Journal of *Imaging*

MDPI

Article

Forgery Detection in Digital Images by Multi-Scale Noise Estimation

Marina Gardella [1,*], Pablo Musé [2], Jean-Michel Morel [1] and Miguel Colom [1,*]

[1] Centre Borelli, ENS Paris-Saclay, Université Paris-Saclay, CNRS, 91190 Gif-sur-Yvette, France; jean-michel.morel@ens-paris-saclay.fr
[2] IIE, Facultad de Ingeniería, Universidad de la República, Montevideo 11300, Uruguay; pmuse@fing.edu.uy
* Correspondence: marina.gardella@ens-paris-saclay.fr (M.G.); miguel.colom-barco@ens-paris-saclay.fr (M.C.)

Abstract: A complex processing chain is applied from the moment a raw image is acquired until the final image is obtained. This process transforms the originally Poisson-distributed noise into a complex noise model. Noise inconsistency analysis is a rich source for forgery detection, as forged regions have likely undergone a different processing pipeline or out-camera processing. We propose a multi-scale approach, which is shown to be suitable for analyzing the highly correlated noise present in JPEG-compressed images. We estimate a noise curve for each image block, in each color channel and at each scale. We then compare each noise curve to its corresponding noise curve obtained from the whole image by counting the percentage of bins of the local noise curve that are below the global one. This procedure yields crucial detection cues since many forgeries create a local noise deficit. Our method is shown to be competitive with the state of the art. It outperforms all other methods when evaluated using the MCC score, or on forged regions large enough and for colorization attacks, regardless of the evaluation metric.

Keywords: blind estimation; forged image detection; heatmap; JPEG; noise level function

Citation: Gardella, M.; Musé, P.; Morel, J.-M.; Colom, M. Forgery Detection in Digital Images by Multi-Scale Noise Estimation. *J. Imaging* **2021**, *7*, 119. https://doi.org/10.3390/jimaging7070119

Academic Editors: Irene Amerini, Gianmarco Baldini and Francesco Leotta

Received: 31 May 2021
Accepted: 6 July 2021
Published: 17 July 2021

1. Introduction

An escalating number of falsified images are being shared on the web and feeding fake news. Indeed, the popularization of digital devices as well as the development of user-friendly manipulation software have resulted in an increase in the traffic of manipulated content. The credibility of images is under question, and therefore, methods relying on scientific evidence are required to assess the authenticity of images.

Two different approaches have emerged to address this issue. On the one hand, techniques such as digital image watermarking prevent image forgery by embedding data at the moment of digitization. Such data can be detected or extracted later to authenticate the image [1]. Although these methods provide reliable authentication, they are limited to specifically equipped cameras.

On the other hand, passive methods that do not depend on prior knowledge have also been developed. These methods rely on the fact that image forgery techniques leave specific traces that can be detected as local inconsistencies in the image statistics [2,3]. Most classic methods aim to detect specific cues such as misalignment of the Bayer pattern or perturbations in the demosaicing traces [4–6], differences in the camera response function [7,8], or inconsistencies in the JPEG-compression grid or quality [9–12].

Recent deep-learning models have been developed to tackle the task of forgery detection [13]. These methods can be trained to detect specific falsification techniques such as splicing [14,15], copy-move [16,17] and inpainting [18,19], or to detect general attacks [20–22]. The main challenge shared by these methods is the construction of adequate training datasets ensuring good results on new real-world examples.

As first suggested by [3], noise residuals can provide substantial cues for detecting forgeries. Indeed, the initial Poisson noise [23] is transformed by multiple operations

specific to each image formation process [24], leading to the final JPEG image. Hence, detecting noise inconsistencies is a rich source of forgery evidence. The use of noise residuals has evolved over time. Early methods [25,26] directly search inconsistencies in this residual whereas more recent algorithms use it as an input for further feature extraction [27,28]. Accurately estimating the residual noise traces after the complex set of transformations of the camera's processing chain is the main challenge of this class of algorithms.

With these considerations in mind, we propose a noise-based method built on non-parametric multi-scale noise estimation [29]. The multi-scale approach has been shown to effectively deal with the correlations introduced by the demosaicing and JPEG-compression processes [30] and stands out as a suitable framework for noise inconsistency analysis.

The rest of the article is organized as follows. Section 2 reviews the image forgery detection techniques based on noise inspection. The proposed method is described in Section 3. Section 4 presents experimental results in addition to a comparison with other state-of-the-art techniques. The main conclusions are summarized in Section 5, where future work directions are also highlighted.

2. Related Work

The residual noise observable in images depends on the in-camera processing pipeline. It can therefore reveal the presence of tampered regions by detecting local inconsistencies in the noise statistics that are incompatible with a unique camera processing chain. Such inconsistencies can be produced by the forgery or its post-processing.

The most outstanding source of non-uniform noise is the photo-response non-uniformity (PRNU) which is caused by small differences in the way sensors react to the light source. PRNU-based forensics methods, such as [31–33], are mostly used for source camera identification. However, since PRNU varies across the image itself, it can also provide evidence of a local manipulation. The main limitation is that PRNU-based detection methods require access to a certain number of (untampered) images taken with the same camera, to accurately estimate the PRNU pattern.

Blind noise-based detection methods usually estimate noise variance locally to detect suspicious regions and then apply a classification criterion to locate forgeries. In [25], the noise variance is estimated in blocks using a median absolute deviation (MAD) estimator in the wavelet domain. Classification is performed using homogeneous noise standard deviation criteria. In turn, Ke et al. [34] proposes noise level estimation using principal component analysis (PCA) [35]. K-means is then applied to group image blocks into two clusters. A similar approach can be found in [36]. A different method was introduced in [37], where block-wise noise estimation is based on the observation that the kurtosis values across different band-passed filter channels are constant [38]. The method concludes by segmenting the image into regions with significantly different noise variances by k-means. In [39], the image is segmented using the simple linear iterative clustering (SLIC) algorithm. Then, for each region, five filters are used to extract noise. The computed noise features are then used for classification, which is performed by energy-based graph cut.

The aforementioned methods estimate a single and constant noise level, namely an additive white Gaussian noise (AWGN) model. However, this hypothesis does not hold in realistic scenarios since noise levels depend on the image intensity [40]. More recent methods consider this fact and estimate a noise level function (NLF) rather than a single noise level. In [41], the authors proposed to jointly estimate the NLF and the camera response function (CRF) by segmenting the image into edge and non-edge regions. Noise level functions are then compared and an empirical threshold is fixed in order to detect salient curves. The methods introduced in [42,43] instead analyze a histogram based on the noise density function at the local level in order to reveal suspicious areas. The method proposed in [44] computes an NLF-based on Wiener filtering. Local noise levels in regions with a certain brightness are assumed to follow a Poisson distribution, according to which, the larger the distance to the NLF, the higher the probability of forgery. On the other hand,

the approach developed in [45] consists of estimating a noise level function that depends on the local sharpness rather than on the intensity.

Recently, forgery detection methods based on deep learning and feature modeling have been developed. The method reported in [27] proposes using noise residuals to extract local features and compute their co-occurrence histograms, which are then classified in two classes using the expectation–maximization algorithm. More recently, the same authors presented a novel CNN-based method for noise residual extraction [28]. A similar approach can be found in [46]. On the other hand, Zhou et al. [47] proposed a two-stream CNN, one for the detection of tampering artifacts and the other to leverage noise features. Deep learning-based methods are more general than previously described ones. A major limitation of these methods is that they require large training datasets, which are not always available. Furthermore, their performance generally remains dataset dependent.

3. The Proposed Method

We propose a new method for JPEG-compressed image forgery detection based on multi-scale noise estimation. The method addresses the fact that, after going through the complete camera processing pipeline, noise is not only signal-dependent but also frequency-dependent. In particular, after demosaicing, noise becomes spatially correlated, and furthermore, the quantization of the DCT coefficients during JPEG-compression differently affects the noise at each frequency. In this context, multi-scale noise estimation is the most suitable approach since it enables capturing noise at medium and low frequencies.

Let I be an image with C color channels. We first split the image into $W \times W$ blocks with $1/2$ overlap, extending the image in the borders by mirroring if necessary. We will refer to these blocks as macroblocks.

For each color channel, we estimate the global image noise curve as well as the local noise curves for each macroblock using an adaptation of the technique [29], described in Appendix A. For each channel, we compare the global noise curve with the ones locally obtained by computing the number of bins of the local noise curve that are below the global noise curve. By doing so, we obtained a heatmap for each channel that shows, for each macroblock, the percentage of bins in its noise curve whose count is below the global estimation. The information contained in the C obtained heatmaps is then combined by taking their geometric mean. As a result, we obtain a single heatmap.

For non-forged images, we expect the macroblocks to show similar noise levels functions as the one computed for the whole image. However, noise estimation is highly affected by image content. Indeed, noise overestimation is expected to happen in textured regions [48]. As a consequence, local noise curves computed over textured areas may be above the global one, even if no tampering has been performed. To prevent this kind of macroblock being perceived as suspicious, we only consider the number of bins below the global noise curve. Indeed, the global noise curve provides a lower bound for local noise curves since the noise estimation algorithm [29] has more samples from which to choose the adequate ones to estimate noise. Therefore, local noise curves that are below the global one are suspected to correspond to a different source. Figure 1 depicts the previously described situation. Indeed, we can observe that the non-forged macroblock shows higher noise levels than the global image, even though it is not tampered. On the other hand, the manipulated macroblock exhibits lower noise levels.

The next step consists of repeating the previously described process but replacing the image I and the macroblocks by their down-scaled version. To this aim, let S be the operator that tessellates the image into sets of 2×2 pixels blocks, and replaces each block by the average of the four pixels. We define $S_n(I)$as the n-th scale of an image I obtained by applying n times the operator S to the image I. This procedure allows noise curves to show the noise contained in lower frequencies and can provide further evidence of tampering that could be hidden under strong JPEG-compression.

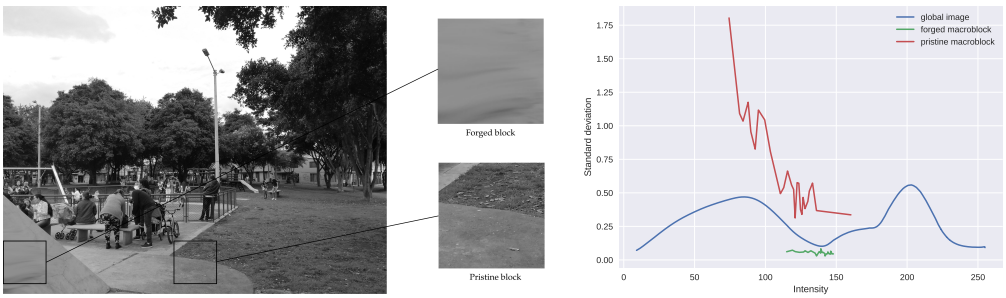

Figure 1. Estimated noise curves for the global image and for two macroblocks—one of which is contained in the manipulated region and the other is coming from the non-manipulated part of the image.

By iterating the process at successive scales, we obtain one heatmap per scale which shows the geometric mean of the percentages obtained at each channel. Each of these heatmaps may provide useful information to detect tampering since they account for noise contained at different frequencies. The sum of the heatmaps obtained at the different scales is computed and then normalized in the $[0, 255]$ interval. To obtain the final heatmap, for each pixel we compute the average of the values of each macroblock containing it.

The residual noise present in images having undergone demosaicing and JPEG-compression is correlated and therefore creates medium-sized noise spots. This may cause the blocks of size 8×8 used for noise estimation to fit inside these spots, thus causing noise underestimation. Again, estimating noise in sub-sampled versions of the image enables these spots to fit inside the scanning blocks and to accurately measure low-frequency noise. We propose repeating the sub-scaling process until reaching $S_2(I)$, as suggested in [30].

Further scales could be also considered. However, the most relevant information is already retrieved at S_2. Furthermore, the macroblock's size would become critically small and unfit to estimate noise curves: if the original macroblocks are sized $W \times W$ in S_0, in S_1 they will be of size $(W/2) \times (W/2)$, and in S_2 of size $(W/4) \times (W/4)$. Indeed, as shown in Appendix B, the best performance for the proposed method is achieved when considering macroblocks of size $W = 256$. In this context, the macroblocks are sized 128×128 in S_1 and 64×64 in S_2.

Figure 2 shows the pipeline of the proposed method, from the moment that the algorithm is fed with the input image until the final heatmap is delivered. Additionally, a summarized version of the proposed method is given below.

Given a suspect image and the parameters for the method (macroblock side, stride and number of scales), the proposed algorithm goes as follows:

1. Open the suspect image.
2. Get a list of all macroblocks according to the given macroblock size and the considered stride.
3. For each scale and each color channel, estimate the global NLF of the image and compare it to NLF computed at each macroblock. We are interested in the percentage of histogram bins below the global curve.
4. To obtain the final result of the algorithm, the heatmaps obtained at each of the scales are combined.

Please refer to Algorithm 1 for a detailed pseudo-code description. The actual source code is available at (accessed on 31 May 2021) https://github.com/marigardella/PB_Forgery_Detection, together with the instructions and requirements to run the method. Further implementation details are given in Appendix C.

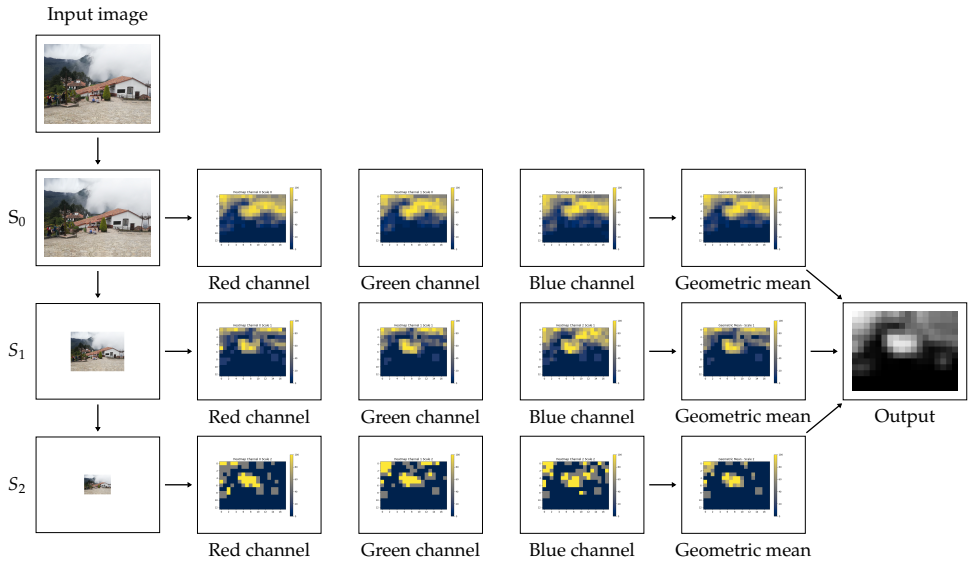

Figure 2. Complete pipeline of the method: successive scales are extracted from the input image. At each scale, one heatmap per color channel is computed and then combined according to their geometric mean. Finally, the obtained heatmaps at each scale are summed and normalized to produce the final output.

Algorithm 1 Pseudo-code for the proposed method

Input: image I of shape $N_x \times N_y$ with C color channels.
Parameters: $W = 256$ macroblock side, $S = 0.5$ stride, num_scales = 3 number of scales.

$M_x = \lfloor N_x/(W \times S) \rfloor - 1$. ▷ horizontal number of macroblocks
$M_y = \lfloor N_y/(W \times S) \rfloor - 1$. ▷ vertical number of macroblocks
macroblocks_list $\leftarrow$ list of all $W \times W$ macroblocks with S stride.
for each scale s **do**
  for each channel c **do**
    $I_s^c \leftarrow$ get image in scale s and channel$c$.
    $f_{I_s^c} \leftarrow$ noise curve estimation for I_s^c using [29] as described in A.
    $H^c \leftarrow$ zeros$(M_x \times M_y)$.
    for each macroblock in macroblocks_list **do**
      $M_s^c \leftarrow$ get macroblock in scale s and channel c.
      $f_{M_s^c} \leftarrow$ noise curve estimation for M_s^c using [29] as described in A.
      $H^c[M_s^c] \leftarrow$ percentage of bins of $f_{M_s^c}$ below $f_{I_s^c}$.
    end for
  end for
  $H_s \leftarrow$ geometric mean of the heatmaps H^c.
end for
$H_{\text{aux}} \leftarrow$ sum and normalization of heatmaps H_s.
$H \leftarrow$ compute for each pixel the average of H_{aux} for each macroblock containing it.
return H.

4. Experimental Results

We conducted two experiments. First, we evaluated the relevance of the multi-scale approach by comparing the results obtained using a single scale ($S_0(I)$), two sub-scales ($S_0(I)$ and $S_1(I)$) and three sub-scales ($S_0(I)$, $S_1(I)$ and $S_2(I)$). Second, we compared our method with state-of-the-art forgery-detection algorithms based on noise analysis.

Datasets

All experiments were conducted on the CG-1050 database [49] which contains four datasets, each one corresponding to a different forgery technique: colorization, copy-move, splicing and retouching. The total number of forged images is 1050. This database is

varied in nature, including images captured in 10 different places. The size of the images is 3456 × 4608 or 4608 × 3456 pixels. The database includes both RGB and grayscale images, all of which are JPEG-compressed. The estimated JPEG-quality [50] for each dataset is shown in Table 1.

Table 1. Average JPEG-quality and range for each of the datasets.

	Retouching	Colorization	Splicing	Copy-Move
Average JPEG-quality	86.9	86.8	87.3	86.8
JPEG-quality range	[71,88]	[71,88]	[71,88]	[71,88]

Forgery masks were constructed by computing the absolute difference between the original image and the forged one in each channel. To avoid pixels whose values had changed due to global manipulations rather than tampering, the difference from one image to another was thresholded. Only pixels whose value varied more than this threshold for at least one channel were kept. Masks were then further refined in order to prevent isolated pixels from being regarded as forged. The thresholds used were 15 for the copy-move, colorization and splicing datasets and 10 for the retouching one.

The distribution of the mask's size on each of the four datasets is shown in Figure 3.

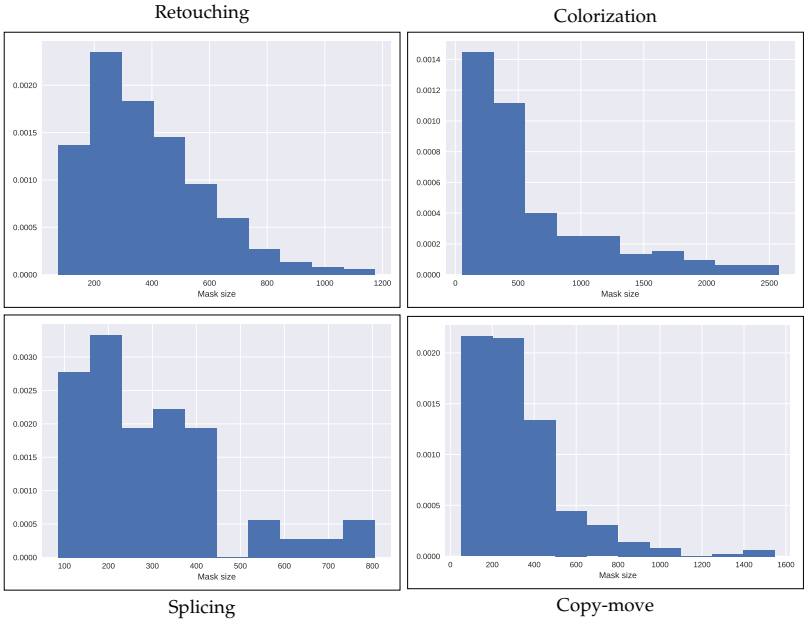

Figure 3. Distribution of the forgery size in each of the datasets considered. The forgery size is shown as the square root of the mask size, which represents the side of its equivalent square.

Evaluation Measures

Forgery localization is a particular case of binary classification. Indeed, there are two possible classes for each pixel: forged (positive) or non-forged (negative). Performance measures are usually based on the confusion matrix [51], which has four values, each one corresponding to the four possible combinations of predicted and actual classes, as shown in Figure 4.

		Predicted class: *Positive*	Predicted class: *Negative*
Actual class	*Positive*	True positive	False negative
	Negative	False positive	True negative

Figure 4. Confusion matrix: rows represent the actual classes while columns represent the prediction. The matrix has four possible values, corresponding to the four possible combinations of predicted and actual classes.

Three metrics based on these four quantities are proposed in order to compare the results obtained in both experiments. Namely, we evaluated the results using the IoU, the F1 and the MCC scores, defined as

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \times (\mathrm{TP} + \mathrm{FN}) \times (\mathrm{TN} + \mathrm{FP}) \times (\mathrm{TN} + \mathrm{FN})}},$$
$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}},$$
$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}.$$

where TP stands for true positive, TN for true negative, FN for false negative and FP for false positive.

These metrics are designed to evaluate binary-estimated masks. However, all of the methods analyzed in this paper propose continuous heatmaps rather than binary masks. 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 true negatives, weighted false negatives and weighted false positives as:

$$\mathrm{TP}_w = \sum_x H(x) \times M(x),$$
$$\mathrm{TN}_w = \sum_x (1 - H(x)) \times (1 - M(x)),$$
$$\mathrm{FN}_w = \sum_x H(x) \times (1 - M(x)),$$
$$\mathrm{FP}_w = \sum_x (1 - H(x)) \times M(x),$$

respectively, where H is the output heatmap normalized between 0 and 1, and M is the ground-truth binary mask where pixels with a value of 1 are forged. Then, the weighted version of the IoU, F1 and MCC scores are obtained replacing TP, TN, FN and FP with their weighted versions. It is important to point out that for some of the methods, the output is a two-sided heatmap (meaning that suspicious regions can appear in lighter or darker colors). Taking this into consideration, both the output heatmap and the inverted one are evaluated and only the highest score is kept.

4.1. Relevance of the Multi-Scale Approach

We first examined the pertinence of a multi-scale scheme. For this purpose, we computed the results obtained when considering one single scale $S_0(I)$ (which would correspond to the input image), using two scales $S_0(I)$ and $S_1(I)$, and using three scales $S_0(I)$, $S_1(I)$ and $S_2(I)$. The scores obtained for each of these settings are shown in Table 2.

Table 2. MCC, IoU and F1 scores for our method with one scale (PB1), two scales (PB2) and three scales (PB3).

		MCC		
	Retouching	**Colorization**	**Splicing**	**Copy-Move**
PB1	0.0672	0.0958	0.0276	**0.0380**
PB2	0.0848	0.1066	0.0310	0.0377
PB3	**0.0915**	**0.1108**	**0.0316**	0.0362
		IoU		
	Retouching	**Colorization**	**Splicing**	**Copy-Move**
PB1	0.0242	0.0721	0.0112	0.0148
PB2	0.0284	0.0756	0.0122	**0.0149**
PB3	**0.0300**	**0.0761**	**0.0123**	0.0145
		F1		
	Retouching	**Colorization**	**Splicing**	**Copy-Move**
PB1	0.0454	0.1122	0.0216	0.0281
PB2	0.0529	0.1175	0.0234	**0.0282**
PB3	**0.0557**	**0.1192**	**0.0236**	0.0276

We can observe that using multiple scales leads to better results compared to a single one. Indeed, in all four datasets, the scores obtained by PB2 and PB3 are better than those obtained by PB1 for the three metrics. Regarding the number of scales yielding a better performance, the use of three scales obtains the best scores for the retouching, colorization and splicing datasets, whereas the use of two scales achieves a better performance in the copy-move dataset. However, the results obtained for the copy-move dataset are poor for the three variants of the method, and furthermore, they have very similar scores. We conclude that the use of three scales, $S_0(I)$, $S_1(I)$ and $S_2(I)$, gives the best performance among the evaluated alternatives. In fact, given that JPEG-compression is applied in 8×8 blocks without overlap, it is at S_2 that the most accurate noise estimation is achieved since we are able to capture noise contained in lower frequencies, which is less affected by the quantization of the DCT coefficients.

4.2. Comparison with State-of-the-Art Methods

In order to assess the performance of our method, we compared the results obtained on the CG-1050 dataset with those delivered by state-of-the-art noise-based methods: Splicebuster [27], Noiseprint [28], Mahdian [25], Pan [26], Zeng [36], Zhu [45] and Median [52]. For each algorithm, we used a publicly available implementation [53]. Table 3 lists all the evaluated methods as well as their reference article and the link to the source code used for the comparison.

Table 3. State-of-the-art methods used for the comparison as well as their reference and link to source code.

Method	**Ref.**	**Source Code**
Mahdian	[25]	https://github.com/MKLab-ITI/image-forensics (accessed on 31 May 2021)
Pan	[26]	https://github.com/MKLab-ITI/image-forensics (accessed on 31 May 2021)
Zeng	[36]	https://github.com/MKLab-ITI/image-forensics (accessed on 31 May 2021)
Median	[52]	https://github.com/MKLab-ITI/image-forensics (accessed on 31 May 2021)
Splicebuster	[27]	http://www.grip.unina.it/research/83-multimedia_forensics (accessed on 31 May 2021)
Noiseprint	[28]	http://www.grip.unina.it/research/83-multimedia_forensics (accessed on 31 May 2021)
Zhu	[45]	https://github.com/marigardella/Zhu_2018 (accessed on 31 May 2021)

The obtained results are given in Table 4. We observe that Splicebuster outperforms the rest of the methods in the retouching and splicing datasets regardless of the metric.

Table 4. Results of the evaluated methods measured by the average weighted IoU, F1 and MCC scores for each dataset that maximized the score.

MCC					
	Retouching	**Colorization**	**Splicing**	**Copy-Move**	**Average Ranking**
PB3	0.0915 (2)	**0.1108** (1)	0.0316 (2)	**0.0362** (1)	1.5
Splicebuster	**0.1176** (1)	0.0535 (4)	**0.0502** (1)	0.0233 (4)	2.5
Mahdian	0.0434 (6)	0.0566 (3)	0.0247 (4)	0.0257(3)	4
Pan	0.0513 (4)	0.0681 (2)	0.0282 (3)	0.0306 (2)	2.75
Noiseprint	0.0558 (3)	0.0361 (6)	0.0182 (6)	0.0177 (6)	5.25
Median	0.0479 (5)	0.0469 (5)	0.0204 (5)	0.0195 (5)	5
Zeng	0.0180 (7)	0.0262 (7)	0.0119 (8)	0.0117 (8)	7.5
Zhu	0.0147 (8)	0.0201 (8)	0.0180 (7)	0.0123 (7)	7.5
IoU					
	Retouching	**Colorization**	**Splicing**	**Copy-Move**	**Average Ranking**
PB3	0.0300 (3)	**0.0761** (1)	0.0123 (2)	0.0145 (2)	2
Splicebuster	**0.0600** (1)	0.0577 (2)	**0.0242** (1)	**0.0166** (1)	1.25
Mahdian	0.0168 (5)	0.0548 (4)	0.0102 (5)	0.0131(5)	4.75
Pan	0.0198 (4)	0.0576 (3)	0.0109 (4)	0.0138 (4)	3.75
Noiseprint	0.0312 (2)	0.0450 (7)	0.0114 (3)	0.0142 (2)	3.5
Median	0.0163 (6)	0.0513 (5)	0.0095 (7)	0.0123(6)	6
Zeng	0.0136 (7)	0.0441 (8)	0.0084 (8)	0.0114 (8)	7.75
Zhu	0.0129 (8)	0.0453 (6)	0.0102 (5)	0.0116(7)	6.5
F1					
	Retouching	**Colorization**	**Splicing**	**Copy-Move**	**Average Ranking**
PB3	0.0557 (3)	**0.1192** (1)	0.0236 (2)	0.0276 (2)	2
Splicebuster	**0.1081** (1)	0.0965 (2)	**0.0448** (1)	**0.0314** (1)	1.25
Mahdian	0.0324 (5)	0.0902 (4)	0.0199 (6)	0.0250(5)	5
Pan	0.0380 (4)	0.0946 (3)	0.0211 (4)	0.0264 (4)	3.75
Noiseprint	0.0588 (2)	0.0778 (7)	0.0222 (3)	0.0271 (3)	3.75
Median	0.0315 (6)	0.0857 (5)	0.0185 (7)	0.0236 (6)	6
Zeng	0.0264 (7)	0.0765 (8)	0.0165 (8)	0.0220 (8)	7.75
Zhu	0.0250 (8)	0.0779 (6)	0.0200 (5)	0.0224(7)	6.5

Our method ranks first for colorization attacks for all the three metrics considered. This forgery technique shows the relevance of considering noise curves instead of single noise levels. Indeed, when changing the color in a region of the image, noise levels are not necessarily perturbed. However, those noise levels will not be consistent with the new intensity but with the original. Estimating noise curves as the proposed method does enables detecting this kind of inconsistency which only appears when considering intensity-dependent noise models.

Regarding the copy-move dataset, Splicebuster delivers the best results when considering the F1 and IoU scores. However, our approach obtains the best MCC score.

The average ranking shows that Splicebuster outperforms the rest of the methods when considering both the F1 and IoU scores, followed by our method. Nevertheless, our method achieves the best average ranking when considering the MCC score, followed by Splicebuster.

Noiseprint stands out as the third best performing method for the IoU and F1 scores. It even ranks second for retouching and copy-move attacks when considering these scores. However, it shows a poor performance for the colorization dataset. This can be explained by the fact that the camera signature is left unchanged when performing this kind of manipulation.

The Pan and Mahdian methods are middle-ranked, showing better results when considering the MCC score. Finally, Median, Zeng and Zhu show the worst performance of all the considered methods regardless of the metric considered.

All of the evaluated methods have different resolutions which may affect their performance when forgeries are too small. To analyze the effects of the size of the forgeries, we computed the average score as a function of the forgery size. Figure 5 shows the average score obtained by each method when setting different lower bounds for the forgery size in each of the datasets considered.

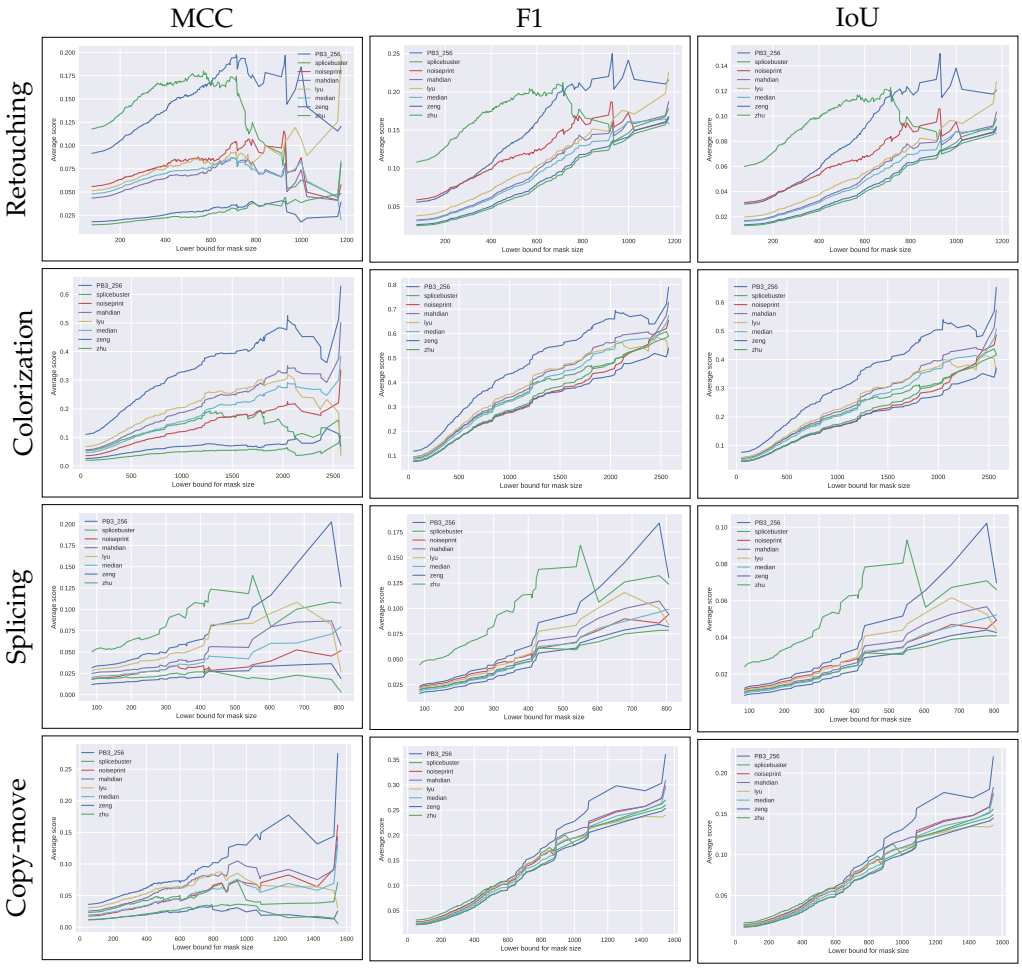

Figure 5. Average weighted MCC (**left**), IoU (**middle**) and F1 (**right**) scores obtained by each method as a function of the lower bound for the forgery size, in each of the datasets considered. Forgery size is shown as the square root of the mask size, which represents the side of its equivalent square.

The results suggest that our method outperforms the state-of-the-art approaches when considering large forgeries in all the datasets regardless of the considered score. The fact that it does not perform that well when considering small manipulations is a direct consequence of the size of the macroblocks. Indeed, for our method to provide reliable detection, the tampered region should be at least of the size of one of the tested macroblocks. In contrast, the performance of Splicebuster decreases as we consider larger forgeries. This is partially expected since the Gaussian-uniform model used in this method is better suited for small forgeries, as suggested by their authors in the original paper [27].

For further evaluation, we used the visual inspection of the results obtained by the proposed method and state-of-the-art approaches. Figure 6 shows examples of the outputs obtained by these methods for the colorization and retouching attacks, respectively, as well as for the corresponding original untampered images.

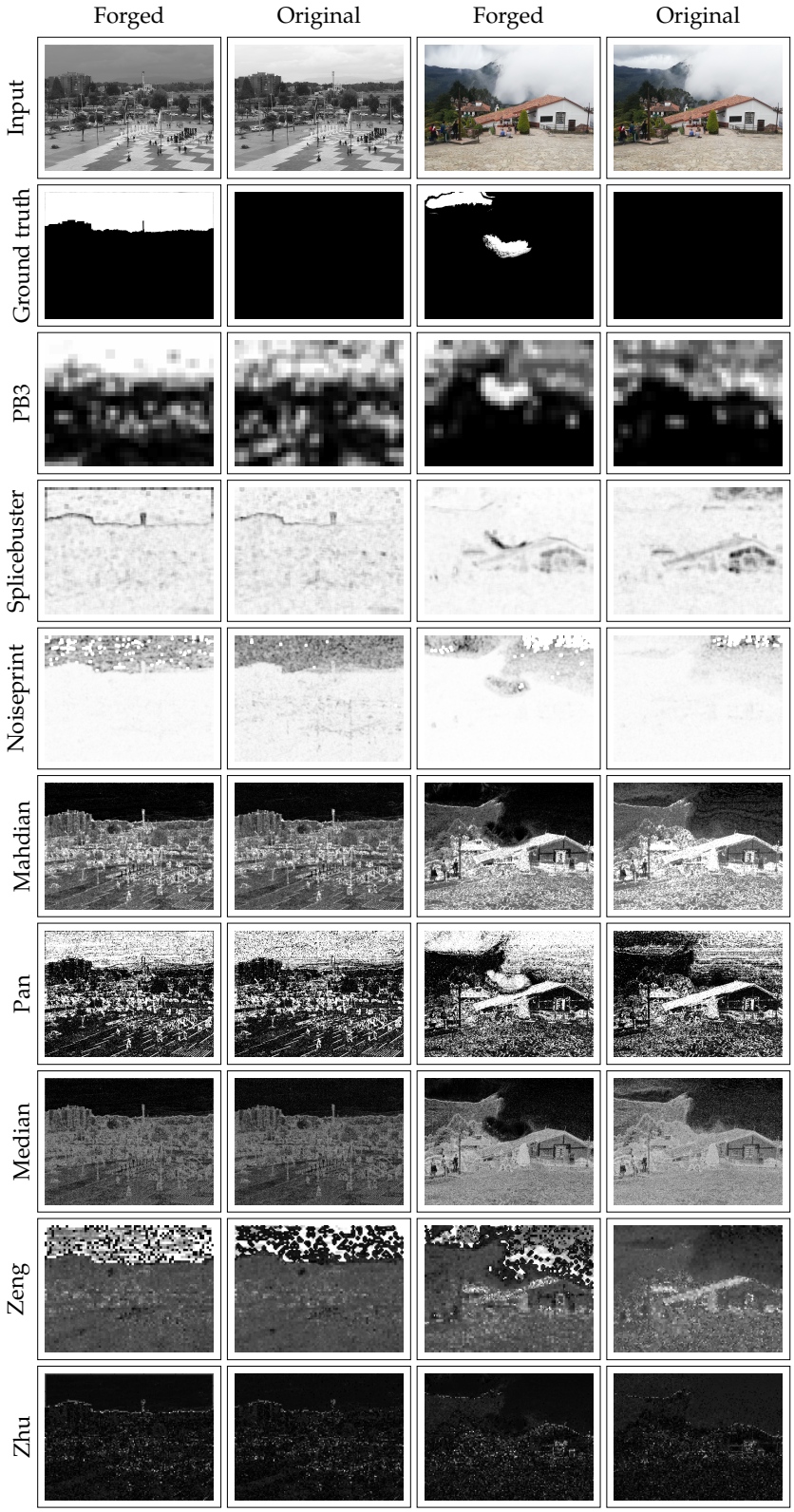

Figure 6. Results obtained for examples where colorization (**first column**) and retouching (**third column**) were performed, as well as for their corresponding original images (**second and fourth columns**). On the successive rows, the results obtained by each of the approaches for these images.

For the colorization attack shown in Figure 6, we can observe that, for all of the approaches except ours, the heatmap obtained when applying the method to the forged and original images are very similar. None of these methods is able to distinguish the

tampered region by detecting the traces of the forgery. Instead, the proposed method provides a significant difference between the forged and pristine image; we observe that the forgery clearly stands out while for the pristine image, the values of the heatmap in that area are moderated.

In the case of retouching, we observe that all of the methods point out the forged region or at least part of it as suspicious. However, several interpretation problems arise. When analyzing the results provided by Splicebuster, we can notice that the heatmap corresponding to the tampered image precisely points to the border of part of the forgery. However, when considering the pristine image, there are several areas of the heatmap showing the same values, even if they are not tampered. The Noiseprint results better localize the forgery even though false alarms are present in the pristine image. Mahdian, Pan, Median, Zeng and Zhu methods show a further drawback: in the heatmap corresponding to the manipulated image, the forged regions stand out at the same level as other non-tampered parts of the image. The interpretation of the heatmaps is left to the user who has to decide whether the regions detected as suspicious should be considered forged or discarded. On the other hand, our method is able to localize the forgery when applied to the tampered image while showing no extreme values for the pristine one, making it easier for users to interpret.

5. Conclusions, Limitations and Future Research

In the fight against disinformation, the use of objective methods able to detect manipulated multimedia content becomes crucial. Providing such tools is the aim of the digital forensics research community, and in particular, of the present work. We believe that image forgery detection is a key resource to fight fake news.

JPEG images are broadly used and clearly stand out as one of the most popular image formats. From the acquired raw image to the final JPEG format delivered by the camera, a complex processing chain is applied. Along this process, the originally Poisson-distributed noise undergoes several transformations, resulting in a complex noise structure in the JPEG image whose model does not match the AWGN hypothesis. Noise inconsistency analysis is a rich resource for forgery detection given that forged regions are likely to have undergone a different processing pipeline or an out-of-camera manipulation. However, noise-based methods require accurately dealing with the changes induced by the successive steps of the camera processing chain.

In the present paper, we proposed a method that can correctly deal with the complex noise residuals observable in the JPEG image. The proposed method implements a multi-scale approach which has shown to be suitable for analyzing the highly correlated noise present in JPEG-compressed images.

Our comparative results show that our method outperforms state-of-the-art approaches when evaluating the results with the MCC score. For colorization attacks, our method performs best, regardless of the metric. In addition, when the size of the forgeries is large enough, our method shows the best performance in all the datasets, for all three considered metrics.

Nevertheless, the proposed method has its own limitations, mainly related to too-small and too-large forgeries. Indeed, if the forgery is too small with respect to the macroblock's size, the method is likely to miss it. On the other hand, if the forgery is comparatively too large, the global noise curve may be distorted by the tampered region. The method is also by construction unable to detect a pure internal copy-move. Indeed, such a manipulation leaves the noise model unaltered. As a final negative note, the method cannot detect splicing when the forged region has more noise than the background image.

Future work includes refining the noise estimation step to use smaller macroblocks and thus improving the localization capabilities of our method.

Author Contributions: Conceptualization, M.G., P.M., J.-M.M. and M.C.; methodology, M.G., P.M., J.-M.M. and M.C.; software, M.G. and M.C.; validation, M.G.; formal analysis, M.G., P.M., J.-M.M. and M.C.; investigation, M.G.; resources, M.C.; data curation, M.G. and M.C.; writing—original draft

preparation, M.G., P.M., J.-M.M. and M.C.; writing—review and editing, M.G., P.M., J.-M.M. and M.C.; visualization, M.G.; supervision, P.M., J.-M.M. and M.C.; project administration, P.M., J.-M.M. and M.C.; funding acquisition, M.G., J.-M.M. and M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Paris Region Ph.D. grant from Région Île-de-France, the International Fact-Checking Network (IFCN) and Agence France Presse (AFP) through the Enhancing Visual Forensics (Envisu4) project, the DGA Defals challenge n° ANR-16-DEFA-0004-01, MENRT and Fondation Mathématique Jacques Hadamard.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Images from the CG-1050 database [49] were used in this article. The full database and documentation can be downloaded from https://data.mendeley.com/datasets/dk84bmnyw9/2 (accessed on 31 May 2021). The source codes for Splicebuster [27] and Noiseprint [28] methods are available at http://www.grip.unina.it/research/83-multimedia_forensics (accessed on 31 May 2021). The source codes for Pan [26], Mahdian [25], Median [52] and Zeng [36] algorithms are available at https://github.com/MKLab-ITI/image-forensics/blob/master/matlab_toolbox (accessed on 31 May 2021). The implementation of the Zhu [45] method is available at https://github.com/marigardella/Zhu_2018 (accessed on 31 May 2021). Finally, the source code for the proposed method is available at https://github.com/marigardella/PB_Forgery_Detection (accessed on 31 May 2021).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the result.

Abbreviations

The following abbreviations are used in this manuscript:

JPEG	Joint Photographic Experts Group
PRNU	Photo-Response Non-Uniformity
MAD	Median Absolute Deviation
PCA	Principal Component Analysis
SLIC	Simple Linear Iterative Clustering
AWGN	Additive White Gaussian Noise
NLF	Noise Level Function
CRF	Camera Response Function
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
MCC	Matthews' Correlation Coefficient
IoU	Intersection Over Union
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Appendix A. Adaptation of Ponomarenko's Noise Estimation Method

The proposed method is an adaptation of Ponomarenko's noise estimation method [29]. We set the default number of samples per bin to 10,000 instead of 40,000 as the original article suggests. In this way, we obtain enough bins to build the NLF of macroblocks. Additionally, the number of filtering iterations that are applied to filter the noise curve is set to 0 for the macroblocks' noise curves, while it is set to 5 for the global noise curve, as suggested by the original article. Since the NLF filtering is intended to reduce the peaks caused by textures, by doing so, the macroblocks' estimated noise curves can be regarded as a conservative upper bound of the actual noise curve.

Appendix B. Optimal Macroblock Size

The main parameter of the proposed method is W, the size of the macroblocks where local noise curves are computed. The larger this size, the more accurate the NLF estimation. However, the size of the macroblocks directly affects the precision with which forgeries are located. As shown in Figure 5, the performance of the method relies on the macroblocks' size.

In order to evaluate the capabilities of the method, we carried out an analysis of such performance depending on the size of the macroblocks. We tested three possible values for W: 512, 384 and 256. The results, presented in Table A1, suggest that the best performance is achieved for $W = 256$. Indeed, for the retouching, colorization and copy-move datasets, the best scores are obtained when considering macroblocks of size 256×256. On the other hand, when considering the splicing dataset, macroblocks of size 512×512 yield a better IoU score. However, the difference is very small and when considering other metrics, $W = 256$ achieves higher scores.

Table A1. MCC, IoU and F1 and scores for our method with one scale (PB1), two scales (PB2) and three scales (PB3) and considering different macroblock sizes: 512, 384 and 256.

			MCC	
	Retouching	**Colorization**	**Splicing**	**Copy-Move**
PB1_512	0.0585	0.0770	0.0246	0.0316
PB2_512	0.0729	0.0830	0.0268	0.0321
PB3_512	0.0804	0.0901	0.0291	0.0320
PB1_384	0.0625	0.0838	0.0242	0.0348
PB2_384	0.0789	0.0924	0.0284	0.0350
PB3_384	0.0869	0.1015	0.0289	0.0344
PB1_256	0.0672	0.0958	0.0276	**0.0380**
PB2_256	0.0848	0.1066	0.0310	0.0377
PB3_256	**0.0915**	**0.1108**	**0.0316**	0.0362
			IoU	
	Retouching	**Colorization**	**Splicing**	**Copy-Move**
PB1_512	0.0226	0.0650	0.0113	0.0141
PB2_512	0.0262	0.0673	0.0120	0.0144
PB3_512	0.0278	0.0691	**0.0124**	0.0142
PB1_384	0.0234	0.0679	0.0110	0.0145
PB2_384	0.0274	0.0708	0.0120	0.0146
PB3_384	0.0289	0.0730	0.0122	0.0144
PB1_256	0.0242	0.0721	0.0112	0.0148
PB2_256	0.0284	0.0756	0.0122	**0.0149**
PB3_256	**0.0300**	**0.0761**	0.0123	0.0145
			F1	
	Retouching	**Colorization**	**Splicing**	**Copy-Move**
PB1_512	0.0428	0.1032	0.0215	0.0268
PB2_512	0.0492	0.1067	0.0229	0.0272
PB3_512	0.0520	0.1099	0.0235	0.0270
PB1_384	0.0441	0.1068	0.0211	0.0275
PB2_384	0.0512	0.1112	0.0229	0.0277
PB3_384	0.0540	0.1151	0.0232	0.0274
PB1_256	0.0454	0.1122	0.0216	0.0281
PB2_256	0.0529	0.1175	0.0234	**0.0282**
PB3_256	**0.0557**	**0.1192**	**0.0236**	0.0276

Appendix C. Implementation Details

The main code is written in Python. The implementation of [29] used in the algorithm is written in C++. The source code for the proposed method was run in an AMD EPYC 7371 server with 16 cores (32 with hyperthreading), at 2.2 GHz clock rate and with 125 Gb of RAM. The run-time employed by the method to analyze an image of size 4608×3456 is 2 min and 22 s.

References

1. Singh, N.; Jain, M.; Sharma, S. A Survey of Digital Watermarking Techniques. *Int. J. Mod. Commun. Technol. Res.* **2013**, *1*, 6.
2. Farid, H. Digital doctoring: How to tell the real from the fake. *Significance* **2006**, *3*, 162–166. [CrossRef]
3. Popescu, A.C.; Farid, H. Statistical Tools for Digital Forensics. In *Information Hiding, Proceedings of the 6th International Workshop, IH 2004, Toronto, ON, Canada, 23–25 May 2004, Selected Papers*; Springer: Berlin, Germany, 2005; pp. 128–147.
4. Choi, C.H.; Choi, J.H.; Lee, H.K. CFA Pattern Identification of Digital Cameras Using Intermediate Value Counting. In Proceedings of the Thirteenth ACM Multimedia Workshop on Multimedia and Security, MM&Sec '11, Buffalo, NY, USA, 29–30 September 2011; Association for Computing Machinery: New York, NY, USA, 2011; pp. 21–26. [CrossRef]
5. Shin, H.J.; Jeon, J.J.; Eom, I.K. Color filter array pattern identification using variance of color difference image. *J. Electron. Imaging* **2017**, *26*, 043015. [CrossRef]
6. Bammey, Q.; Gioi, R.G.v.; Morel, J.M. An Adaptive Neural Network for Unsupervised Mosaic Consistency Analysis in Image Forensics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
7. Lin, Z.; Wang, R.; Tang, X.; Shum, H.Y. *Detecting Doctored Images Using Camera Response Normality and Consistency*; Association for Computing Machinery, Inc.: New York, NY, USA, 2005.
8. Hsu, Y.F.; Chang, S.F. Image Splicing Detection Using Camera Response Function Consistency and Automatic Segmentation. In Proceedings of the International Conference on Multimedia and Expo, Beijing, China, 2–5 July 2007.
9. Ye, S.; Sun, Q.; Chang, E.C. Detecting digital image forgeries by measuring inconsistencies of blocking artifact. In Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, Beijing, China, 2–5 July 2007; pp. 12–15.
10. Bianchi, T.; De Rosa, A.; Piva, A. Improved DCT coefficient analysis for forgery localization in JPEG images. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 2444–2447.
11. Krawetz, N.; Solutions, H.F. A picture's worth. *Hacker Factor Solut.* **2007**, *6*, 2.
12. Nikoukhah, T.; Anger, J.; Ehret, T.; Colom, M.; Morel, J.M.; Grompone von Gioi, R. JPEG grid detection based on the number of DCT zeros and its application to automatic and localized forgery detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 110–118.
13. Castillo Camacho, I.; Wang, K. A Comprehensive Review of Deep-Learning-Based Methods for Image Forensics. *J. Imaging* **2021**, *7*, 69. [CrossRef]
14. Rao, Y.; Ni, J.; Zhao, H. Deep Learning Local Descriptor for Image Splicing Detection and Localization. *IEEE Access* **2020**, *8*, 25611–25625. [CrossRef]
15. Bi, X.; Wei, Y.; Xiao, B.; Li, W. RRU-Net: The Ringed Residual U-Net for Image Splicing Forgery Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 30–39.
16. Rodriguez-Ortega, Y.; Ballesteros, D.M.; Renza, D. Copy-Move Forgery Detection (CMFD) Using Deep Learning for Image and Video Forensics. *J. Imaging* **2021**, *7*, 59. [CrossRef]
17. Liu, Y.; Guan, Q.; Zhao, X. Copy-move Forgery Detection based on Convolutional Kernel Network. *Multimed. Tools Appl.* **2018**, *77*, 18269–18293. [CrossRef]
18. Li, H.; Huang, J. Localization of Deep Inpainting Using High-Pass Fully Convolutional Network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
19. Wang, X.; Niu, S.; Wang, H. Image Inpainting Detection Based on Multi-task Deep Learning Network. *IETE Tech. Rev.* **2021**, *38*, 149–157. [CrossRef]
20. Wu, Y.; AbdAlmageed, W.; Natarajan, P. ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries with Anomalous Features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
21. Jianbo S.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905. [CrossRef]
22. Huh, M.; Liu, A.; Owens, A.; Efros, A.A. Fighting Fake News: Image Splice Detection via Learned Self-Consistency. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
23. Foi, A.; Trimeche, M.; Katkovnik, V.; Egiazarian, K. Practical Poissonian–Gaussian Noise Modeling and Fitting for Single-Image Raw-Data. *IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc.* **2008**, *17*, 1737–1754. [CrossRef]
24. Colom, M. Multiscale Noise Estimation and Removal for Digital Images. Ph.D. Thesis, Universitat de les Illes Balears, Balearic Islands, Spain, 2014.

25. Mahdian, B.; Saic, S. Using noise inconsistencies for blind image forensics *Image Vis. Comput.* **2009**, *27*, 1497–1503. [CrossRef]
26. Pan, X.; Zhang, X.; Lyu, S. Exposing Image Forgery with Blind Noise Estimation. In Proceedings of the Thirteenth ACM Multimedia Workshop on Multimedia and Security, MM&Sec '11, Buffalo, NY, USA, 29–30 September 2011; Association for Computing Machinery: New York, NY, USA, 2011; pp. 15–20. [CrossRef]
27. Cozzolino, D.; Poggi, G.; Verdoliva, L. Splicebuster: A New Blind Image Splicing Detector. In Proceedings of the 2015 IEEE International Workshop on Information Forensics and Security (WIFS), Rome, Italy, 16–19 Novenber 2015. [CrossRef]
28. Cozzolino, D.; Verdoliva, L. Noiseprint: A CNN-based camera model fingerprint. *arXiv* **2018**, arXiv:1808.08396. Available online: https://arxiv.org/abs/1808.08396 (accessed on 1 July 2021).
29. Colom, M.; Buades, A. Analysis and Extension of the Ponomarenko et al. Method, Estimating a Noise Curve from a Single Image. *Image Process. On Line* **2013**, *3*, 173–197. [CrossRef]
30. Colom, M.; Lebrun, M.; Buades, A.; Morel, J. Nonparametric Multiscale Blind Estimation of Intensity-Frequency-Dependent Noise. *IEEE Trans. Image Process.* **2015**, *24*, 3162–3175. [CrossRef] [PubMed]
31. Lukás, J.; Fridrich, J.; Goljan, M. Digital Camera Identification From Sensor Pattern Noise. *IEEE Trans. Inf. Forensics Secur.* **2006**, *1*, 205–214. [CrossRef]
32. Chen, M.; Fridrich, J.; Goljan, M.; Lukáš, J. Determining image origin and integrity using sensor noise. *IEEE Trans. Inf. Forensics Secur.* **2008**, *3*, 74–90. [CrossRef]
33. Korus, P.; Huang, J. Multi-scale Analysis Strategies in PRNU-based Tampering Localization. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 809–824 [CrossRef]
34. Ke, Y.; Zhang, Q.; Min, W.; Zhang, S. Detecting image forgery based on noise estimation. *Int. J. Multimed. Ubiquitous Eng.* **2014**, *9*, 325–336. [CrossRef]
35. Pyatykh, S.; Hesser, J.; Zheng, L. Image noise level estimation by principal component analysis. *IEEE Trans. Image Process.* **2012**, *22*, 687–699. [CrossRef]
36. Zeng, H.; Zhan, Y.; Kang, X.; Lin, X. Image splicing localization using PCA-based noise level estimation. *Multimed. Tools Appl.* **2017**, *76*, 4783–4799. [CrossRef]
37. Lyu, S.; Pan, X.; Zhang, X. Exposing Region Splicing Forgeries with Blind Local Noise Estimation. *Int. J. Comput. Vision* **2014**, *110*, 202–221. [CrossRef]
38. Zoran, D.; Weiss, Y. Scale invariance and noise innature image. In Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009.
39. Liu, B.; Pun, C.M. Splicing forgery exposure in digital image by detecting noise discrepancies. *Int. J. Comput. Commun. Eng.* **2015**, *4*, 33. [CrossRef]
40. Liu, C.; Szeliski, R.; Kang, S.B.; Zitnick, C.; Freeman, W. Automatic Estimation and Removal of Noise from a Single Image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 299–314. [CrossRef]
41. Yao, H.; Wang, S.; Zhang, X.; Qin, C.; Wang, J. Detecting image splicing based on noise level inconsistency. *Multimed. Tools Appl.* **2017**, *76*, 12457–12479. [CrossRef]
42. Julliand, T.; Nozick, V.; Talbot, H. Automatic image splicing detection based on noise density analysis in raw images. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, Lecce, Italy, 24–27 October 2016; Springer: Berlin, Germany, 2016; pp. 126–134.
43. Julliand, T.; Nozick, V.; Echizen, I.; Talbot, H. Using The Noise Density Down Projection To Expose Splicing in JPEG Images. 2017. Available online: https://hal.archives-ouvertes.fr/hal-01589761 (accessed on 1 July 2021)
44. Pun, C.M.; Liu, B.; Yuan, X. Multi-scale Noise Estimation for Image Splicing Forgery Detection. *J. Vis. Commun. Image Represent.* **2016**, *38*, 195–206. [CrossRef]
45. Zhu, N.; Li, Z. Blind image splicing detection via noise level function. *Signal Process. Image Commun.* **2018**, *68*, 181–192. [CrossRef]
46. Mayer, O.; Bayar, B.; Stamm, M.C. Learning unified deep-features for multiple forensic tasks. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, Innsbruck, Austria, 20–22 June 2018; pp. 79–84.
47. Zhou, P.; Han, X.; Morariu, V.I.; Davis, L.S. Learning rich features for image manipulation detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1053–1061.
48. Liu, C.; Freeman, W.T.; Szeliski, R.; Kang, S.B. Noise Estimation from a Single Image. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition—Volume 1, CVPR '06, New York, NY, USA, 17–22 June 2006; IEEE Computer Society: Washington, WA, USA, 2006; pp. 901–908. [CrossRef]
49. Castro, M.; Ballesteros, D.M.; Renza, D. A dataset of 1050-tampered color and grayscale images (CG-1050). *Data Brief* **2020**, *28*, 104864. [CrossRef] [PubMed]
50. Krawetz, N. A Picture's Worth . . . Digital Image Analysis and Forensics Version 2. *Hacker Factor Solut.* **2007**, *6*, 2.
51. Stehman, S.V. Selecting and interpreting measures of thematic classification accuracy. *Remote. Sens. Environ.* **1997**, *62*, 77–89. [CrossRef]
52. Wagner, J. Noise Analysis for Image Forensics. Available online: https://29a.ch/2015/08/21/noise-analysis-for-image-forensics (accessed on 30 May 2021).
53. Zampoglou, M.; Papadopoulos, S.; Kompatsiaris, Y. Large-scale evaluation of splicing localization algorithms for web images. *Multimed. Tools Appl.* **2017**, *76*, 4801–4834. [CrossRef]