Noisesniffer: a Fully Automatic Image Forgery Detector Based on Noise Analysis

Marina Gardella, Pablo Musé, Jean-Michel Morel, Miguel Colom

Noisesniffer: a Fully Automatic Image Forgery Detector Based on Noise Analysis

Marina Gardella[1], Pablo Musé[2], Jean-Michel Morel[1], and Miguel Colom[1]
[1] Centre Borelli, ENS Paris-Saclay, CNRS, Université Paris-Saclay, France
[2] IIE, Facultad de Ingeniería, Universidad de la República, Uruguay

Abstract—Images undergo a complex processing chain from the moment light reaches the camera's sensor until the final digital image is delivered. Each of these operations leave traces on the noise model which enable forgery detection through noise analysis. In this article we define a background stochastic model which makes it possible to detect local noise anomalies characterized by their number of false alarms. The proposed method is both automatic and blind, allowing quantitative and subjectivity-free detections. Results show that the proposed method outperforms the state of the art.

Index Terms—image forensics, automatic forgery detection, noise residual, blind algorithm.

I. Introduction

Powerful image editing software is now widespread, and a fast growing number of falsified images is being shared on the web. Hence, assessing image authenticity has become crucial in many areas of science and society.

Visual inspection is not sufficient to detect tampering [1]: the average human accuracy when determining if an image is fake or not is only slightly better than random chance [2]. The erosion of trust in images combined with this human inability to detect forgeries entails a growing need for automatic detection algorithms.

In the past 15 years, digital forensics tools have been developed to address this problem [3]. These methods can be classified into passive and active. Active methods insert data or a signature when the image is digitized [4]. Hence, they are limited to specially equipped digital cameras. Passive methods do not rely on any prior or preset information. They use the fact that digital forgeries, while not leaving necessarily visual clues, do alter the image's statistics [5], [6]. Most digital images are the result of a complex processing chain. Each digital image has been notably denoised, demosaiced, re-interpolated to correct chromatic aberration and optical distortion, and subject to a color balance, a gamma-correction, and a final compression. These operations leave a dense and uniform set of digital traces enabling the detection of their local disruptions caused by a forgery.

As first suggested by [7], inconsistencies of the final noise model can reveal tampering. Each step of the camera processing pipeline modifies the model of the initial raw Poisson-Gaussian noise. Yet, the noise model remains uniform. A forgery generally alters this spatial uniformity.

This work was supported by a grant from Région Île-de-France.

Fig. 1: Illustration of the approach. Left: two input images, one pristine and the other a forgery from the UMD face-swap dataset. Middle: the distribution of flat blocks (in white or red, containing only noise). Among them, the red ones are the flattest ones in each color bin. Right: the final forgery mask detects a non uniform distribution of the red blocks among the white ones.

Therefore, adding noise to an image can be an efficient counter-forensic tool [8]. In short, local anomalies of the noise model, though imperceptible to the human eye, can become informative cues for forensic analysis.

Our plan is as follows. Section II presents a review of techniques detecting image forgeries based on noise analysis. Section III sets the principles of our method and describes the proposed approach. Section IV shows experimental results and a comparison with state of the art techniques. Section V discusses the presented main ideas and concludes the paper.

II. Related work

The main sources of noise during the acquisition process can be classified in two types: random noise sources and spatial non-uniformity sources [9]. The second class includes the photo-response non-uniformity noise (PRNU). The PRNU is caused by differences in pixel responses to uniform light sources. In [10], [11], the PRNU is estimated by averaging the noise obtained from multiple images. It is a unique stochastic fingerprint of imaging sensors that can be used for device identification and integrity verification. The limitation of this approach is that it requires access to the camera that took the image, or to a sufficient number of images taken by the same camera.

Forgery detection based on random noise analysis in raw images presents a less challenging scenario since noise can be accurately characterized by a Gaussian-Poisson distribution [12], [13]. This fact is exploited in [14], where the authors construct a noise density table and classify the contribution of different regions to the global noise model, encountering two types of contribution if the image is forged. An extension to JPEG images is available in [15].

One of the most popular algorithms for splicing detection based on noise analysis was proposed by Mahdian and Saic [16]. This method performs local wavelet-based noise level estimation using a Median Absolute Deviation estimator. Regions with homogeneous noise standard deviations are then identified. In [17], noise estimation is based on the kurtosis concentration phenomenon [18]. The fact that the kurtosis values across different band-passed filter channels is a constant value opens the way to a noise variance estimation when noise follows an additive white Gaussian noise model in the filtered domain. The final step of the method is a segmentation using k-means.

In [19], noise level estimation is performed using principal component analysis on the saturation component of the HSV colorspace. Classification of blocks in two clusters is primarily done using a k-means clustering technique, and improved by subsequently applying an SVM classifier. A similar approach can be found in [20].

The previous methods share the same drawbacks: they compute local noise levels, but neglect or ignore the noise dependence on the intensity and they compute local noise levels with fixed-size windows. The method presented in [21] addresses these drawbacks by using a multi-scale segmentation method and by estimating noise level functions instead of single noise levels.

The authors of [22] propose to estimate jointly the noise level function (NLF) and the camera response function (CRF) by segmenting the image into edge and non-edge regions. The former is used to estimate the NLF and the latter to estimate the CRF using a Bayesian approach. Noise level functions are then compared and an empirical threshold is fixed in order to detect salient curves.

In [23] the authors propose to extract local features based on noise residuals and their local co-occurrence histograms. The distribution of these histograms is assumed to follow a mixture model of two classes whose parameters are estimated using the expectation-maximization algorithm. The final output shows, for each block, the ratio of its distances to both classes. Methods based on supervised learning are more generic. However, they often lack reliability due to the unavailability of adequate training sets. In [24], a CNN-based method for noise residual extraction was proposed. Using a Siamese architecture, this algorithm is capable of enhancing the camera model artifacts, which are used as an input to the feature extraction process proposed in [23]. These last two methods are good representatives of the state of the art and will be used in the experimental section.

III. Proposed method

The method is based on four assumptions:

1) in each channel, the noise standard deviation is a function on the color level;
2) this property of the raw image is maintained by the whole image processing chain [25];
3) the blocks with small low frequency energy contain only noise;
4) the relative spatial distribution of blocks with very small energy among those with very small low frequency energy is a random uniform Poisson point process.

The principles 1 to 3 are classic in noise estimation methods [25]. Indeed, these algorithms perform roughly as follows: for each color level, they first select the flattest blocks. Then they estimate noise in high frequencies using a small percentile of these blocks, where noise is assumed to dominate over signal. In our setting, we are not interested in the numerical output of the second step but rather in identifying which regions of the image the selected blocks come from. Indeed, if we assume that the variance of the flattest blocks chosen in the first step can only be explained by noise, the small percentile of them used for noise estimation in the second step should have a random uniform repartition with respect to the first. However, if we detect that a particular area of the image concentrates too many of these blocks compared to the flat regions, we can suspect that this region has a different noise history than the rest, and is therefore a forgery.

Hence, our method first computes the distributions of the flattest blocks in each bin and the distribution of those blocks among the previous ones that exhibit the lowest standard deviation. In a second step, an a-contrario approach detects statistically significant deviations from one distribution to the other [26].

This detection will be steered by forgeries having lower noise levels than the background and having flat blocks where noise is measurable. Hence, falsifications in textured areas will not be detected by this method, and local forgeries with higher noise level than the target image should also be missed.

A. Computation of distributions

Given an image I, we first consider all its overlapping $w \times w$ blocks. Since noise is clipped in saturated pixels, we discard blocks containing at least one saturated pixel. For each channel, due to the fact that noise is intensity-dependent, blocks with size $w \times w$ are assigned to a bin according to their mean intensity. We denote by B the prefixed number of sample blocks within each bin.

For each color channel c and bin b, we construct a set V_c^b in the following way: we first apply the orthogonal discrete cosine transform (DCT II) to all the blocks in the corresponding bin and channel and compute the energy (l^2 norm) of their low frequencies. Blocks within the bin are then ordered from lowest to highest energy. A 2D

Algorithm 1 Distributions computation

Input: image I with C color channels.
Parameters: w block size, B bin size, $n << 1$, $m < 1$ two percentages.

$V = [\,]$
$L = [\,]$
blocks $\leftarrow$ list of overlapping $w \times w$ valid blocks
for each color channel $c \in [0, C-1]$ do
    compute bins of B blocks of similar mean
    for each bin b do
        compute low-frequency variance of each block
        $V_c^b \leftarrow$ list of the nB blocks having the lowest low-frequency variance
        compute standard deviation of blocks in V_c^b
        $L_c^b \leftarrow$ list of the mnB blocks having the lowest standard deviation
        if b is valid then
            $V \leftarrow [V[\dots], V_c^b]$
            $L \leftarrow [L[\dots], L_c^b]$
        end if
    end for
end for
return V, L

frequency $(i, j) \neq (0, 0)$ is said to be a low frequency if $i + j < T$ where T is a fixed threshold that depends on the block size w. For $w = 5$, 8 and 11, the corresponding T values are set to 5, 9 and 13. The mean term ($(i, j) = (0, 0)$) is discarded [27]. The elements of V_c^b are the elements contained in the n percentile for this order of block energies, where n is a small percentage value. Since most of the energy corresponding to the image geometry is located at the low and medium frequencies, this block selection ensures that we are actually considering the flattest ones inside each bin.

Given that the elements of V_c^b are the flattest blocks, their standard deviation is mainly explained by the traces of sensor noise left over by the image processing pipeline. We consider a small proportion m of them to construct the subset $L_c^b \subset V_c^b$ consisting of the $m\%$ of the blocks in V_c^b having the lowest standard deviation. If, however, the bin has a majority of completely flat blocks (i.e. blocks with zero standard deviation), the entire bin is declared invalid and discarded.

We obtain, for each channel and each valid bin, a set V_c^b consisting of a small percentile n of blocks whose variance in the low and medium frequencies is the lowest, and a subset $L_c^b \subset V_c^b$ consisting of the proportion m among them of blocks having the lowest standard deviation. Finally, we define:

$$V = \bigcup_{c} \bigcup_{\text{valid b}} V_c^b \text{ and } L = \bigcup_{c} \bigcup_{\text{valid b}} L_c^b.$$

In short, the elements of V are the flat blocks of the image in each channel bin, and the blocks in L have the lowest observed noise levels in each bin. This block selection procedure is summarized in Algorithm 1.

B. A contrario model

Our null hypothesis (H_0) is the absence of any forgery. Under H_0, blocks in V and L should have a similar spatial distribution in the image. We need to detect significant deviations from one distribution to the other that could not happen by chance. For this purpose, we now consider $W \times W$ non-overlapping blocks which we will refer to as macroblocks. For each macroblock M we observe $s_1, \dots s_N$ blocks in V, of which $s_1, \dots s_K$ are also in L. We define the random variables X_i for $i = 1, \dots, N$ as:

$$X_i = \begin{cases} 1 & \text{if } s_i \in L \\ 0 & \text{if } s_i \notin L \end{cases}$$

Under the null hypothesis, the random variables X_i are Bernoulli distributed with parameter $p = \frac{mnB}{nB} = m$, for all $i = 1, \dots N$. Following the a contrario approach [26], the number of false alarms (NFA) of the macroblock M is be defined by

$$\text{NFA}(M) = N_{\text{tests}} P_{H_0}(X \geq K) \text{ where } X = \sum_{i=1}^{N} X_i.$$

The probability $P_{H_0}(X \geq K)$ is difficult to compute directly because the random variables X_i with $i = 1, \dots, N$ are not independent. We solve this problem by considering that we are making w^2 separate tests: one for each $w \times w$ grid without overlap and assuming that for each of these tests we observe N/w^2 blocks in V and K/w^2 blocks in S. Then, the NFA of a macroblock M is defined

$$\text{NFA}(M) = N_{\text{tests}} \mathcal{B}\left(\frac{K}{w^2}, \frac{N}{w^2}, p\right).$$

This is actually an upper bound, since at least one of the grids will have more favourable parameters. Here, $\mathcal{B}$ denotes the tail of the binomial law, N_{tests} is the number of tests we are theoretically performing, namely $w^2 \times (N_x/W) \times (N_y/W)$, where (N_x, N_y) is the size of the image and therefore $(N_x/W) \times (N_y/W)$ the number of macroblocks. Algorithm 2 summarizes the NFA computation.

The NFA of a macroblock M is a (conservative) upper bound of the expected number of occurrences of the observed event, namely, observing $s_1, \dots s_N$ blocks in V, of which $s_1, \dots s_K$ are also in L. A macroblock M is ε-meaningful if $NFA(M) < \varepsilon$. Once ε is fixed, a macroblock is detected if it is ε-meaningful. For a given image, forgery detection masks are then defined as the union of all macroblocks which featured a detection.

C. Optimal parameters

The proposed method has parameters $w, b, n, m,\ W$. Optimal parameters were computed using the area under the curve (AUC) criterion for the receiver operating characteristic (ROC) curves obtained for each parameter combination on a dataset of 300 forged images from the DEFALS [28] challenge. Since false-positive rates higher than 0.1 make little sense for forgery-detection, the AUC values obtained when restricting the ROC curves to the

Algorithm 2 NFA computation

Input: image I of size $N_x \times N_y$, V, L two lists of blocks, w block size, m proportion.
Parameters: W macroblock size.

$N_{\text{tests}} \leftarrow w^2 \times (N_x/W) \times (N_y/W)$
for each macroblock M do
    $V_M \leftarrow$ number of blocks in V for macroblock M
    $S_M \leftarrow$ number of blocks in S for macroblock M
    $\text{NFA}[M] \leftarrow N_{\text{tests}}\, \mathcal{B}\left(\frac{S_M}{w^2}, \frac{V_M}{w^2}, m\right)$
end for
return NFA

$[0, 0.1]$ interval were also considered. The results showed that, regardless of the interval, the optimal parameters are $w = 5$, $B = 20000$, $n = 0.05$, $m = 0.3$, and $W = 256$.

As stated previously, to define a detection we need to set a threshold for the NFA. This threshold provides a bound to the NFA in the total number of performed tests. This choice is left to the users since it should be chosen accordingly to the level of confidence needed in the detections. Notwithstanding the foregoing, we will use in the rest of the article a threshold $\varepsilon = 1$, meaning that less than one false detection can occur in each image. As our estimate of the NFA is conservative, this 1-threshold turns out to be sufficient for a good control of the false positives. The source code of the method is available at https://github.com/marigardella/Noisesniffer.

IV. Experiments and results

A. Evaluated methods

To assess performance we compared our method with state-of-the-art algorithms based on noise analysis. Namely, we compared with Splicebuster [23], Noiseprint [24], Pan [17], and Mahdian [16]. For each algorithm, we used a publicly available implementation [29].

B. Evaluation datasets

The CG-1050 [30], the Korus [31], [32], and the UMD face-swap [33] databases were used for the comparison.

The CG-1050 database is classified in four datasets according to the type of forgery: retouching, colorization, splicing and copy-move attacks. Forgery masks were constructed by computing and thresholding the absolute difference between the original image and the forged one, at each channel. Masks were then further refined in order to prevent isolated pixels from being regarded as forged. The Korus dataset comprises object insertion and removal attacks together with accurate handmade masks.

Finally, the UMD face-swap dataset is obtained by performing face-swapping operations using two different algorithms. A bounding box of the face-swap area which serves as a mask is provided.

C. Evaluation metrics

The results obtained by the methods were evaluated using the Intersection over Union (IoU) and the $F1$ score. The classic version of these metrics are defined as:

$$IoU = \frac{TP}{TP + FN + FP},\ F1 = \frac{2TP}{2TP + FN + FP}, \quad (1)$$

where TP stands for true positive, FN for false negative and FP for false positive. These metrics are both designed to evaluate binary maps. However, all the methods used for comparison deliver continuous heatmaps. To adapt the metrics to the continuous setting, we used their weighted version. In this approach, the value of a heatmap H at each pixel x is regarded as the probability of forgery of the pixel. Therefore, we define the weighted true positives, weighted false negatives and weighted false positives as:

$$TP_w = \sum_x H(x) \cdot M(x),$$
$$FN_w = \sum_x H(x) \cdot (1 - M(x)),$$
$$FP_w = \sum_x (1 - H(x)) \cdot M(x),$$

respectively, where H is the output heatmap normalized between 0 and 1 and M is the ground-truth binary mask where pixels with value 1 are forged. Then, the weighted version of the IoU and $F1$ scores replace TP, FN and FP with their weighted versions in Eq. 1. It is important to note that, when H is a binary output, its weighted score coincides with the classic one.

Taking into account that the output of some of the evaluated methods is a two-sided heatmap, both the output heatmap and the inverted one were evaluated and only the highest score was kept for each image.

D. Results

The results obtained by each of the evaluated methods on each database are shown in Tab. I. Visual results on some selected examples are shown in Fig. 2. In the caption, we list the extremely low NFAs of the detected masks. Regardless of the evaluation metric, we observe that our method ranks first on the retouching, colorization, and splicing datasets from the CG-1050 database as well for the UMD face-swap database, whereas Splicebuster shows the best performance on the copy-move dataset from the CG-1050 database. On the Korus database, when considering the IoU score both Splicebuster and our method show the best performance. However, Splicebuster outperforms all the methods on the Korus dataset when considering the $F1$ score. This difference is explained by the fact that the IoU score penalizes false detections more than the $F1$ score. Our method, by introducing a statistical validation step, might discard true detections when they do not represent a significant deviation from the background model. But, on the other hand, this validation step controls the number of false alarms producing fewer false detections.

The best performance for all methods is achieved in the Korus dataset, which targets object insertion/removal forgery techniques. Object insertion by splicing can be detected by noise analysis whenever the background image and the spliced region have different noise models. Object

Database	CG-1050				Korus	UMD	Average
	Retouching	Colorization	Copy-move	Splicing		face-swap	ranking
Ours	0.087(1)-0.144(1)	0.077(1)-0.127(1)	0.014(2)-0.027(2)	0.035(1)-0.063(1)	0.120(1)-0.193(2)	0.091(1)-0.129(1)	1.2-1.3
Splicebuster	0.060(2)-0.108(2)	0.058(2)-0.097(2)	0.017(1)-0.031(1)	0.024(2)-0.045(2)	0.120(1)-0.200(1)	0.049(2)-0.088(2)	1.7-1.7
Noiseprint	0.031(3)-0.059(3)	0.045(5)-0.078(5)	0.014(2)-0.027(2)	0.011(3)-0.022(3)	0.094(3)-0.163(3)	0.037(3)-0.068(3)	3.2-3.2
Mahdian	0.017(5)-0.032(5)	0.055(4)-0.090(4)	0.013(5)-0.025(5)	0.010(5)-0.020(5)	0.074(4)-0.130(4)	0.024(4)-0.045(4)	4.5-4.5
Pan	0.020(4)-0.038(4)	0.058(2)-0.095(3)	0.014(2)-0.026(4)	0.011(3)-0.021(4)	0.072(5)-0.130(4)	0.021(5)-0.039(5)	3.5-4

TABLE I: Average weighted IoU - average weighted $F1$ scores obtained by each method in each of the databases considered and overall average ranking. For the proposed method, we used a NFA threshold equal to 1. Our method stands first with an average rank of 1.2 when considering the IoU score and 1.3 when considering the $F1$ score.

insertion by copy-move attacks can also disrupt the underlying noise model when performed together with counter-forensics method such as boundary blurring. However, when copy-move is done by simple copy-paste, noise-based methods are less suitable for detection. This explains the poor performance shown by all methods in the copy-move dataset of the CG-1050 database. Face-swapping can be regarded as a particular splicing technique. However, because of the particular semantic of the manipulation, it usually involves stretching the spliced face to fit the original. This leaves further traces in the noise model, which explains why noise methods perform better in this particular splicing technique rather than when considering common splicing attacks, as in the splicing dataset of the CG-1050 database. Nevertheless, face-swapping is still challenging for most of the evaluated methods, because images containing people are likely to present textured regions that can cause false detections, as shown in Fig. 2. Colorization attacks, on the other hand, require noise-based methods to take into account that noise is intensity-dependent. Our method does so by comparing blocks with similar intensities. The results show that this is the most suitable approach to detect this kind of tampering.

A visual inspection shows that methods such as Mahdian and Pan are less suited for automatic detection but might rather be considered as enhancers since their output consist of residual noise. More recent methods such as Splicebuster and Noiseprint, which use noise residue for feature extraction, provide more reliable detections. However, none of the mentioned methods is able to provide binary masks based on statistical confidence but only a heatmap. This kind of heatmaps, as given in Fig. 2, show edges and texture related to the image content rather than to noise inconsistencies. This makes the heatmaps less readable and challenges the user when interpreting them. On the contrary, our method discards many of these distractions by just keeping the regions that have a statistically significant different behaviour.

V. Discussion

In this article we proposed a new automatic forgery detection algorithm based on noise analysis. As a relevant addition to existing noise-based methods, our method incorporates a statistical validation step detecting only the inconsistencies that could not happen by chance. Each detection is associated a number of false alarms (NFA), and a threshold on the NFA provides an effective global control of the false positives: Given a set of images, an a priori threshold on the NFA can reduce the number of false alarms to an acceptable level for the user. Furthermore, the NFA associates a confidence level to each detection, which, like a p-value, can be very small for strong detections and therefore furnish a secure diagnose to users. This is a major improvement to existing algorithms in contexts where an objective statistical proof is required. Our method also provides, together with the statistically validated detection mask, a visual exploration of the relative flat patch distributions, as shown in Fig. 1. It can also aid the interpretation of the results. The performance achieved by our method in the three databases shows its relevance in the detection of different forgery techniques. The proposed approach is able to deal with different image processing pipelines, including both uncompressed and JPEG-compressed images.

Nevertheless, the method is by construction unable to detect a pure internal copy-move (because then the noise model is unaltered), and it cannot detect splicing in the (not too frequent) case where the forged region is more noisy than its target image.

References

[1] H. Farid, "Digital doctoring: How to tell the real from the fake," Significance, vol. 3, pp. 162 – 166, 11 2006.

[2] V. Schetinger, M. M. Oliveira, R. da Silva, and T. J. Carvalho, "Humans are easily fooled by digital images," Computers & Graphics, vol. 68, pp. 142–151, 2017.

[3] A. Kashyap, R. S. Parmar, M. Agrawal, and H. Gupta, "An evaluation of digital image forgery detection approaches," pp. 4747–4758, 03 2017.

[4] Z. Guojuan and L. Dianji, "An overview of digital watermarking in image forensics," 05 2011, pp. 332 – 335.

[5] W. Luo, Z. Qu, F. Pan, and J. Huang, "A survey of passive technology for digital image forensic," Frontiers of Computer Science in China, vol. 1, pp. 166–179, 05 2007.

[6] X. Lin, J. Li, S. Wang, A. Liew, F. Cheng, and X. Huang, "Recent advances in passive digital image security forensics: a brief review," Engineering, vol. 4, no. 1, pp. 29 – 39, 2018.

[7] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in Information Hiding, 2004.

[8] S. Walia and M. Kaur, "Forgery detection using noise inconsistency: A review," International Journal of Computer Science and Information Technologies, vol. 5, no. 6, pp. 7618–7622, 2014.

[9] C. Aguerrebere, J. Delon, Y. Gousseau, and P. Musé, "Study of the digital camera acquisition process and statistical modeling of the sensor raw data," 08 2013.

[10] J. Lukás, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," IEEE Trans. on Information Forensics and Security, vol. 1, pp. 205 – 214, 07 2006.

[11] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," IEEE Trans. on Information Forensics and Security, vol. 3, pp. 74 – 90, 04 2008.

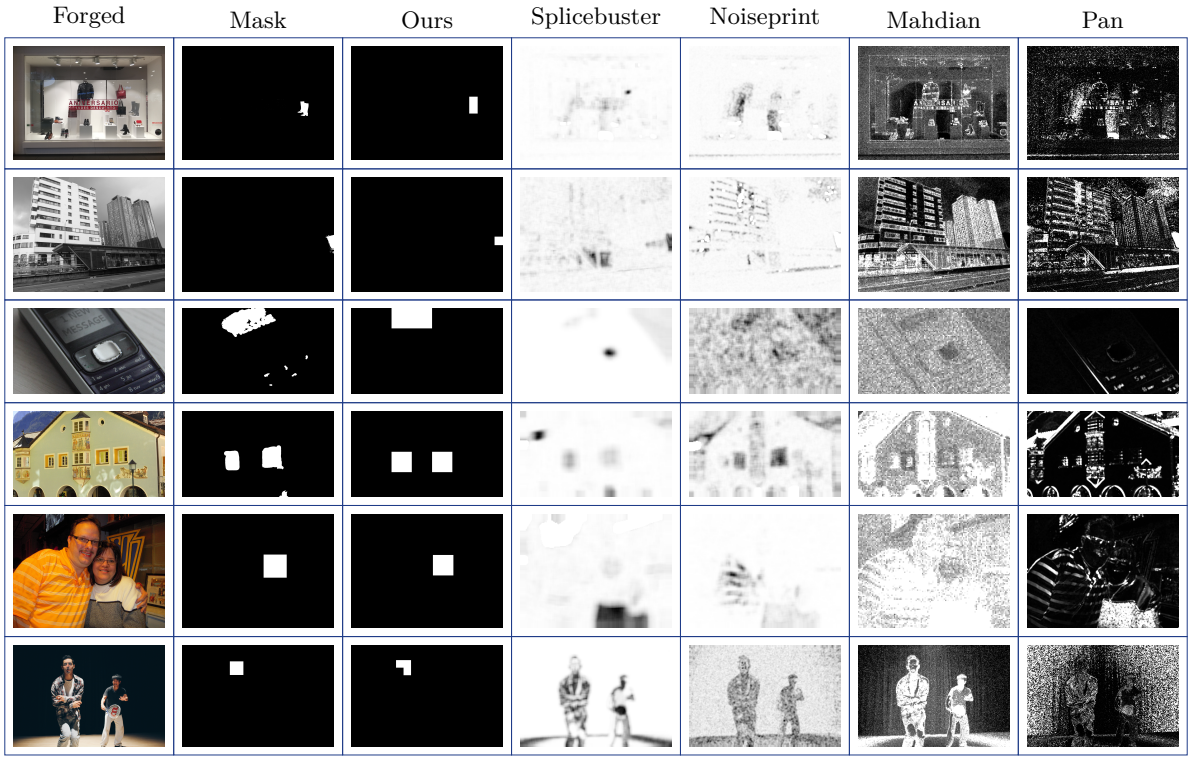

Fig. 2: Results of the proposed and reference methods on some examples. Rows 1-2 show examples from CG-1050 database, rows 3-4 show examples from Korus database and rows 5-6 from UMD face-swap database. The most significant NFA of each detection is (top to bottom): 7.4×10^{-13}, 0.17, 1.1×10^{-13}, 4.2×10^{-9}, 7.9×10^{-10} and 8.2×10^{-13}. Their probability of being false positives is overwhelmingly low.

[12] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," IEEE Trans. on image processing, vol. 17, pp. 1737–54, 11 2008.

[13] M. Colom, A. Buades, and J. Morel, "Nonparametric noise estimation method for raw images," Journal of the Optical Society of America A, vol. 31, no. 4, pp. 863–871, 2014.

[14] T. Julliand, V. Nozick, and H. Talbot, "Automatic image splicing detection based on noise density analysis in raw images," in International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, 2016, pp. 126–134.

[15] T. Julliand, V. Nozick, I. Echizen, and H. Talbot, "Using the noise density down projection to expose splicing in JPEG images," 2017.

[16] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," Image and Vision Computing, vol. 27, pp. 1497–1503, 09 2009.

[17] X. Pan, X. Zhang, and S. Lyu, "Exposing image forgery with blind noise estimation," in Proc. of the 13th ACM Multimedia Workshop on Multimedia and Security, ser. MM&Sec '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 15–20.

[18] D. Zoran and Y. Weiss, "Scale invariance and noise innature image," IEEE ICCV, Kyoto, Japan., 2009.

[19] Y. Ke, Q. Zhang, W. Min, and S. Zhang, "Detecting image forgery based on noise estimation," International Journal of Multimedia and Ubiquitous Engineering, vol. 9, no. 1, pp. 325–336, 2014.

[20] H. Zeng, Y. Zhan, X. Kang, and X. Lin, "Image splicing localization using pca-based noise level estimation," Multimedia Tools and Applications, vol. 76, no. 4, pp. 4783–4799, 2017.

[21] C.-M. Pun, B. Liu, and X.-C. Yuan, "Multi-scale noise estimation for image splicing forgery detection," Journal of Visual Communication and Image Representation, vol. 38, pp. 195 – 206, 2016.

[22] H. Yao, S. Wang, X. Zhang, C. Qin, and J. Wang, "Detecting image splicing based on noise level inconsistency," Multimedia Tools and Applications, vol. 76, no. 10, pp. 12 457–12 479, 2017.

[23] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in IEEE WIFS, 2015, pp. 1–6.

[24] D. Cozzolino and L. Verdoliva, "Noiseprint: a CNN-based camera model fingerprint," IEEE Trans. on Information Forensics and Security, vol. 15, pp. 144–159, 2020.

[25] M. Lebrun, M. Colom, A. Buades, and J. M. Morel, "Secrets of image denoising cuisine," Acta Numerica, vol. 21, p. 475–576, 2012.

[26] A. Desolneux, L. Moisan, and J.-M. Morel, From Gestalt Theory to Image Analysis: A Probabilistic Approach, 1st ed. Springer Publishing Company, Incorporated, 2007.

[27] M. Colom and A. Buades, "Analysis and Extension of the Ponomarenko et al. Method, Estimating a Noise Curve from a Single Image," IPOL, vol. 3, pp. 173–197, 2013.

[28] "Anr's DEFALS challenge," https://defals.fr/.

[29] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, "Large-scale evaluation of splicing localization algorithms for web images," Multimedia Tools and Applications, vol. 76, no. 4, pp. 4801–4834, 2017.

[30] M. Castro, D. Ballesteros, and D. Renza, "A dataset of 1050-tampered color and grayscale images (cg-1050)," Data in Brief, vol. 28, p. 104864, 2020.

[31] P. Korus and J. Huang, "Multi-scale analysis strategies in prnu-based tampering localization," IEEE Trans. on Information Forensics & Security, 2017.

[32] P. Korus and J. Huang, "Evaluation of random field models in multi-modal unsupervised tampering localization," in 2016 IEEE WIFS, 2016, pp. 1–6.

[33] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection," 2018.