Color noise-based feature for splicing detection and localization

Christophe Destruel, Vincent Itier, Olivier Strauss, William Puech

Color noise-based feature for splicing detection and localization

Christophe Destruel, Vincent Itier , Olivier Strauss and William Puech
LIRMM, Univ. Montpellier, CNRS, France

***Abstract*—Images that have been altered and more specifically spliced together have invaded the digital domain due to the ease with which we are able to copy and paste them. To detect such forgeries the digital image processing community is proposing new automatic algorithms designed to help human operators reveal manipulated images. In this paper, we focus on a local detection system, which considers which tampered areas produce local statistical effects that do not impact neighboring areas or the image as a whole. We propose to study how the definition of local blocks, considering their size and overlap, impacts final pixel detection. We also propose new features which are an original way to consider the noise of an image as a colored signal. Indeed, in a non-forged image, there is a high correlation of noise between the three color channels R, G and B. We show that an optimal configuration can be defined and in this case the proposed approach outperforms several previously proposed methods using the same tested dataset, in uncompressed and JPEG modes. Note, in this paper we only focus on feature extraction without using machine learning.**

I. Introduction

Today it is almost a platitude to say that everyone, once in their life, will be mislead by an altered digital image. Image manipulation software is now widely available and make it possible for anyone to produce fake images. The daily flow of images is continuous and the work to identify which images have been tampered with becomes very complex and time consuming. A large number of image modifications can be considered: from simple transformations (contrast enhancement, resize and resample) to more malicious examples (copy/move from a single image -cloning- and copy/paste between several images -splicing- *etc*.). To try to detect these modifications, several clues can be investigated.

The first source of traces that can be addressed is the acquisition device itself. The imperfections of any optical system is part of the full system and impacts the final obtained image. Under given conditions it is possible to detect variations in chromatic aberrations [1], in the underlaying noise [2] or in color interpolated pixels [3]. The camera response function (CRF) can also provide evidence of splice manipulations. It converts captured irradiance values into pixel brightness and is deeply linked to the physical elements that composes a camera. In [4], checking the consistency is made using the segments highlighted by the segmentation process: after a classification step, suspicious segments allow the automatic system to detect a suspicious image.

The method used to encode the image is another useful source of clues. Images compressed with JPEG format are widely used and are very susceptible to anti-forgery mechanisms. The nature of this compression algorithm introduces traces and small alterations in the produced images that can be used directly [5] or in the spatial domain [6]. The "blockiness" nature of JPEG can also be used to detect an image that has been double compressed by considering aligned or non-aligned [7] merged blocks. The physical captured scene itself can be considered as a source of clues and several investigations try to identify elements in the real world to detect forgeries. To be captured, a scene has to be illuminated and the 3D position of the lights can be detected to prove inconsistencies [8]. The projection function defines the point of view and gives a unique look to a captured scene: an incorrect or deformed perspective can be used to reveal a modified image [9].

In this paper, we propose several new features based on RGB noise to detect spliced images and to classify altered pixels. In the same way as methods based on the recapture of CCD or mosaicking, we focus on the inconsistency in color noise correlation. In the case of a spliced image the high correlation between channels is broken. We analyze how such features are sensitive to sliding window size and sliding step, which is used in the detection process.

In Section II, we propose a noise color-based feature to detect if images and pixels have been tampered with. Section III presents detection results over a test dataset and a study about its sensitivity to block size and block overlapping. In Section IV we conclude and discuss future work.

II. Tampered image detection

Images can be forged in many different ways. In this paper we focus on splicing forgery: a region of an external source is pasted into a host image which then becomes forged. Of course it can be done using several regions from several external sources. The aim of such a forgery is to either hide details or to add new elements to the original image.

We do not consider general features over the whole doctored image, but we focus on locally detectable variations in the images statistics introduced by the splicing process. To detect these variations, the suspicious image is split into regular windows and each feature is processed separately in an individual window. Each sliding window is then described by features and any inconsistency of the noise, highlights a probable act of forgery. In any local approach, the definition of the local concept is a main preliminary state. A balance has to be found between large areas, which contain a huge amount of information, but provide a poor final resolution and small areas which provide good resolution, but have to deal with less information. In this paper we will present how we use a sliding window to enhance the final resolution and to maintain a sufficiently high level of available information.

In Section II-A we present how the color noise is extracted and how it is used. In Section II-B we explain the sliding window approach and how to compute using the proposed features. Section II-C presents the image classification process and Section II-D is about the pixel classification process.

A. Overview of color noise feature extraction

Images are mainly in color, and since the recapture of CCD and mosaicking produce an interplay noise between channels, we focus on the inconsistency in color noise correlation. The main idea is to propose new features which are able to highlight and characterize the correlation between RGB channels of image noise. We propose two specific features that can be used as a tiny fingerprint to differentiate two parts of an image acquired by two different devices or in different conditions.

Fig. 1 presents the full process used to compute these features. At each pixel i of the tested image, we independently extract noise from each RGB channel by processing the difference between the image and its median filtered version. We obtain three noise values – namely r_i, g_i, b_i for each pixel. The features we propose are based on characterizing the statistical properties of those noise values in a sliding window. The noise related feature κ_k is used to decide if an image is forged or not, and the Eigen related feature C_k is used to localize the spliced area.

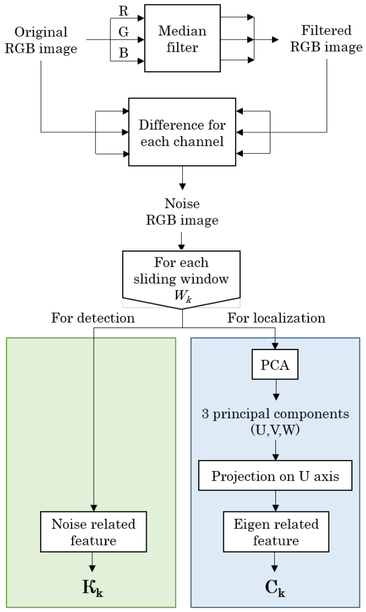


Fig. 1: Overview of the proposed method: two proposed features are computed over the noise color image using sliding windows.

B. Sliding window analysis for feature extraction

The objective is to extract local features that consider the pixels' neighbourhood, we propose to use a sliding window. Therefore, the analysis we propose is based on splitting the considered image of $m \times n$ pixels into K sliding windows $(W_k)_{k=1 \ldots K}$ measuring $w \times w$. Sliding windows are shifted by a step measuring s such as $s \leq w$. In order to avoid border problems we only consider entire sliding windows, then $K = \frac{mn}{s^2} - w(n+m)$. The sliding step size defines the sampling size of the resulting heat map and is a trade-off between precision and resolution. A small size focuses on local variability, whereas a larger size focuses on global features in the area.

Both parameters w and s define a particular configuration noted $(w; s)$.The sliding window size defines the amount of information used to estimate the feature, while the step size defines the precision of the final detection on the corresponding pixel in the image. Naturally, estimating features over samples of different sizes leads to different results.

C. Statistical analysis of the color noise cloud

We propose a noise related feature which can be seen as a fourth central statistical moment. It is dedicated to deciding whether the image under analysis has been forged or not. A value κ_k of this feature is computed with the point cloud of each sliding window W_k. The purpose of this feature is to find blocks that overlap the authentic and the spliced area. We use the variance of the euclidean distances between each point of the point cloud to its barycenter in order to characterize its dispersion. The distance histogram of an overlapping area is generally more spread out than the distance histogram of a uniform area. Let us suppose that each triplet $p_i^{rgb} = (r_i, g_i, b_i)_{i=1 \ldots n=w^2}$ in the sliding window W_k are lexicographically ordered, then κ_k is:

$$\kappa_k = \frac{1}{n} \sum_{i=1}^{n} (d(p_i^{rgb}, \bar{p}^{rgb}) - \mu)^2 \ , \qquad (1)$$

where $d(x, y)$ is the euclidean distance and μ the average of the $d(p_i^{rgb}, \bar{p}^{rgb})$ values in W_k and $\bar{p}^{rgb} = (\bar{r}, \bar{g}, \bar{b})$ is the barycenter of the point cloud in the sliding window W_k: $\bar{p}^{rgb} = \frac{1}{n} \sum_{i=1}^{n} p_i^{rgb}$.

The set $(\kappa_k)_{k=1 \ldots K}$ is then analyzed. If, for any k, the distance of κ_k to the median value of the set is greater than an experimentally defined threshold, the k^{th} sliding window is marked as suspicious. An image with at least one suspicious sliding window is considered suspect.

D. Localization analysis of the color noise cloud

We propose an Eigen related feature dedicated to localizing parts of the images that have been forged (*i.e.* for pixel classification). For each sliding window W_k, we project the $(p_i^{rgb})_{i=1 \ldots n}$ on the main axis obtained by principal component analysis (PCA). The main idea is to characterize the local image properties by the noise dispersion along the principal axis in the RGB color space. This set is denoted $\mathcal{E} = \{(u_i)_{i=1 \ldots n}\}$.

We then select, in these projected values, those that belong to the interval $[u_2, u_7]$, where $(u_t)_{t=1...8}$ are the 8 octiles of $\mathcal{E}$. The resulting set, of size $q < n$, is denoted $\mathcal{E}'$ such that $\mathcal{E}' = \{u_i \in \mathcal{E}/(u_i \geq 2) \wedge (u_i \leq 7)\}$. The value C_k of the second feature attached to the sliding window W_k is:

$$C_k = \frac{1}{q}\sum_{i=1}^{q} u'_i,\ u'_i \in \mathcal{E}'. \tag{2}$$

As in the previous case, the distance between the median of $(C_k)_{k=1...K}$ and each C_k is used to detect if the k^{th} sliding window has been tampered with or not. The decision is given by comparing this distance to an experimentally defined threshold.

III. Experimental results

To evaluate the two proposed features, we used images from the "Columbia Uncompressed Image Splicing Detection Evaluation Dataset" [10]. It is composed of 183 original and uncompressed images, captured with four different cameras. There are 180 spliced images in the dataset which are the result of a copy/paste operation between two different original images. The image sizes range from 757×568 to 1152×768 pixels. Even if this database does not contain visually realistic spliced images, it is widely diffused and used for evaluation purposes. An example of a spliced image and its true map are shown in Fig. 2.

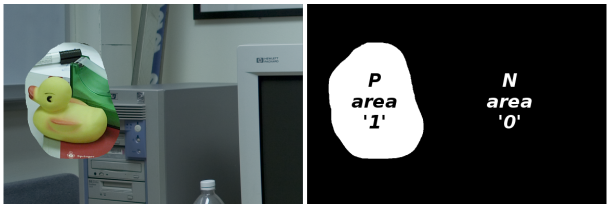


Fig. 2: Image test and its true map from the Columbia dataset [10].

A. Image classification (detection)

In this section, we present the capability of the proposed feature κ_k (see Eq. 1) to classify the images from the database. Fig. 3 illustrates the ROC curve of classification results for the 363 images of the dataset, taking into account three different block sizes. As shown in Section III-C, the best results are obtained with the $(32; 8)$ configuration which reaches 0.85 as Area Under the ROC Curve (AUC). Table I summarizes the classification results: our score (0.78 for precision and recall) outperforms the results presented in [4] (0.70 for both precision and recall).

The proposed feature is based on color characteristics, but they are still robust enough to be used on compressed images, even if the compression process reduces color information. To analyze the capability of our method to classify compressed images, we compress the original dataset in JPEG format using various compression quality factors. The JPEG compression is done with standard sub-sampling (4:2:2) basically it divides by 2 the block size by averaging neighbour pixel values, which is not a huge compression of chroma channels. The JPEG images are then decoded and the RGB channels are used in our classification algorithm.

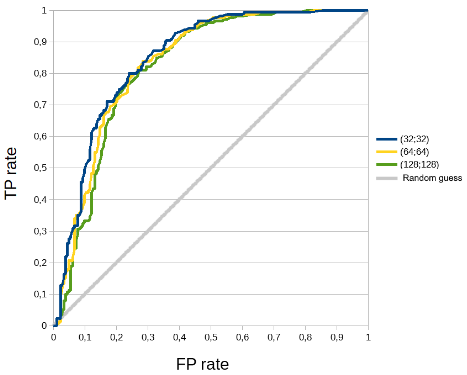


Fig. 3: ROC curves for the full dataset, processed without overlapping (sliding window size $w = s$) for 3 different sliding window sizes (32, 64 and 128).

Method	precision	recall
Hsu and Chang [4]	0.70	0.70
Our method	**0.78**	**0.78**
Our method - JPEG 50	0.73	0.68

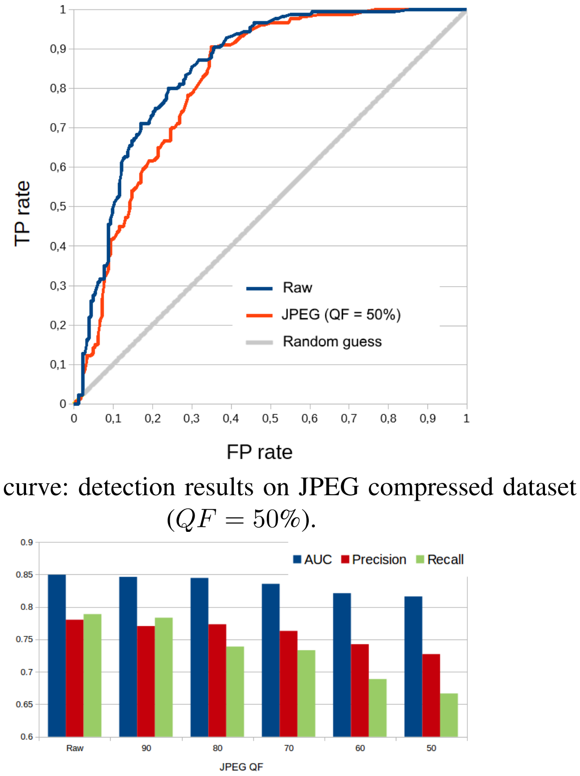
TABLE I: Comparison of image classification scores.

We performed tests using five different quality factors (QF): 50%, 60%, 70%, 80% and 90%. For each level, we consider the ROC curve (Fig. 4.(i)) and the AUC value (Fig. 4.(ii)). We also plot the precision/recall pair at the same threshold point (Fig. 4.(ii)). To be more realistic, the threshold is chosen as the best solution for the uncompressed dataset. Note, all values slightly decrease when compression increases. The JPEG algorithm is a lossy compression process and some information is lost, but results of the classification are still usable. With QF = 50%, this feature produces precision and recall values (respectively 0.73 and 0.68) which denote a good classification capability *cf.* Table I.

B. Pixel classification (localization)

This section is about the localization of forged pixels in an image which has been classified as a forged image and is based on the C_k (see Eq. 2) feature. The pixel classification is represented by a heat map that can be easily interpreted. Fig. 5 shows heat maps for different steps $(16, 32, 64)$ considering a single sliding window size of 64. This shows how the resolution of the detection increases when the sliding step decreases. Fig. 6 shows heat maps for different sliding window sizes $(8, 32, 64)$ using a fixed sliding step of 8 to highlight the impact of the sliding window size. Note that noise increases when the sliding window size decreases (and overlapping tends to 0).

To evaluate the ability of our method to classify the pixels of spliced images from the Columbia dataset, we process the



(i) ROC curve: detection results on JPEG compressed dataset ($QF = 50\%$).

(ii) Metric evolution with the JPEG quality factor.

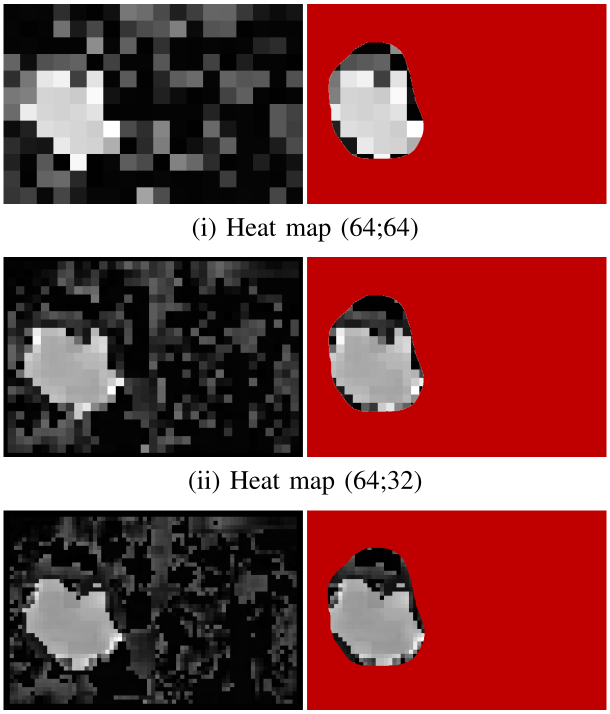
Fig. 4: JPEG compression effects - original non compressed images and compressed dataset processed with the same configuration $(32; 32)$.

standard F_1 and Matthews Correlation Coefficient (MCC) scores:

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} , \qquad (3)$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{P.(TP + FP) \cdot N \cdot (TN + FN)}} , \qquad (4)$$

where TP and TN are respectively the number of true positive and true negative, FP and FN correspond to the number of false positive and false negative. In the same way, we have the number of positive pixel $P = TP + FN$ and the number of negative pixels $N = TN + FP$. In [11], the authors performed a full comparison of the existing algorithm for pixel classification using the same dataset. We applied the same test parameters on this dataset to obtain comparable scores. Table II presents comparisons between our proposed approach and previous similar work. We succeeded in obtaining a score of $F1 = 51\%$ and $MCC = 39\%$ without using machine learning. Only work based on fully convolutional networks [11] and the modeling of local image noise variance by wavelet filtering [12] gave better results.

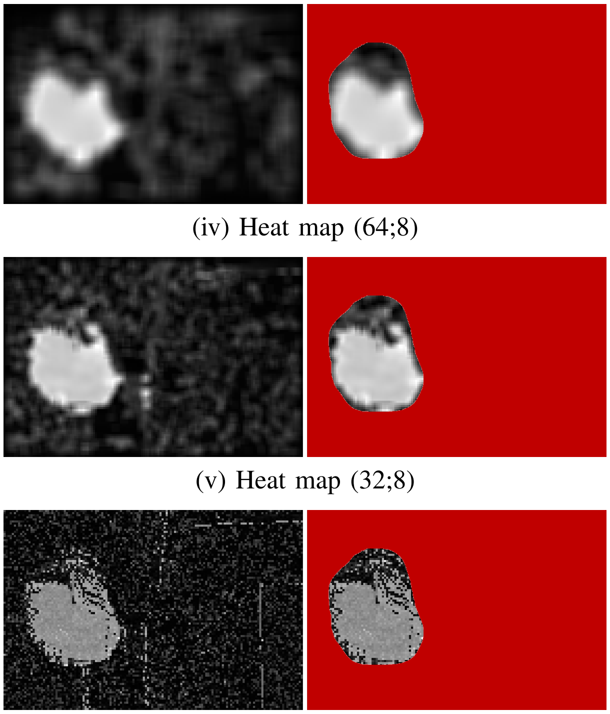


(i) Heat map (64;64)

(ii) Heat map (64;32)

(iii) Heat map (64;16)

Fig. 5: Image heat maps: sliding step variation with a window size of 64.



(iv) Heat map (64;8)

(v) Heat map (32;8)

(vi) Heat map (8;8)

Fig. 6: Image heat maps: sliding window size variation with a sliding step of 8.

Methods	F1	MCC
Our	**0.51**	**0.39**
CFA2 [13]	0.50	0.33
NOI3 [14]	0.45	0.21
ELA [15]	0.47	0.23
CFA1 [16]	0.47	0.23
NOI2 [17]	0.53	0.35
FCN [11]	*0.61*	*0.48*
NOI1 [12]	*0.57*	*0.41*

TABLE II: Comparison of pixel classification scores: CFA2: CFA simulation [13], NOI3: Co-occurence of high-frequency component [14], ELA: Error level analysis [15], CFA1: CFA patterns [16], NOI2: Local noise [17], FCN: Fully Convolutional Network [11], NOI1: Local noise [12].

C. Effects of sliding window size and sliding step

In this section we focus on the relationship between sliding window size, sliding step and pixel classification. In order to evaluate the sensitivity of the detection of these parameters, we carried out detection tests on a set of spliced images, varying the sliding window size and the sliding step. We chose a set of parameters to test, in order to sample the largest field of possibilities and to avoid an exhaustive approach. Each parameter of the configuration is chosen from the set of values $[4, 8, 16, 32, 64, 128]$ and we focus on two sets of configurations:

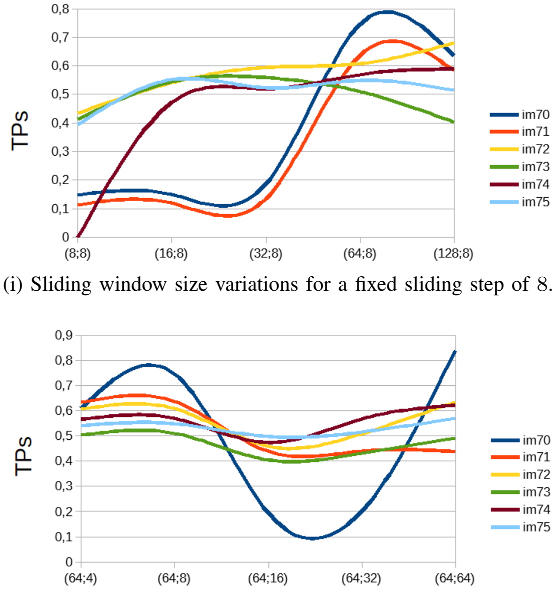
- Different sliding step values in $(4, 8, 16, 32, 64)$ for a given sliding window size (64).
- Different sliding window size in $(8, 16, 32, 64, 128)$ for a given sliding step (8).

For each selected configuration, we evaluate the ability of the feature to distinguish between authentic and tampered pixels. We propose to use a measure based on the proposed localization feature C_k (see Eq. 2) to evaluate the pixel classification. The feature values are first normalized between $[0, 1]$ over the whole image and denoted f_i, $0 \leq i \leq N$, where N is the number of pixels of the forged image (and of the true map). The true positive score TP_s is the average of this signal over the tampered areas. We note TM the pixel true map in which 1 marks the positive pixels and 0 marks the negative (original) pixels:

$$TP_s = \frac{\sum_{i=0}^{N-1} f_i \times \text{TM}_i}{\sum_{i=0}^{N-1} \text{TM}_i} \ , \tag{5}$$

Both values are between $[0, 1]$ and the best result is obtained when $TP_s \to 1$.

In this experiment the pixel classification is evaluated considering TP_s (Eq. 5) which is the mean of C_i signal over the positive area. We randomly choose six images from the dataset, and the image "$im71$", in Fig. 7, represents an image that is not detected as forged by the system. The other curves denote the difficulties to correctly choose the most efficient configuration. Majority of curves presented in Fig. 7.(i) show a logical behaviour and have a maximum value for a central sliding window size value between 16 and 64 pixels. We believe that small areas (8 and 16) do not contain sufficient information, while large areas (128) induce an averaged signal that makes the useful information hard to detect.



(i) Sliding window size variations for a fixed sliding step of 8.

(ii) Sliding step variations for a fixed sliding window size of 64.

Fig. 7: True Positive score (TP_s) variations as a function of sliding window size and sliding step .

On the other hand, Fig. 7.(ii) presents an unexpected zigzag shape. The first local maximum value (around the sliding step value 8) is fully compatible with the previous results: an optimal configuration can be defined at this point. As the sliding step decreases, the precision of the detection increases up to a maximum value. Beyond this point, increasing the sliding window size is inefficient: very small sliding step values induce a noisy solution that reduces final detection results. We have to investigate why all curves present an increasing slope whilst sliding step increases. An explanation for this effect can be found in the structure of the images we used: sliced images in the Columbia Dataset present large and uniform forged areas that may artificially overvalue results for these configurations. Finally, in Fig. 7, we can see that an optimal configuration can be selected at around $(32; 8)$.

IV. Conclusion and future work

In this paper we propose a study on sliding window effects (size and sliding step) over pixel classification results in a spliced image context. We also proposed a new method to automatically detect spliced images. This work is based on a local evaluation algorithm, considering local relationships between noise colors. The main objective is to detect significant local variations between feature values processed over small areas of a tested image. The method has been evaluated with the Columbia Uncompressed Image Splicing Detection Evaluation Dataset. We highlighted the incidence of sliding steps using a local algorithm and we showed that an optimal configuration can be found to improve final classification results. Our detection algorithm reaches performance levels of 0.78% for precision and recall, better than results from previous work (0.70 for [4]). We also tested this method with JPEG formatted images: we compressed the original dataset with various quality factors to evaluate the robustness of our method. Up to QF = 70% our tests show better performance levels than [4] for both precision and recall. With a quality factor QF = 50% our tests show performance levels of 0.73% precision and 0.67% for recall. The method shows convincing behavior with the Columbia database which is promising for testing on more realistic database with the experimentally obtained window size and threshold.

Although in this work we have focused on evaluating the proposed features by themselves, it is only natural to consider how it would perform in a machine-learning setup. Therefore, future work will be devoted to exploring how it can be allied with well known learning methods (*e.g.* random forest, support vector machine and convolutional neural networks).

Acknowledgment

We would like to thank the financial support of the ANR-16-DEFA-0001 OEIL (statistiques rObustEs pour l'apprentIssage Léger) research project of the French ANR/DGA challenge DEFALS (DEtection de FALSifications dans des images).

References

[1] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *MM&Sec '06: Proceedings of the 8th workshop on multimedia and security*. 2006, pp. 48–55, ACM.

[2] J. Fridrich, "Digital image forensics using sensor noise," *IEEE Signal Processing Magazine*, vol. 26, pp. 26–37, 2009.

[3] A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, 2005.

[4] Y.-F. Hsu and S.-F. Chang, "Camera response functions for image forensics: an automatic algorithm for splicing detection," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 816–825, December 2010.

[5] W. Luo, J. Huang, and G. Qiu, "JPEG error analysis and its applications to digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 480–491, September 2010.

[6] Z. Fan and R.L. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, February 2003.

[7] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 842–848, April 2012.

[8] E. Kee, J. F. O'Brien, and H. Farid, "Exposing photo manipulation with inconsistent shadows," *ACM Transactions on Graphics*, vol. 32, no. 3, pp. 28:1–12, september 2013, Presented at SIGGRAPH 2013.

[9] H. Yao, S. Wang, Y. Zhao, and X. Zhang, "Detecting image forgery using perspective constraints," *IEEE Signal Processing Letters*, vol. 19, pp. 123–126, March 2012.

[10] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *International Conference on Multimedia and Expo*, Toronto, Canada, 2006.

[11] R. Salloum, Y. Ren, and C.-C.-J. Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *ArXiv e-prints*, September 2017.

[12] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image and Vision Computing*, vol. 27, no. 10, pp. 1497–1503, 2009.

[13] N. N. D. Wandji, X. Sun, and K. M. Fah, "A survey of partition-based techniques for copy-move forgery detection," *The Scientific World Journal*, vol. 2014, pp. 1–13, July 2014.

[14] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *WIFS'15: 2015 IEEE International Workshop on Information Forensics and Security*, Rome, Italy, November 2015, pp. 1–6.

[15] N. Krawetz, "A pictures worth: Digital image analysis and forensics," in *Black Hat Briefings*, USA, November 2007, pp. 1–31, Online article on: http://www.hackerfactor.com/papers/bh-usa-07-krawetz-wp.pdf.

[16] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 1566–1577, 10 2012.

[17] S. Lyu, X. Pan, and X. Zhang, "Exposing region splicing forgeries with blind local noise estimation," *International Journal of Computer Vision*, vol. 110, no. 2, pp. 202–221, November 2014.