicing detection based on dwt and lbp histograms, in the proceedings of international symposium on security in computing and communication, chandigarh, india
18. Muhammad G, Al-Hammadi M, Hussain M, Bebis G (2014) Image forgery detection using steerable pyramid transform and local binary pattern. Mach Vis Appl 25:985–995. https://doi.org/10.1007/s00138-013-0547-4
19. Niyishaka P, Bhagvati C (2018) Digital image forensics technique for copy-move forgery detection using dog and orb. In: Chmielewski L, Kozera R, Or??owski A, Wojciechowski K, Bruckstein A, Petkov N (eds) Computer Vision and Graphics. ICCVG 2018. Lecture Notes in Computer Science, vol 11114. Springer, Cham. https://doi.org/10.1007/978-3-030-00692-1_41
20. Niyishaka P, Bhagvati C (2020) Copy-move forgery detection using image blobs and brsik feature. Multimed Tools Appl 79:26045-26059. https://doi.org/10.1007/s11042-020-09225-6
21. Parihar AS, Singh K (2018) A study on retinex based method for image enhancement. In: 2018 2nd International Conference on Inventive Systems and Control (ICISC), pp 619–624
22. Pedregosa F, Varoquaux G, Gramfort A, Michel V (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830
23. Peng J, Lee K, Ingersoll G (2002) An introduction to logistic regression analysis and reporting. Journal of Educational Research - J EDUC RES 96:3–14. https://doi.org/10.1080/00220670209598786
24. Peng Z, Xintong H, Morariu VI, Larry S. D (2018) Learning rich features for image manipulation detection. arXiv:1805.04953
25. Rachna M, Navneet A (2019) Image splicing detection with markov features and pca, International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering. IJIREEICE, 7
26. Rao Y, Ni J (2016) A deep learning approach to detection of splicing and copy-move forgeries in images, 2016 IEEE International Workshop on Information Forensics and Security (WIFS), 1–6

27. Redi JA, Taktak W, Dugelay J-L (2011) Digital image forensics: a booklet for beginners. Multimedia Tools and Applications 51(1):133–162. https://doi.org/10.1007/s11042-010-0620-1. https://doi.org/10.1007/s11042-010-0620-1,
28. Steve E (2013) Homomorphic filtering, Accessed: 2020-01-19
29. Steven B (2014) Why luminance is the key component of color, Accessed: 2020-06-20. https://vanseodesign.com/web-design/color-luminance/
30. Tharwat A, Gaber T, Ibrahim A, Hassanien AE (2017) Linear discriminant analysis: A detailed tutorial. Ai Communications 30:169–190,. https://doi.org/10.3233/AIC-170729
31. Thing VLL, Chen Y, Cheh C (2012) An improved double compression detection method for jpeg image forensics. In: 2012 IEEE International Symposium on Multimedia, pp 290–297
32. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: ICCV, pp 839–846. citeseer.ist.psu.edu/tomasi98bilateral.html
33. Wei Wang, Dong J, Tan T (2009) Effective image splicing detection based on image chroma. In: 2009 16th IEEE International Conference on Image Processing (ICIP), pp 1257–1260
34. Willy W (2020) Human vision and color, Accessed: 2020-01-20. https://biomachina.org/courses/imageproc/121.pdf
35. Yuan R, Jiangqun N, Huimin Z (2020) Deep learning local descriptor for image splicing detection and localization. IEEE Access 8:25611–25625. https://doi.org/10.1109/ACCESS.2020.2970735
36. Zhang Z, Zhang Y, Zhou Z, Luo J (2018) Boundary-based image forgery detection by fast shallow cnn, 2658–2663
37. Zhongwei H, Wei L, Wei S, Jiwu H (2012) Digital image splicing detection based on markov features in dct and dwt domain. Pattern Recogn 45:4292–4299