A PNU-Based Technique to Detect Forged Regions in Digital Images

Giuseppe Cattaneo[1], Umberto Ferraro Petrillo[2], Gianluca Roscigno[1(✉)], and Carmine De Fusco[1]

[1] Dipartimento di Informatica,
Università degli Studi di Salerno, 84084 Fisciano, SA, Italy
{cattaneo,giroscigno}@unisa.it, c.defusco1@studenti.unisa.it
[2] Dipartimento di Scienze Statistiche,
Università di Roma "La Sapienza", 00185 Roma, Italy
umberto.ferraro@uniroma1.it

Abstract. In this paper we propose a non-blind passive technique for image forgery detection. Our technique is a variant of a method presented in [8] and it is based on the analysis of the *Sensor Pattern Noise* (SPN). Its main features are the ability to detect small forged regions and to run in an automatic way. Our technique works by extracting the SPN from the image under scrutiny and, then, by correlating it with the reference SPN of a target camera. The two noises are partitioned into non-overlapping blocks before evaluating their correlation. Then, a set of operators is applied on the resulting *Correlations Map* to highlight forged regions and remove noise spikes. The result is processed using a multi-level segmentation algorithm to determine which blocks should be considered forged. We analyzed the performance of our technique by using a dataset of 4,000 images.

Keywords: Digital image forensics · Image integrity · Image forgery detection · Forgery localization · Pixel non-uniformity noise

1 Introduction

Nowadays, there are plenty of tools that allows even an inexperienced user to modify the content of a digital image without leaving a visible trace of alternation (i.e., *digital image forgery*). This practice may be harmful if used, e.g., to alter the digital evidences in a criminal trial or to support the spread of false news for political propaganda.

Several techniques have been proposed in the past years for detecting forged images. These typically work by searching for tracks released during the forgery process. Most of these techniques can determine if an image has been forged or not, but are not able to identify which parts of an image have been forged or require some sort of human intervention for this purpose (e.g. select a possibly forged area or its shape in advance).

S. Battiato et al. (Eds.): ACIVS 2015, LNCS 9386, pp. 486–498, 2015.
DOI: 10.1007/978-3-319-25903-1_42

In this paper we present a technique that is able to determine whether an image has been forged or not and, in case of forgery, it is able to automatically locate areas of the image that are likely to have been forged, provided that the camera originally used to take the image is available. It is a variant of the technique originally presented by Fridrich *et al.* in [8] and it is based on the analysis of the *Pixel Non-Uniformity* (PNU) noise, i.e., a characteristic noise of every camera. The ability of our technique for operating in an automatic and adaptive way has been obtained through a clever use of the multi-level segmentation process proposed by Otsu [12] and exploiting an accurate experimental calibration phase. Our technique supports the identification of *splicing*, *copy-move* and *inpainting* forgery operations.

Organization of the Paper. The rest of the paper is organized as follows. In Section 2 we briefly review the state of the art in the field of image forgery detection. In Section 3 we present and detail our technique. The assignment of several parameters for our technique has been done with an experimental calibration phase discussed in Section 4. In Section 5 we present the results of an experimental analysis involving our technique. Finally, in Section 6 we give some concluding remarks for our work.

2 State of the Art

The digital image forgery detection research field is concerned with the development of automatic or semi-automatic techniques able to determine any forgery of an image. These techniques can be *active* or *passive*. In the case of active techniques, a watermark placed initially in the image is useful to determine the authenticity of an image. Passive techniques cannot rely on the existence of any explicit signature on the image under scrutiny and, for this reason, they are often called *blind* techniques. In this paper we focus our attention on passive techniques.

Following the description presented in [13], we distinguish three general types of digital artifacts that can be used to detect the forgery of a digital image, regardless from the type of alteration: compression artifacts, alterations in the camera *Sensor Pattern Noise* (SPN) and re-sampling traces. Compression artifacts are caused by the encoding of a digital image through a *lossy* compression format, like the JPEG format. In this last case, we can detect forgeries by checking for the existence of the *Double Quantization* (DQ) effect in the histograms of the *Discrete Cosine Transform* (DCT) coefficients. Many of the algorithms using the DQ effect rely on the fact that the presence of this effect is caused by the initial compression that an image has undergone when saved for the first time, and by the second compression, performed when saving again the image after having forged it. One popular algorithm based on this approach is the one presented by Lin *et al.* in [11] and further discussed in [2,3]. This algorithm determines the authenticity of an image and it locates any forged region through a *probability map*; this is used to mark each 8×8 block of a target image with

a probability of being forged. The algorithm then extracts from this map some features needed to train a dichotomous *Support Vector Machine* (SVM) classifier used to determine if the overall image is likely forged or not.

The sensor pattern noise (SPN) is the noise left by digital sensors on the images they capture. This noise is mainly due to some imperfections derived from the manufacturing process of the sensor and, thus, can be used to recognize the images that have been captured using a particular digital camera. The analysis of this noise has been extensively used for solving problems related to identification of the digital camera used to take a picture (see, e.g., [1,4,9]). However, there are also several forgery detection approaches based on the analysis of this noise. These typically work by calculating the statistical correlation between the SPN extracted from the image under scrutiny and the *reference sensor pattern noise*, which is a sort of fingerprint of a digital camera. Forged areas can be localized as those lacking the corresponding SPN. The most relevant contribution is the one presented by Fridrich *et al.* in [8]. This method assumes that either the camera that took the image is available to the analyst or at least some other non-forged images taken by the camera are available. The method comes in two variants. The first variant requires the investigator to manually select in advance the region of the image that is suspected of having being forged, then it is calculated the statistical evidence that the region was tampered. The second variant attempts to automatically determine the forged region without assuming any a priori knowledge about its location, shape, or size. This region is determined as the area with the lowest pattern noise presence in the image. This is achieved by sliding over the image a set of basic shapes with different orientations and size, and accumulating the lowest correlation values. Once the region is localized, it is inspected using the original algorithm. The computational cost of this variant is proportional to the number of shapes and sizes used for locating the tampered region. An improvement of this method has been presented by Fridrich *et al.* in [5]. Here the authors introduced a correlation predictor working on small blocks and trained by taking into account the intensity of the images taken with a given camera, the textures existing in the images being analyzed and the flattened areas existing in these images due to image processing operations. The predictor is obtained from blocks coming from a few non-forged images from the same camera. The same authors in [6] have presented another improved version of the previous method, while Chierchia *et al.* in [7] have recently proposed a strategy to improve the resolution of SPN-based forgery detection techniques. Namely, they used a spatially adaptive filtering technique with weights computed over a suitable pilot image to improve the resolution of the technique when evaluating the correlation between the SPN of an input image with the reference SPN. In particular, a guided filtering approach is adopted, obtaining performance much superior to the reference technique when small forgery areas are involved.

3 Our Technique

Our technique is inspired from the method presented by Fridrich *et al.* in [8] and it is based on the analysis of a particular type of *Sensor Pattern Noise*, the *Pixel*

Non-Uniformity (PNU) noise. Let I be an image under investigation and C the camera that has been used to take it, our technique works by first extracting the PNU noise from I and by correlating it with an estimation of the reference sensor pattern noise (*Reference Pattern*, RP) of C. The correlation is evaluated by first partitioning the two noise images into non-overlapping blocks. Then, a *Correlations Map* is built by correlating the corresponding pair of blocks of the two images. The out coming map may be very noisy and may incorrectly consider as forged small boundary regions. To overcome these problems we first apply a smoothing filter to remove *noise spikes* (i.e., isolated pixels with exceptionally high or low intensity). Then, we use an *opening operator* to further highlight large regions having homogeneous correlation values while leaving out smaller boundary regions. Once the Correlations Map has been post-processed, we use a multi-level segmentation algorithm to adaptively compute a set of thresholds that will be used, in turns, to determine which blocks should be considered forged. More details about this technique are provided in the rest of this section, while a set of pictures showing the different steps of our technique on three sample images is available in Figure 1.

Camera Sensor Pattern Noise Extraction (Setup). In this initial step we calculate the reference pattern of the camera C using the technique presented in [9]. Let $\mathcal{D}_C$ be a set of authentic images taken with C, called enrollment set. We extract the PNU noise estimation RN_e existing in each image $e \in \mathcal{D}_C$ and, then, we apply a pixel-by-pixel average operation on these noises to obtain an approximation of the reference pattern RP_C. The PNU noise is extracted from an image by first denoising it and, then, by subtracting from the input image the denoised one.

In the rest of the paper, unless stated otherwise, we assume that all the considered images have the same resolution. In addition, we only work on the green channel of the RGB color space.

Correlations Map Extraction (Step 1). In this step we first partition RN_I (i.e. the PNU noise estimation of the image I under scrutiny) and RP_C in blocks of size $s \times s$ pixels. Thus, if the two images have initially size $M \times N$, the partitioning will return two matrices RN'_I and RP'_C, composed of $M/s \times N/s$ blocks each. Then, we calculate the Bravais-Pearson Correlations Map between RN'_I and RP'_C using the approach defined in [9]:

$$corr(RN'_I(i,j), RP'_C(i,j)) = \frac{(RN'_I(i,j) - \overline{RN'_I(i,j)})(RP'_C(i,j) - \overline{RP'_C(i,j)})}{\left\|RN'_I(i,j) - \overline{RN'_I(i,j)}\right\| \left\|RP'_C(i,j) - \overline{RP'_C(i,j)}\right\|} \quad (1)$$

In Equation 1, the numerator is the covariance of $RN'_I(i,j)$ and $RP'_C(i,j)$, while the denominator is the product of their two standard deviations. The resulting index is in the range $[-1;+1]$, where values tending to $+1$ indicate that the block at index (i,j) has been taken by using C, while values tending to -1 indicate that it is not from C. This correlation is evaluated for each pair of

blocks $b(i,j)$ of RN'_I and RP'_C, where $i \in \{1,...,M/s\}$ and $j \in \{1,...,N/s\}$. The resulting values will be used to fill a Correlations Map matrix (CM).

Smoothing (Step 2). Since the Correlations Map is noisy, in this step, we apply a mean filter to reduce the noise in CM. This filter replaces each pixel value in an image with the average value of its neighbors, including itself (for details, see [15]). We call CM_{filt} the resulting Correlations Map.

Homogeneous Regions Highlighting (Step 3). In this step, we apply an *opening operation* [10] to remove some of the foreground (bright) pixels from the edges of regions of foreground pixels existing in CM_{filt}. This operation preserves foreground regions that have a shape that is similar to the chosen structuring element, or that can completely contain the structuring element, while eliminating all other regions of foreground pixels. In our case, the opening operation uses a *disk structuring element*. We define CM_{open} the Correlations Map resulting from the application of the opening operation to CM_{filt}.

Multi-level Segmentation (Step 4). In this step, we mark the regions of I that are considered to be forged by analyzing the content of CM_{open}. For this purpose, we first determine the correlation threshold below which blocks have to be considered forged, using the multi-level segmentation process proposed by Otsu [12]. Let l the number of distinct levels, the multi-level segmentation returns a vector T, sorted in ascending order, containing l different thresholds. We select among these thresholds the smallest, $l_{min} = min(T)$. Then, for each block $b(i,j)$, we mark it as 0 (i.e., forged), if $CM_{open}(i,j) \leq l_{min}$, and 1 (i.e., authentic), otherwise. The binary map obtained in this way is called $BinaryImage$. The benefit of using the multi-level segmentation algorithm is that a single fixed global threshold classifying the whole image in a binary way, could label as forged even blocks that are authentic, only because adjacent to forged blocks.

4 Experimental Calibration

The general technique presented in Section 3 required the calibration of several parameters. In our case, the assignment for these parameters was estimated through an empirical assessment conducted on a set of reference images.

4.1 Spatial Filter

The Correlations Map obtained during Step 1 may be very noisy. This may either be due to random noise or to other factors such as an oversaturated image or the presence of a strong texture in the portrayed scene. To overcome this problem we tried different smoothing filters and kernels. In the first case, we evaluated three different smoothing filters: gaussian filter (Figure 2b), median filter (Figure 2c) and mean filter (Figure 2d). The gaussian filter modifies the input signal by convolution with a gaussian function. The mean filter replaces

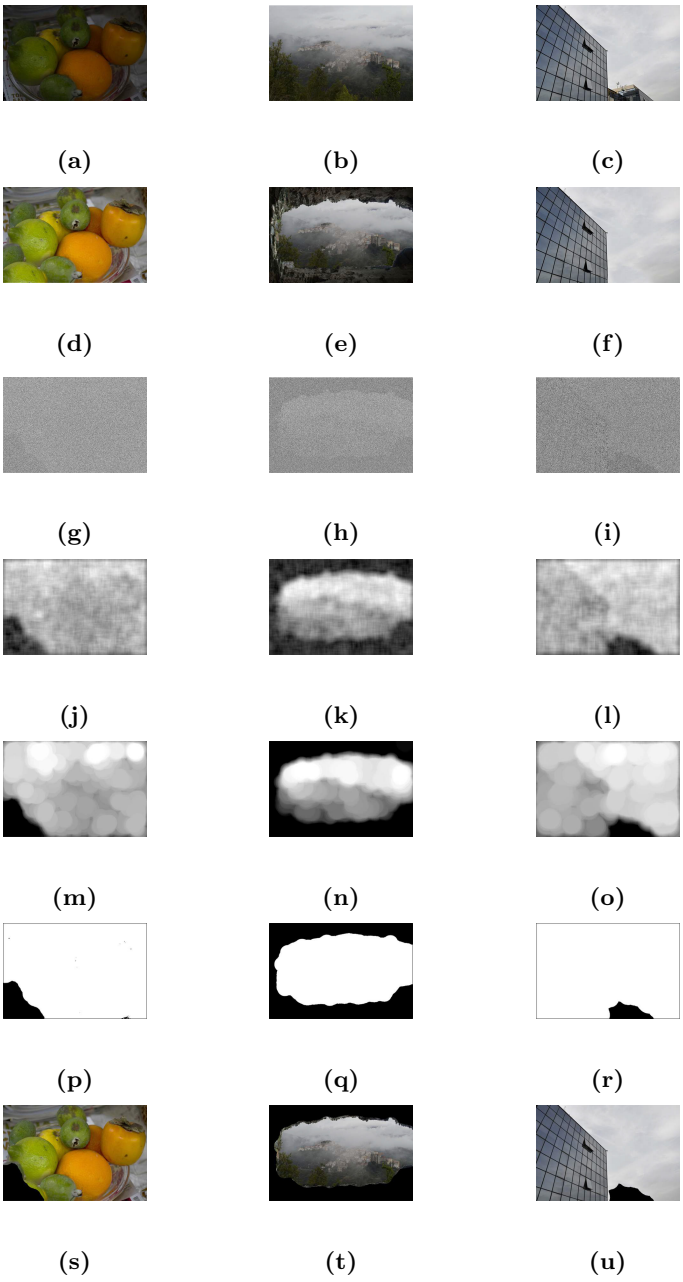

Fig. 1. (a, b, c) Authentic images. (d, e, f) Forged images with copy-move, splicing and inpainting respectively. (g, h, i) Correlations Maps (CMs). (j, k, l) Filtered Correlations Maps with mean filter (CMs_{filt}). (m, n, o) Filtered Correlations Maps after opening (CMs_{open}). (p, q, r) Binary Maps ($BinaryImages$). (s, t, u) Marked forged images ($MarkedImages$).

each pixel in an image with the average value of its neighbors, including itself. Finally, the median filter does the same, but by using the median value of the neighboring pixels. A qualitative analysis revealed that the median filter was the most effective in denoising the input image, but introduced artifacts near the edges of the image. The best trade-off was reached by using the mean filter.

Concerning the choice of the kernel size in the mean filter, we tried several different squared sizes for this parameter (i.e., 8×8, 16×16, 32×32 and 64×64). The results, presented in Figures 2e, 2f, 2g and 2h, show that when using values up to 16×16, the resulting map is still too noisy. If we use, instead, high values like 64×64, the resulting map is too smoothed because of the excessive approximation of the edges of the forged areas. Therefore we fixed 32×32 as the size of kernel to use in mean filtering.

4.2 Block Size

We recall that in our technique the pixels of RP and RN are partitioned into square blocks of size $s \times s$, before being analyzed. This introduces the problem about the block size to choose for this purpose. We investigated this problem through a qualitative analysis. We tried different assignments for s as reported in Figures 2i, 2j, 2k and 2l. Here we show the Correlations Maps obtained in Step 2 using various sizes of blocks ($s = 4, 8, 16, 32$). In these figures, each resulting Correlations Map has been denoised using a mean filter with kernel size 32×32. As it can be seen, if s is very small, the resulting Correlations Map is very noisy, even after the application of the mean filter. Instead, if s is very large, the Correlations Map loses too much detail because of the smoothing effect introduced by the mean filter. We found that the size giving the best results is $s = 8$.

4.3 Mathematical Morphology Operation

We decided to use the opening operation in our technique for separating objects that are poorly connected in the Correlations Maps while removing small regions. In our case, we determined that the best morphology operator to use is a disk kernel because it allows a more precise definition of the boundaries and of large forged regions than the other kernels. Then, we experimented with different sizes for this kernel, roughly equivalent to the radius of the disk kernel. According to our experimental results, shown in Figures 3a, 3b, 3c, 3d, 3e and 3f, the best performance are achieved when using a disk with radius 32. Using larger values would let the structuring element consider as part of the forged even adjoining regions that are not. Conversely, using values smaller than 32 would raise the probability of excluding some forged regions from the detection.

4.4 Segmentation Process

The multi-level segmentation algorithm used by our technique requires the indication of the number of levels to be used for partitioning the CM_{open}. We found

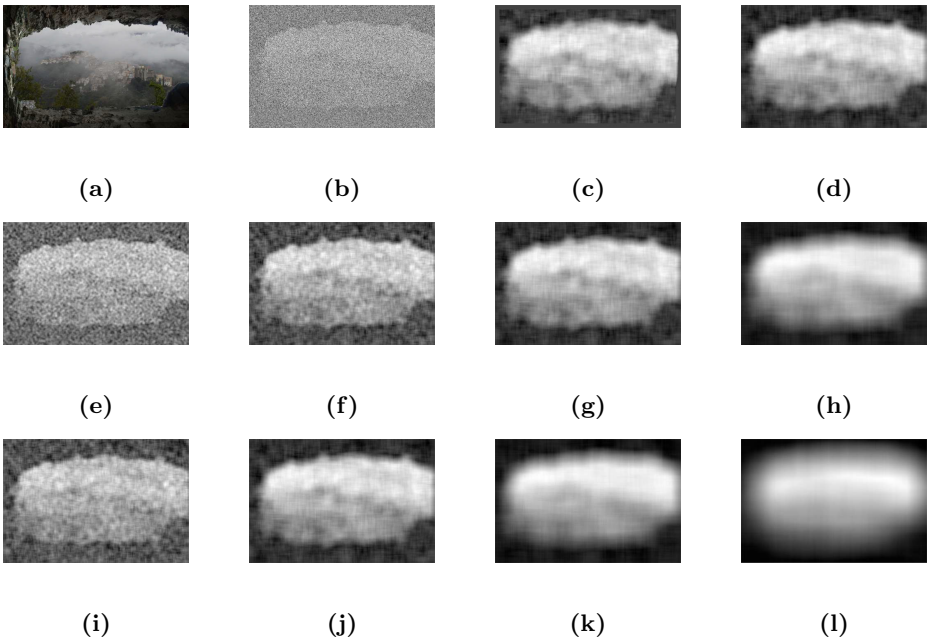

Fig. 2. (a) Forged image. (b, c, d) Gaussian, Median and Mean filter on Correlations Map. (e, f, g, h) Mean filters with kernel size $k \times k$, $k = 8, 16, 32, 64$. (i, j, k, l) Block size $s \times s$ of mean filtered Correlations Map with $s = 4, 8, 16, 32$.

in our experiments that a small number of levels (e.g., $l = 3$) is enough to obtain better performance that when using a single global threshold, but tends to give rise to false positives. We then tried using an increasing number of levels and found that, in our tests, the good partitioning is obtained with $l = 10$. Figure 4a shows an authentic image, while its forged counterpart is presented in 4b. Figures 4c, 4d, 4e and 4f show how the output, i.e. *BinaryImage*, changes according to the number of levels used to partition the input image.

5 Experimental Study

In this section we present the results of an experimental study we conducted to evaluate the effectiveness of our technique[1]. The study has been organized in two stages. In the first stage, we analyzed the localization performance of the technique by *per se*. In the second stage, we compared the performance of our technique to detect authentic and forged images with the one of the Lin *et al.* algorithm [11], a popular method used to detect the image forgery exploiting DQ effect.

The PNU filter used during the experiments is based on the one presented in [9] and it uses the Daubechies wavelet 16 with 8 vanishing moments. The dataset

[1] A reference implementation of our technique is available upon request.

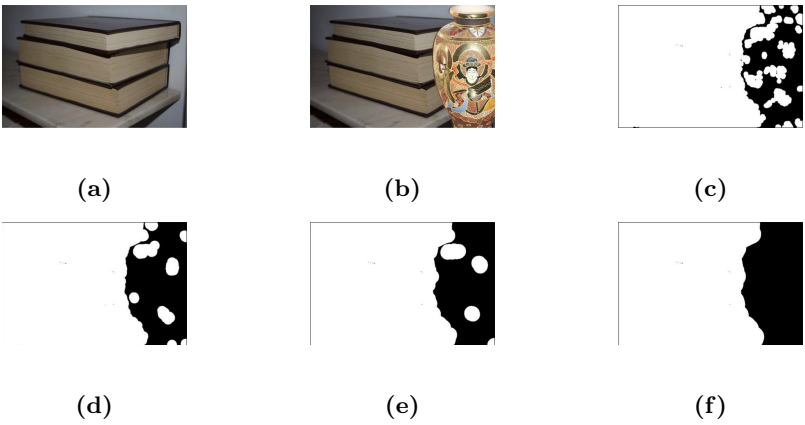

Fig. 3. (a) Authentic image. (b) Forged image with splicing. (c, d, e, f) Binary images resulting from the application of the opening operation using a disk with increasing radius ($r = 8, 16, 24, 32$).

used for our experiments is the UniSa TIDE dataset. It has been first presented in [2] and consists of 2,000 authentic JPEG images and 2,000 forged JPEG images. It was assembled from a set of 200 authentic raw images taken by using two digital cameras, Nikon D5100 and Nikon D600, and compressed in JPEG employing the following quality factors: $QF = \{100, 98, 95, 90, 85, 80, 75, 70, 65, 60\}$ (obtaining 2,000 authentic JPEG images). At this point, 200 of these images were chosen uniformly at random and were forged using splicing operations. The objects were pasted either from the same picture, from another image from the same digital camera, or from an image from a completely different camera. The resulting images were compressed again in JPEG employing the previous quality factors, obtaining 2,000 forged JPEG images. In our tests, for each of the two considered digital cameras, we used a first batch of 50 authentic images with low compression rate to calculate the reference pattern. We then used a second batch of 1,000 authentic images and 1,000 forged images for evaluating our technique in the experiments presented in this section.

5.1 Test 1: Forged Blocks Detection

In this test we investigate the ability of the proposed technique to correctly detect the forged blocks belonging to a set of reference images coming from the UniSa TIDE dataset. We report in Table 1 the outcome of the test. The overall performance of the technique seems to be very good as over the 74% of total forged blocks (True Positive Rate, TPR) and over the 87% of the total authentic blocks (True Negative Rate, TNR) were correctly classified. Conversely, the amount of authentic blocks erroneously classified as forged (False Positive Rate, FPR) and the amount of forged blocks erroneously classified as authentic (False Negative Rate, FNR) is relatively small. In sums, the *accuracy* (ACC) of the technique,

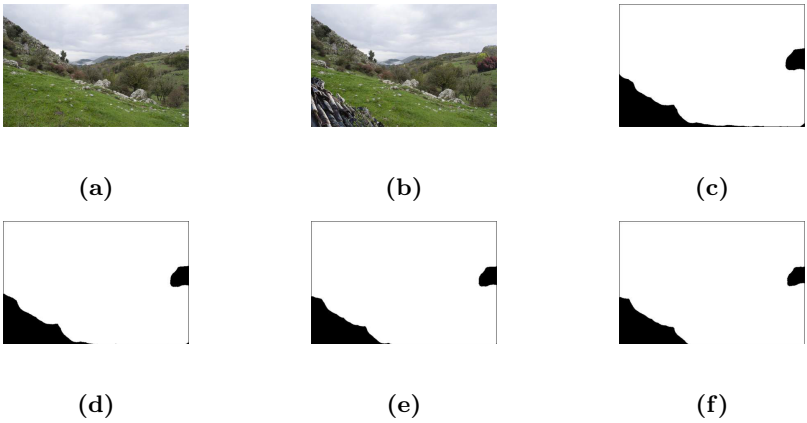

Fig. 4. (a) Authentic image. (b) Forged image using splicing. (c, d, e, f) *BinaryImages* resulting from partitioning using an increasing number of levels ($l = 5, 10, 15, 20$).

Table 1. Performance evaluation of our technique on a dataset made of 1,000 authentic images and 1,000 forged images. Performance is measured by considering the rate of blocks successfully classified as forged or not, grouped according to their quality factors. Concerning forged images, the quality factor (QF) indication refers to the one used to recompress them after the digital forgery.

Test	ACC	BAC	TPR	FPR	TNR	FNR
$Images_{\,QF \leq 75}$	0.79	0.76	0.72	0.20	0.80	0.28
$Images_{\,75 < QF \leq 90}$	0.89	0.82	0.72	0.08	0.92	0.28
$Images_{\,90 < QF \leq 100}$	0.90	0.85	0.79	0.08	0.92	0.21
$Images_{\,All}$	0.85	0.80	0.74	0.13	0.87	0.26

measured as the ratio between the number of correctly classified blocks and the total number of blocks, is good as it reaches the 85%. Since the number of authentic blocks in our dataset is, in the average, larger than the number of forged blocks, we also estimated the *balanced accuracy* (BAC) of our technique, as defined in [14]. Even in this case, the performance is good, as the balanced accuracy is approximately the 80%.

We further investigated the performance of our technique by examining how it changed according to the quality factors of the images being analyzed. The results, reported in the same table, suggest a strong correlation between the quality of the images and the ability to correctly classify authentic and forged blocks. Conversely, we analyzed the blocks that were not correctly recognized and found that some errors were due to oversaturated images or to pictures portraying outdoor environments with very complex subjects, like the foliage of trees. Many of these errors were due to blocks covering boundary regions between forged areas and authentic areas.

5.2 Test 2: Forged Images Detection

In this second test, we faced the more general problem of determining whether an input image is authentic or not. Since our technique operates at block-level and given the results presented in Section 5.1, we introduced a decision threshold so that we consider an image to be forged if more than the 10% of them blocks are classified as forged by our technique. The outcoming results have been compared with those of the Lin *et al.* algorithm [11], a popular technique for determining if an image is forged by exploiting the DQ effect (see Section 2). The two approaches have been evaluated on 2,000 testing images of the UniSa TIDE dataset (1,000 authentic and 1,000 forged) plus 2,000 other images used for training the SVM classifier used by the Lin *et al.* algorithm. The results show that our technique is able to correctly classify the 74% of the considered images, against the 55% achieved by the Lin *et al.* algorithm. If we leave out from the dataset all the images with a $QF \leq 75$, the recognition rate of our technique reaches the 83% of the images against the 59% of the Lin *et al.* algorithm.

6 Conclusion and Future Work

In this paper we proposed a non-blind passive image forgery detection technique based on the analysis of the Pixel Non-Uniformity noise, under the assumption that the camera used to take the image under investigation is available. Our proposal is based on the technique first proposed by Fridrich *et al.* in [8]. Differently other methods, such as [5,6], our technique does not require a preliminary training phase. In addition, our technique is able to operate on forged areas that are smaller than the ones used by other methods like [5–8].

We conducted a thorough experimental analysis of our technique using a reference dataset of 4,000 forged and authentic JPEG images. In the general case, our technique exhibits a very high accuracy, and easily outperforms the popular detection algorithm by Lin *et al.*. However, its performance deteriorates when working with images that are oversaturated, that have been saved using a low quality factor or that portray outdoor environments with very complex subjects. In fact, the SPN is absent in completely saturated regions and largely suppressed in dark areas. In addition, the denoising filter is less effective in removing the SPN in highly textured areas with many fine edges. Therefore, in these areas, the correlation will be naturally lower and care must be taken not to misinterpret such regions as forged. Moreover, our technique is only able to identify *splicing*, *copy-move* and *inpainting* forgery operations, while it does not support alterations of digital images like compensations (e.g., changing the color of an image), affine transforms (e.g., rotating and translating an image) and recompressions.

There are several directions worth to be investigated from here on. It could be explored the possibility to further improve the performance of our technique when locating the forged regions of an image by adopting graph-cut methods in order to better detect the boundaries of the forged regions. Similarly, it should be investigated the possibility to evaluate the results at a single-pixel level rather

than at block-level. Moreover, it would be interesting to consider in our experimentation other approaches based on SPN, and to develop a very large scale data set of images featuring forgeries of different shapes and sizes.

References

1. Cattaneo, G., Faruolo, P., Ferraro Petrillo, U.: Experiments on improving sensor pattern noise extraction for source camera identification. In: Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 609–616. IEEE (2012)
2. Cattaneo, G., Roscigno, G.: A possible pitfall in the experimental analysis of tampering detection algorithms. In: 17th International Conference on Network-Based Information Systems (NBiS), pp. 279–286 (2014)
3. Cattaneo, G., Roscigno, G., Ferraro Petrillo, U.: Experimental evaluation of an algorithm for the detection of tampered JPEG images. In: Linawati, Mahendra, M.S., Neuhold, E.J., Tjoa, A.M., You, I. (eds.) ICT-EurAsia 2014. LNCS, vol. 8407, pp. 643–652. Springer, Heidelberg (2014)
4. Cattaneo, G., Roscigno, G., Ferraro Petrillo, U.: A scalable approach to source camera identification over Hadoop. In: IEEE 28th International Conference on Advanced Information Networking and Applications (AINA), pp. 366–373. IEEE (2014)
5. Chen, M., Fridrich, J., Lukáš, J., Goljan, M.: Imaging sensor noise as digital X-ray for revealing forgeries. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 342–358. Springer, Heidelberg (2008)
6. Chen, M., Fridrich, J., Goljan, M., Lukáš, J.: Determining image origin and integrity using sensor noise. IEEE Transactions on Information Forensics and Security **3**(1), 74–90 (2008)
7. Chierchia, G., Cozzolino, D., Poggi, G., Sansone, C., Verdoliva, L.: Guided filtering for PRNU-based localization of small-size image forgeries. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2014, pp. 6231–6235. IEEE (2014)
8. Fridrich, J., Goljan, M., Lukáš, J.: Detecting digital image forgeries using sensor pattern noise. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII. **6072**, 1–11 (2006)
9. Fridrich, J., Goljan, M., Lukáš, J.: Digital camera identification from sensor pattern noise. IEEE Transactions on Information Forensics and Security **1**(2), 205–214 (2006)
10. Haralick, R.M., Sternberg, S.R., Zhuang, X.: Image analysis using mathematical morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence **9**(4), 532–550 (1987)
11. Lin, Z., He, J., Tang, X., Tang, C.K.: Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recognition **42**(11), 2492–2501 (2009)
12. Otsu, N.: A threshold selection method from gray-level histograms. Automatica **11**(285–296), 23–27 (1975)

13. Redi, J.A., Taktak, W., Dugelay, J.L.: Digital image forensics: a booklet for beginners. Multimedia Tools and Applications **51**(1), 133–162 (2011)
14. Sokolova, M.V., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, F-Score and ROC: A family of discriminant measures for performance evaluation. In: Sattar, A., Kang, B.-H. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1015–1021. Springer, Heidelberg (2006)
15. Yadav, P., Yadav, A.: Digital Image Processing. University Science Press, New Delhi (2010)