Non-Semantic Evaluation of Image Forensics Tools: Methodology and Database

Quentin Bammey, Tina Nikoukhah, Marina Gardella, Rafael Grompone von Gioi, Miguel Colom, Jean-Michel Morel

Non-Semantic Evaluation of Image Forensics Tools: Methodology and Database

Quentin Bammey, Tina Nikoukhah, Marina Gardella,
Rafael Grompone von Gioi, Miguel Colom, Jean-Michel Morel
Centre Borelli – École Normale Supérieure Paris-Saclay – Université Paris-Saclay

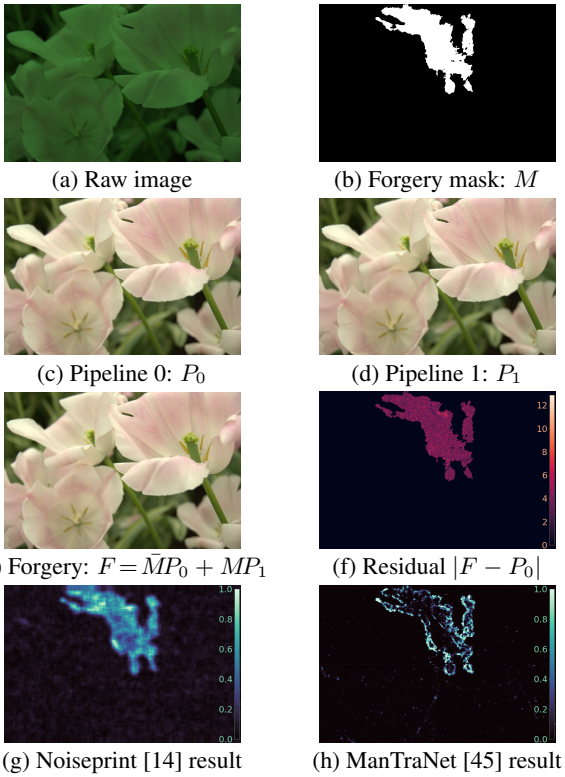
Abstract

We propose a new method to evaluate image forensics tools, that characterizes what image cues are being used by each detector. Our method enables effortless creation of an arbitrarily large dataset of carefully tampered images in which controlled detection cues are present. Starting with raw images, we alter aspects of the image formation pipeline inside a mask, while leaving the rest of the image intact. This does not change the image's interpretation; we thus call such alterations "non-semantic", as they yield no semantic inconsistencies. This method avoids the painful and often biased creation of convincing semantics. All aspects of image formation (noise, CFA, compression pattern and quality, etc.) can vary independently in both the authentic and tampered parts of the image. Alteration of a specific cue enables precise evaluation of the many forgery detectors that rely on this cue, and of the sensitivity of more generic forensic tools to each specific trace of forgery, and can be used to guide the combination of different methods. Based on this methodology, we create a database and conduct an evaluation of the main state-of-the-art image forensics tools, where we characterize the performance of each method with respect to each detection cue. Check qbammey.github.io/trace for the database and code.

1. Introduction

Digital images play an extensive role in our lives and forgeries are present everywhere [18]. Creating visually realistic image alterations is easy. Yet each modification of the image imprints traces onto it, that are cues of the tampering. Some forgery detection tools aims at detecting a specific trace in a suspicious image by finding local inconsistencies, while other methods, usually learning-based, are more generic. Semantic analysis of an image can provide hints, but the rigorous proof of a forgery should not be solely semantic. The situation is akin to the dilemma arising from the observations of Galileo, which contradicted the knowledge of his time. In the words of Bertolt Brecht [7]:

> GALILEO: How would it be if your Highness were now to observe these impossible as well as unnecessary stars through this telescope?
> THE MATHEMATICIAN: One might be tempted to reply that your telescope, showing something which cannot exist, may not be a very reliable telescope, eh?



(a) Raw image (b) Forgery mask: M
(c) Pipeline 0: P_0 (d) Pipeline 1: P_1
(e) Forgery: $F = \bar{M}P_0 + MP_1$ (f) Residual $|F - P_0|$
(g) Noiseprint [14] result (h) ManTraNet [45] result

Figure 1: Different image formation pipelines are applied to the same RAW image to obtain two images, that are combined to obtain a forged image. The authentic and forged regions present different camera pipeline traces, but are otherwise perfectly coherent. The last row shows the result of two forensic tools on this image.

Work funded by French *Ministère des Armées – Direction Générale de l'Armement* and *Région Île-de-France*.

The telescope could have been unreliable, indeed, and a scientific inquiry on the instrument could have been justified. However, concluding, as the Mathematician does, that the telescope was unreliable *just* based on the contents of the observations is not prudent. Similarly, the proof of a forgery must be based on image traces, not on semantic arguments, because the semantics of an image are usually the purpose and not the means of a forgery.

Image forensics algorithms are mainly evaluated by their performance in benchmark challenges. This practice has several limitations: in many cases, the same database is split into training and evaluation data. As a consequence, algorithms are trained and evaluated on images that have gone through similar image processing pipelines, forgery algorithms and anti-forensic tools. Hence, there is no guarantee that such learning-based methods will work in the wild, where those parameters vary much more. Regardless of the variety of the training set, the question arises of whether the forgeries are being detected by trained detectors for semantic reasons, or because of local inconsistencies in the image.

With these considerations in mind, we propose a methodology and a database to evaluate image forensic tools on images where authentic and forged regions only differ in the traces left behind by the image processing pipeline. Using this methodology, we create the *Trace database* by adding various forgery traces to raw images from the Raise [15] dataset, as shown in Fig. 1. This procedure avoids the difficulties of producing convincing and unbiased semantic forgeries, which often requires manual work. We create several datasets, each of which corresponding to a specific pipeline inconsistency, such as a different noise level or compression pattern. This gives us insight into the sensitivity of forensic tools to specific traces, and thus highlights the complementarity of different methods.

Our contribution is twofold: 1. we create a database of "fake" images with controlled inconsistencies in their formation pipeline, 2. using this database, we conduct an evaluation of existing forensic tools.

2. Related Works

There is a large literature on image forensics, starting from the seminal work of Farid [18]. Some methods focus on the detection of a specific tampering attack such as copy-move or splicing, but the most classic forgery detection methods aim at detecting local perturbations of the traces left in the image by the processing chain. Such local disruptions hint at a local forgery. To do so, these methods strive to suppress image content and highlight intrinsic artefacts left by demosaicking, JPEG encoding, etc. [39]. These forgery detection methods can therefore be grouped by their specifically-targeted traces, which we now briefly review.

Noise-level-based methods analyse the noise model of images (see Section 3) to find regions with a different amount of noise, that could result from tampering. Mahdian and Saic [34] perform local wavelet-based noise level estimation using a median absolute deviation estimator. Lyu et al. [33] relies on the kurtosis concentration phenomenon. More recently, Noisesniffer [21] defines a background stochastic model enabling the detection of local and statistically-significant noise anomalies. These methods can potentially detect a relatively wide variety of forgeries, as each can alter the noise level.

Detecting the specific image **demosaicing** algorithm (see Section 3) has not been attempted since the 2005 pioneer paper by Popescu and Farid [40], conceived at a time where those algorithms were simpler and easier to distinguish, although some generic noise-pattern analysis method can distinguish different algorithms given large enough regions [14]. However, detecting the mosaic pattern has received more extensive coverage. Choi et al. [11] used the fact that sampled pixels were more likely to take extremal values, while Shin et al. [41] noticed that they had a higher variance. Bammey et al. [4] combined the translation invariance of convolutional neural networks with the periodicity of the mosaic pattern to train a self-supervised network into implicitly detecting demosaicing artefacts. Because demosaicing artefacts lie in the high frequencies, they are lost under a strong JPEG compression or when the image has been downsampled. As such, they are usually best used on high-quality images.

JPEG compression leaves blocking effects and quantization of the DCT coefficient of each block. JPEG forensic tools can thus be divided into two categories. BAG [29] and CAGI [25] analyse blocking artefacts, while other methods analyse the DCT coefficients. More precisely, CDA [31] and I-CDA [6] are based on the AC coefficient distributions, while FDF-A [3] is based on the first digit distribution of AC coefficients. Zero [38] counts the number of null DCT coefficients in all blocks and deduces the grid origin. These methods can only work when the forgery was done after a first JPEG compression. And when this is the case, they usually yield very good results.

In the past few years, **multi-purpose** tools were proposed to detect inconsistencies from multiple traces simultaneously. Splicebuster [13] uses the co-occurences of noise residuals as local features revealing tampered image regions. Noiseprint [14] extends on Splicebuster and uses a Siamese network trained on authentic images to extract the noise residual of an image, which is then analysed for inconsistencies. ManTraNet [45] is a bipartite end-to-end network, trained to detect image-level manipulations with one part, while the second part is trained on synthetic forgery datasets to detect and localise forgeries in the image. Finally, Self-consistency [24] analysis also uses a Siamese network with the goal of detecting whether two patches

have been processed with the same pipeline. They make use of N-Cuts segmentation [26] to automatically cluster and detect relevant traces of forgeries. With these methods, exhaustiveness is theoretically possible. However, results are not self-explanatory and those method's decisions are harder to justify. Furthermore, learning-based methods can be limited by the training data, and may fail to generalize well in uncontrolled scenarios.

There is also considerable literature proposing datasets for the evaluation of forensic tools. An early example is the Columbia Dataset [37], which only contains spliced 128×128 grayscale blocks for which no masks are provided. New benchmarks were proposed in 2009 with CASIA V1.0 and V2.0 [17]. These datasets included splicing and copy-move attacks, with a total of 8000 pristine images and 6000 tampered images. Post-processing was introduced as a counter-forensics technique. MICC F220 and F2000 datasets [2] as well as the IMD dataset [12] provide further benchmarks for copy-move detection. These datasets were constructed in an automatic way. While the first two randomly select the region of the image to be copy-pasted, IMD dataset performed snippets extraction. Other datasets adressing copy-move forgeries with post-processing counter attacks are also available [42, 44].

Image forgery-detection challenges are another source of benchmark datasets. The National Institute of Standards and Technology (NIST) organizes, since 2017, an annual challenge for which different datasets are released [22]. It includes automatically and manually generated forgeries of considerable variety, and can thus be useful to evaluate image forgery detection in uncontrolled scenarios.

Some datasets aim at performing forgeries imperceptible to the naked eye. A good example is the Korus dataset [27, 28] which contains 220 pristine images and 220 handmade tampered images targeting object removal or insertion.

The recent DEFACTO dataset [35] is constructed on the MSCOCO dataset [30] and includes a wide range of forgeries such as copy-move, splicing, inpainting and morphing. Semantically meaningful forgeries are generated automatically but with several biases such as copy-pasting objects in the same axis or only performing splicing with simple objects.

Most recent forgery-detection datasets start from pristine images and perform several sorts of forgeries on them [48]. Since the creation of early datasets [17, 23, 37], the number of tampering techniques has increased to include new ones such as colorization [8], inpainting [8, 35] and morphing [35, 49]. Post-processing and counter-forensic techniques have been increasingly used to produce visually imperceptible forgeries; but such approaches may also introduce detectable traces.

Efforts have also been made to automatically obtain large datasets. Yet, the resulting forged images are either semantically incorrect [2, 12] or biased [35]. Both scenarios pose problems for training neural networks, which risk overfitting on the forgeries' methods and semantic content.

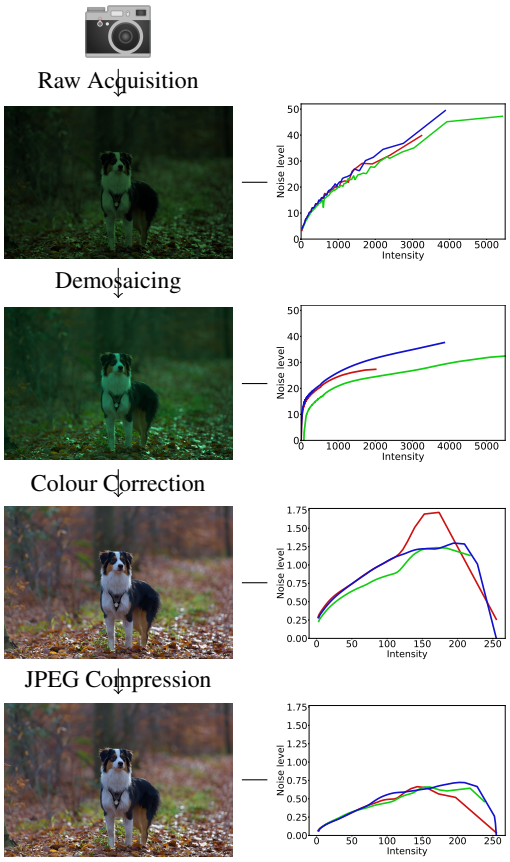


Figure 2: Evolution of the noise curves when passing through the successive steps of a (simplified) image processing pipeline.

The variety of forgery methods makes the evaluation of forensic tools difficult to interpret, as the performance depends on the suitability of the detection tool for the specific forgery method. In quantitative experiments, using multiple datasets, and especially datasets with varied forgeries, helps assess the quality of a forensic tool. However, those results also become harder to interpret. On the other hand, while results using the proposed database will not be reflective of uncontrolled scenarios, they help precisely identify which traces a forensic tool can and cannot detect.

3. Image formation pipeline

Figure 2 summarises the image processing pipeline [16] and shows how the noise curves change at its different steps.

Raw image acquisition The value at each pixel can be modelled as a Poisson random variable [20]. Noise vari-

ance at this step thus follows an affine relation $\sigma^2 = A + Bu$ where u is the intensity of the ideal noiseless image and A and B are constants (see Fig. 2). Furthermore, given the nature of the noise sources at this step, noise can be accurately modelled as uncorrelated, meaning that noise at one pixel is not related with the noise at any other pixel.

Demosaicing Most digital cameras are equipped with a single sensor array. In order to obtain a colour image, a colour filter array (CFA) is placed in front of the sensor to split incident light components according to their wavelength. The raw image obtained from the sensor therefore is a mosaic containing a single colour component per pixel: red, green, or blue. Demosaicing methods interpolate the missing colours at each pixel to reconstruct a full colour image. After demosaicing (Fig. 2), each channel has a different noise curve, and noise becomes spatially correlated.

Colour Correction In order to obtain a faithful representation of the colours as perceived by the observer, white balance adjusts colour intensities in such a way that achromatic objects from the real scene are rendered as such [32]. This is done by scaling each channel separately, thus also scaling differently the noise level of each channel. Given that the relationship between stimulus and human perception is logarithmic [19], cameras then apply a power law function to the intensity of each channel. After this step, known as gamma correction, the noise level is no longer monotonously increasing with the intensity.

JPEG compression The JPEG image standard is the most popular lossy compression scheme for photographic images [43]. The image goes through a colour space transformation and each channel is partitioned into non-overlapping 8×8-pixel blocks. The type-II discrete cosine transform (DCT) is applied to each of these blocks. The resulting coefficients are quantized according to a table and the coefficients are then compressed without additionnal loss. Due to the cancellation of high-frequency coefficients, the noise is reduced after compression.

4. The Proposed Methodology

We created a database of "forged" images which leaves the semantics of the images intact. The overall idea of our method is to take a raw image, process it with two different pipelines, and merge the two processed images as follows: the first image is used for the authentic region and the second image for the "forged" area determined by a mask, as can be seen in Fig. 1. As a base we use the RAISE-1k dataset [15], which contains one thousand pristine raw images of varied categories, taken from three different cameras. We note that the variety of source cameras is not important to our database, as we erase the previous camera traces by downsampling the image, then resimulate the whole image processing pipeline ourselves, as explained below. Furthermore, our open source generation code can be applied on any other source of images, to automatically generate arbitrarily large quantities of "forged" images.

Figure 3: Details of the same image with forgeries made using the two masks. On the left, the endomask coincides with the image's structure, here a tree. The forgery is less conspicuous than on the right where the exomask is in the sky, where the borders do not coincide with the images' content.

Methodology for the creation of the database A raw image already contains noise, furthermore its pixels are all sampled in the same CFA pattern. In order to reduce the noise and eliminate the CFA pattern, we start by downsampling each image by a factor 2. This enables us to choose the amount of noise to be added, and to mosaic the image in any of the four possible patterns. Once the image has been downsampled, we process the image with two different pipelines. The two images are then merged as explained above.

Forgery masks For each image we construct two different kinds of masks, which we shall call *endomasks* and *exomasks*. Since inconsistencies in the image processing pipeline are usually most visible at the border of the forgery, *endomasks* are obtained as regions of a segmentation of the image. To do this, we segment the original images with EncNet [47]. For each image, we take a pixel at random, and select the image region it belongs to. We accept the mask if its size is less than half the image's, otherwise we pick another pixel until we find a suitable mask. This ensures that each image has only one forgery, whose size is at most half the image's. Using such endogenous masks or *endomasks* corresponding to a region of the segmented image ensures almost invisible forgeries. Indeed their borders are natural image borders, as shown in Fig. 3.

The *exomasks* are instead unrelated to the image's content. To determine them, we start by pairing the images of the dataset according to their endomasks' sizes. Then, the

Figure 4: For each image, we use an *endomask* (left) taken from the image's segmentation, and an *exomask* (right) taken from another image and thus decorrelated from the image's contents. The last two images were paired during mask creation, thus the endomask of each becomes the exomask of the other.

endomask of each image is used as the exogenous mask, or *exomask*, of its paired image. Using a mask from another image ensures that the mask is not linked to the image's semantic. The chosen pairing enables comparisons separately on each image, as the size of the masks is similar. See Fig. 4 for examples of endo- and exomasks.

Multiple datasets One of our goals is to determine which inconsistencies each forensic tool is sensitive to. Changes in the image processing pipeline, done at different steps of the chain, lead to different inconsistencies (see Section 3). In consequence, we created five specific datasets, each of which features a specific change in the image processing pipeline. For each image, we started by randomly choosing the three parameters that are used for this image across all datasets:

- The mosaic pattern, chosen among the four possible offsets of the camera's Bayer pattern.
- The demosaicing algorithm, chosen randomly among those available in the LibRaw library [1].
- The gamma-correction power.

The gamma correction is the same for both regions of the image, and the mosaic pattern is the same except for the CFA Grid, CFA Algorithm and Hybrid datasets. For each image, both the endo- and exomasks, constructed as explained above, are the same across all datasets.

Raw Noise Level dataset In this dataset we add random noise to each raw image before processing it. As pointed out in Section 3, noise variance in raw images follows a linear relation given by $\sigma^2 = A + Bu$, where A and B are constants and u is the noiseless image. We start by randomly selecting two different pairs of constants (A_0, B_0) and (A_1, B_1), in a range that ensures the resulting images look natural. Both images are then processed with the same pipeline. This dataset mimics the inconsistencies in noise models that could be found in spliced images.

CFA Grid dataset In this dataset we only change the mosaic pattern of the forged image inside the mask. Thus, the original image and the forged one would be identical if not for their mosaic grid origins. This kind of trace may appear (with probability $\frac{3}{4}$) when the forgery was an internal copy-move.

CFA Algorithm dataset In this dataset, the two processing pipelines use different demosaicing algorithms. The demosaicing pattern is chosen independently for each pipeline. Thus there is a $\frac{1}{4}$ chance that they are aligned. A new mosaic pattern is also randomly chosen, thus having a $\frac{3}{4}$ chance of being different from the one of the main image. This dataset represents the change in the mosaic that would occur from splicing, as two different images most likely do not share the same demosaicing algorithms, and the alignment of their patterns after splicing is random.

JPEG Grid dataset In this dataset we only change the compression grid origin. Similarly to the CFA Grid dataset, if the forgery is an internal copy-move, the JPEG grid of the forged region is different from the grid in the authentic region, with probability $\frac{63}{64}$. The JPEG compression quality used in both pipelines is then chosen randomly, keeping the values in a range that is typical of most compressed images and challenging enough for JPEG-based algorithms.

JPEG Quality dataset In this dataset, both the authentic and forged regions are processed with the same pipeline, except for the JPEG compression which is done in the two regions with different quality factors, again chosen uniformly between 75 and 100. Like with the CFA Algorithm dataset or the JPEG grid data, a new JPEG grid pattern is also randomly chosen, which has a $\frac{63}{64}$ chance of being different from the main region's grid. This dataset simulates the effect of the splicing of an image onto another, both images being compressed at different quality factors.

The hybrid dataset One could argue that although generic learning-based forensics tools may not be able to

point out a single inconsistency in an image, they might be best suited to find multiple inconsistencies stacked together. Clearly, a splicing may introduce joint inconsistencies in noise level, JPEG encoding and demosaicing; while a direct copy-move can introduce alterations in the JPEG and CFA grids. To investigate such possibilities, in addition to the five specific datasets described above, we created a sixth, hybrid dataset. In this dataset, forgeries combine noise, demosaicing and/or JPEG compression traces. At least two of those traces are altered in each images.

See the supplementary materials for more details on the parameters selection.

5. Experiments

5.1. Evaluated methods

We used the constructed database to conduct an evaluation of image forensics tools. We tested both classic and SOTA forgery detection methods pertaining to different traces: noise-level-based detection methods Noisesniffer [21], Lyu [33, 46] and Mahdian [34, 46]; CFA-grid detection methods Bammey [4], Shin [41] and Choi [5, 11]; JPEG-based methods Zero [38], CAGI [25, 46], FDF-A [3, 46], I-CDA [6, 46], CDA [31, 46] and BAG [29, 46], as well as generic methods Splicebuster [13], Noiseprint [14], ManTraNet [45] and Self-Consistency [24].

5.2. Evaluation Metrics

We evaluated the results of these methods using the Matthews correlation coefficient (MCC) [36]. This metric varies from -1 for a detection that is complementary to the ground truth, to 1 for a perfect detection. A score of 0 represents an uninformative result and is the expected performance of a random classifier. The MCC is more representative than the F1 and IoU scores [9, 10], partly as it is less dependant on the proportion of positives in the ground truth, which is especially important given the large variety of forgery mask sizes in the database.

The MCC was computed for each image, and then averaged over each dataset. As most surveyed methods do not provide a binary output but a continuous heatmap, we weighted the confusion matrix using the heatmap. See the supplementary materials for more details, as well as for the score tables with the F1 and IoU metrics.

5.3. Results

The complete results are given in Table 1. Visualization of the detection by several methods on one image across all datasets can be seen in Figure 5. In the CFA and JPEG datasets, state-of-the-art methods that focus on those specific traces, such as Bammey [4] for CFA and ZERO [38] for JPEG, perform much better than generic tools. This is partly expected, as those methods aim to detect exactly this specific trace. This observation is more nuanced in the Noise Level dataset where, depending on the type of mask considered, Noisesniffer [21] and Self-Consistency [24] achieve the best results. Indeed, exomasks cover a wider range of intensities enabling a better comparison between noise models, which is exploited by Noisesniffer. Also, half of the forgeries present in this database are undetectable for this method since it is only able to detect forgeries having lower noise levels.

On the hybrid dataset, the scores of the specific methods are lower than on the specific datasets. For the JPEG-based methods, this is explained by the fact that one sixth of this dataset does not feature JPEG compression traces. For the CFA and Lyu and Mahdian noise-based methods, this is made worse by the fact that JPEG compression alters the previous noise and demosaicing artefacts, as shown in Fig. 2. In particular, CFA-based methods are notoriously weak on JPEG images, since JPEG compression removes the high frequencies, in which mosaic artefacts lie. This can be seen in Fig. 5, where the CFA-based method Bammey cannot make any prediction on the hybrid image, where the main and forged region were compressed with quality factors of 93 and 75, respectively.

While multi-purpose forensic methods can, to some extent, detect noise-level inconsistencies, in the demosaicing algorithm and in the JPEG quality, they are blind to shifts in both the JPEG and CFA grids. This is not entirely surprising; with the exception of Splicebuster, the tested generic tools are based on mostly-convolutional neural networks, which are invariant to translation. Although Noiseprint [14] adapts its training scheme to be able to detect shifts in periodic patterns, it cannot see the demosaicing and JPEG compression grids, although it is sensitive to JPEG quality inconsistencies and to some extent to demosaicing algorithm changes as well.

Most methods perform similarly on the endomask and exomask datasets. Two notable exceptions are Noisesniffer which underperforms on endomasks, and Self-Consistency, which works much better on endomasks. Both observations are easily explained: the noise model is better estimated by Noisesniffer on a flat region. The same explanation is valid for Noiseprint, which also loses performance with endomasks. In contrast, Self-consistency's content-awareness is lost when segmenting forgeries with exomasks. Regardless of the dataset considered, the scores obtained by all of the methods have a high standard deviation with respect to their mean value. This suggests that, given a dataset, the scores in each individual image are not concentrated around the mean but rather spread on a large range of values. Hence, even for methods having low scores, some good detections are likely to happen.

		Dataset					
		Noise Level	CFA Grid	CFA Algorithm	JPEG Grid	JPEG Quality	Hybrid
Noise-level-based	Noisesniffer [21]	0.128 (0.228) 0.091 (0.198)	-0.008 (0.070) -0.011 (0.073)	0.029 (0.153) 0.005 (0.111)	-0.007 (0.076) -0.009 (0.082)	0.052 (0.179) 0.020 (0.140)	0.098 (0.210) 0.061 (0.182)
	Lyu [33]	0.010 (0.090) 0.007 (0.137)	0.002 (0.093) 0.010 (0.157)	0.002 (0.094) 0.009 (0.159)	0.000 (0.089) 0.007 (0.148)	0.002 (0.091) 0.013 (0.156)	0.012 (0.097) 0.018 (0.150)
	Mahdian [34]	0.046 (0.146) 0.055 (0.171)	0.005 (0.082) 0.023 (0.159)	0.039 (0.128) 0.057 (0.183)	0.005 (0.086) 0.014 (0.146)	0.036 (0.132) 0.052 (0.180)	0.055 (0.158) 0.067 (0.191)
CFA-based	Bammey [4]	0.007 (0.084) 0.021 (0.153)	0.682 (0.329) 0.665 (0.349)	0.501 (0.427) 0.491 (0.429)	0.023 (0.095) 0.018 (0.107)	0.029 (0.091) 0.020 (0.100)	0.133 (0.288) 0.128 (0.290)
	Shin [41]	0.007 (0.101) 0.004 (0.123)	0.104 (0.166) 0.099 (0.171)	0.085 (0.172) 0.084 (0.179)	-0.002 (0.042) -0.005 (0.058)	-0.001 (0.043) -0.006 (0.059)	0.015 (0.109) 0.012 (0.114)
	Choi [5, 11]	0.026 (0.025) 0.030 (0.018)	0.603 (0.203) 0.575 (0.191)	0.420 (0.208) 0.385 (0.210)	0.001 (0.002) -0.001 (0.002)	-0.001 (0.003) 0.001 (0.001)	0.156 (0.114) 0.139 (0.116)
JPEG-based	Zero [38]	0.000 (0.000) 0.000 (0.000)	0.000 (0.000) 0.000 (0.000)	0.000 (0.000) 0.000 (0.000)	0.796 (0.349) 0.756 (0.387)	0.732 (0.413) 0.708 (0.421)	0.638 (0.451) 0.624 (0.453)
	CAGI [25]	0.004 (0.045) 0.003 (0.052)	0.000 (0.027) 0.000 (0.042)	0.002 (0.033) 0.001 (0.044)	0.038 (0.077) 0.023 (0.077)	0.044 (0.080) 0.028 (0.082)	0.031 (0.071) 0.021 (0.073)
	FDF-A [3]	0.031 (0.139) 0.014 (0.169)	-0.004 (0.087) -0.015 (0.139)	-0.003 (0.085) -0.017 (0.139)	0.226 (0.242) 0.216 (0.265)	0.228 (0.249) 0.216 (0.273)	0.203 (0.244) 0.187 (0.264)
	I-CDA [6]	0.000 (0.000) 0.000 (0.000)	0.000 (0.000) 0.000 (0.000)	0.000 (0.000) 0.000 (0.000)	0.416 (0.417) 0.423 (0.408)	0.422 (0.407) 0.414 (0.414)	0.381 (0.407) 0.385 (0.408)
	CDA [31]	-0.001 (0.034) -0.004 (0.068)	0.000 (0.055) -0.003 (0.098)	0.000 (0.052) -0.005 (0.097)	0.485 (0.339) 0.449 (0.351)	0.474 (0.344) 0.442 (0.350)	0.401 (0.360) 0.378 (0.354)
	BAG [29]	0.000 (0.015) 0.002 (0.029)	0.006 (0.078) 0.025 (0.164)	0.009 (0.079) 0.026 (0.164)	0.232 (0.461) 0.227 (0.459)	0.229 (0.458) 0.223 (0.455)	0.171 (0.430) 0.161 (0.430)
Multi-purpose tools	Noiseprint [14]	0.127 (0.200) 0.108 (0.232)	-0.001 (0.069) 0.002 (0.114)	0.066 (0.149) 0.060 (0.179)	0.013 (0.087) 0.016 (0.140)	0.178 (0.248) 0.138 (0.279)	0.153 (0.230) 0.128 (0.261)
	ManTraNet [45]	0.049 (0.091) 0.032 (0.099)	0.000 (0.040) -0.004 (0.065)	0.074 (0.169) 0.053 (0.165)	0.004 (0.023) 0.000 (0.043)	0.095 (0.164) 0.086 (0.171)	0.112 (0.169) 0.107 (0.176)
	Self- -Consistency [24]	0.082 (0.323) 0.154 (0.429)	0.028 (0.261) 0.077 (0.393)	0.036 (0.270) 0.082 (0.403)	0.011 (0.262) 0.060 (0.386)	0.078 (0.335) 0.151 (0.440)	0.138 (0.370) 0.246 (0.425)
	Splicebuster [13]	0.099 (0.188) 0.100 (0.217)	0.003 (0.085) 0.012 (0.157)	0.075 (0.157) 0.072 (0.202)	0.005 (0.083) 0.006 (0.135)	0.084 (0.175) 0.082 (0.220)	0.101 (0.192) 0.099 (0.215)

Table 1: Results of different state-of-the-art forensics tools on our six datasets, using the Matthews Correlation Coefficient (MCC), detailed in Sec. 5.2. The methods, on the left, are grouped by categories. As a baseline, a random classifier is expected to yield a score of 0. The mean of the MCC scores over each image of the dataset, as well as the standard deviation in parentheses, are shown for the exogenous mask and endogenous mask datasets. Grayed-out numbers represent results of methods on datasets that are irrelevant to said methods. The best two scores are underlined for each database.

6. Discussion

Most methods yield similar results on exo- and endo-masks. While one kind is usually sufficient, comparing the results on both shows some methods are content-aware.

The goal of this evaluation was not to rank different methods, but to offer a rigorous insight on the capabilities of each. Knowing to which kind of inconsistencies forensic tools are sensitive helps understand and explain its detec-

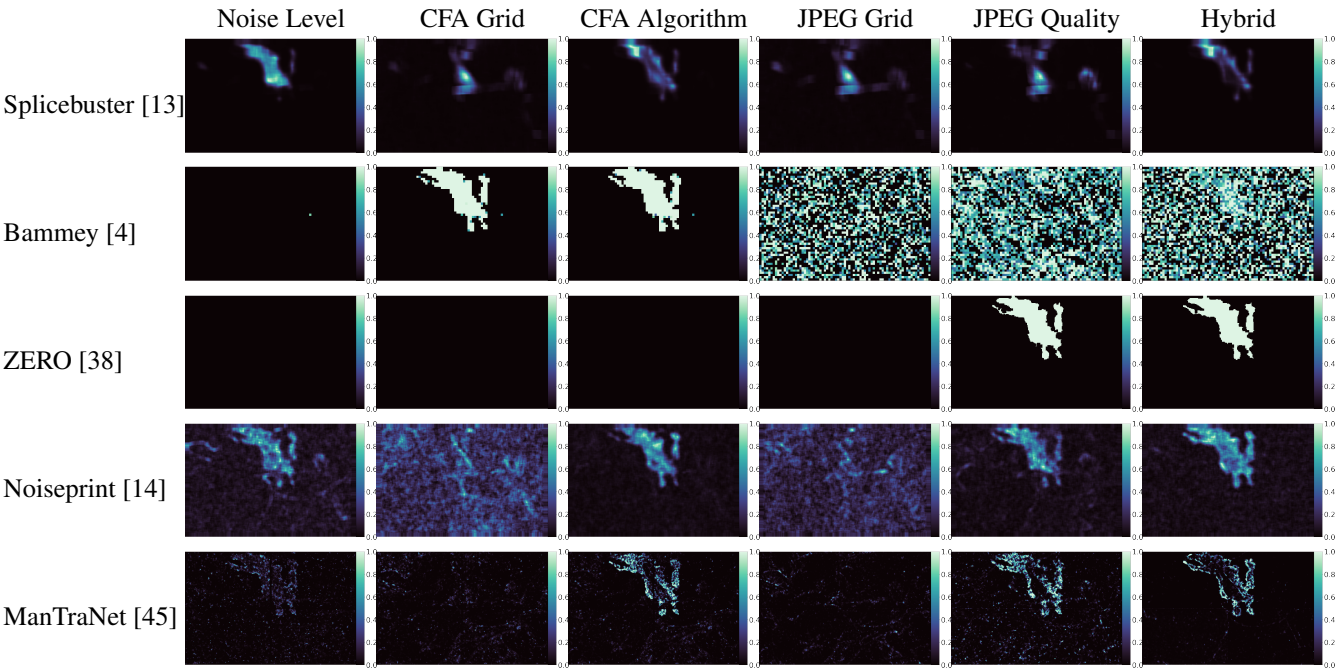


Figure 5: Visualization of the results of several methods for one image on all the datasets. Some methods, such as Noiseprint or Bammey, correctly detect the forgeries in the relevant images, but tend to make noise-like false detections in the images for which they cannot see the forgery. Automatically selecting the relevant detections of an algorithm would make it easier to use without needing interpretation. The image and mask can be seen in Fig. 1.

tions in uncontrolled cases, and can help efforts to combine different methods. In that sense, the proposed database is complementary to more traditional databases.

Even though many methods can yield decent scores, the standard deviations of theses scores over all images of the same dataset is often very high. Even though algorithms perform well on many forgeries, they also often yield false positives that require interpretation to be distinguished from true detections, such as Fig. 1. This phenomenon is further evidenced in the supplementary material. This is a critical point for many methods, as it makes them usable only to a trained eye.

7. Conclusion

Image forensics datasets are usually grouped according to forgery types (eg. splicing, inpainting, or copy-moves), and do not separate the semantic content from the actual traces left by the forgery. In this paper, we proposed to remove the semantic value of forgeries and to focus only on the traces. We designed a methodology to automatically create image "forgeries" that leave no semantic traces, by introducing controlled changes in the image processing pipeline. We built datasets by focusing on noise-level inconsistencies, mosaic and JPEG artefacts, and conducted an evaluation of some image forensics tools using this dataset.

Although we focused on three kinds of changes in the forgeries, the same methodology can be applied to more traces, including PRNU inconsistencies, multiple compression, or image manipulations such as resampling. In fact, we can address all forgeries where two different camera pipelines are involved. This includes copy-move, splicing and some methods of inpainting. Further work would incorporate other traces, such as those left by synthesis methods. Although not surveyed here, the same methodology can be applied to study robustness of detection under adverse events such as global JPEG compression, by passing the images through compression before analysis. Our images were not post-processed, except for inconsistencies linked to JPEG compression. This makes it easy to assess the robustness to any kind of post-processing.

Note that there are no authentic images in the dataset. Testing the frequency of false positives is for now complementary to the proposed methodology, but could be included in further work by comparing the response of forensic tools to the forged images and their authentic counterparts, otherwise-processed with the same pipeline.

Our method can transform automatically large sets of images into forged images with fully controlled tampering cues and no bias that might cause overfitting. Besides evaluation of existing image forensics tools, this methodology could also be used to train forgery detection methods, although care would be needed so as not to overfit if using the same methodology for both training and evaluation.

References

[1] Libraw library, copyright © 2008-2019 libraw llc, https://www.libraw.org.

[2] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Trans. on Information Forensics and Security*, 6(3):1099–1110, Sep. 2011.

[3] Irene Amerini, Rudy Becarelli, Roberto Caldelli, and Andrea Del Mastio. Splicing forgeries localization through the use of first digit features. In *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 143–148. IEEE, 2014.

[4] Quentin Bammey, Rafael Grompone von Gioi, and Jean-Michel Morel. 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)*, June 2020.

[5] Quentin Bammey, Rafael Grompone von Gioi, and Jean-Michel Morel. Image Forgeries Detection through Mosaic Analysis: the Intermediate Values Algorithm. *Image Processing On Line*, 11:317–343, 2021. https://doi.org/10.5201/ipol.2021.355.

[6] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved dct coefficient analysis for forgery localization in jpeg images. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2444–2447. IEEE, 2011.

[7] Bertolt Brecht. *The Life of Galileo*. Methuen, 1968. Translated by D. I. Vesey.

[8] Maikol Castro, Dora M. Ballesteros, and Diego Renza. A dataset of 1050-tampered color and grayscale images (cg-1050). *Data in Brief*, 28:104864, 2020.

[9] Davide Chicco. Ten quick tips for machine learning in computational biology. *BioData mining*, 10:35–35, Dec 2017.

[10] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):6–6, Jan 2020.

[11] Chang-Hee Choi, Jung-Ho Choi, and Heung-Kyu Lee. 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, page 21–26, New York, NY, USA, 2011. Association for Computing Machinery.

[12] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou. An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on Information Forensics and Security*, 7(6):1841–1854, 2012.

[13] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Splicebuster: A new blind image splicing detector. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 11 2015.

[14] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15:144–159, 2020.

[15] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 219–224, 2015.

[16] Mauricio Delbracio, Damien Kelly, Michael S. Brown, and Peyman Milanfar. Mobile computational photography: A tour, 2021.

[17] J. Dong, W. Wang, and T. Tan. Casia image tampering detection evaluation database. In *2013 IEEE China Summit and International Conference on Signal and Information Processing*, pages 422–426, 2013.

[18] Hany Farid. *Photo Forensics*. The MIT Press, 2016.

[19] GT Fechner. Elemente der psychophysik, breitkopf und härtel. *Leipzig: Breitkopf und Härtel*, 1860.

[20] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 17:1737–54, 11 2008.

[21] Marina Gardella, Pablo Musé, Jean-Michel Morel, and Miguel Colom. Noisesniffer: a fully automatic image forgery detector based on noise analysis. In *2021 IEEE International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2021.

[22] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72, 2019.

[23] Y.-F. Hsu and S.-F. Chang. Detecting image splicing using geometry invariants and camera characteristics consistency. In *International Conference on Multimedia and Expo*, 2006.

[24] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *ECCV*, 2018.

[25] Chryssanthi Iakovidou, Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Content-aware detection of jpeg grid inconsistencies for intuitive image forensics. *Journal of Visual Communication and Image Representation*, 54:155–170, 2018.

[26] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[27] P. Korus and J. Huang. Evaluation of random field models in multi-modal unsupervised tampering localization. In *Proc. of IEEE Int. Workshop on Inf. Forensics and Security*, 2016.

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

[29] Weihai Li, Yuan Yuan, and Nenghai Yu. Passive detection of doctored jpeg image via block artifact grid extraction. *Signal Processing*, 89(9):1821–1829, 2009.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[31] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492–2501, 2009.

[32] Olivier Losson and Eric Dinet. From the Sensor to Color Images. In Christine Fernandez-Maloigne, Frédérique Robert-Inacio, and Ludovic Macaire, editors, *Digital Color - Acquisition, Perception, Coding and Rendering*, Digital Image and Signal Processing series, pages 149–185. Wiley, Mar. 2012.

[33] Siwei Lyu, Xunyu Pan, and Xing Zhang. Exposing region splicing forgeries with blind local noise estimation. *International Journal of Computer Vision*, 110:202–221, 11 2013.

[34] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27:1497–1503, 09 2009.

[35] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J-L. Dugelay, and M. Pic. Defacto: Image and face manipulation dataset. In *27th European Signal Processing Conference (EUSIPCO 2019)*, A Coruña, Spain, Sept. 2019.

[36] B.W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, 1975.

[37] Tian-Tsong Ng and Shih-Fu Chang. A data set of authentic and spliced image blocks. Technical report, Columbia University, June 2004.

[38] Tina Nikoukhah, Jérémy Anger, Thibaud Ehret, Miguel Colom, Jean-Michel Morel, and Rafael Grompone von Gioi. 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*, pages 110–118, 2019.

[39] Alin C. Popescu and Hany Farid. Statistical tools for digital forensics. In *Information Hiding*, 2004.

[40] A. C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.

[41] Hyun Jun Shin, Jong Ju Jeon, and Il Kyu Eom. Color filter array pattern identification using variance of color difference image. *Journal of Electronic Imaging*, 26(4):043015, 2017.

[42] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic. Comofod — new database for copy-move forgery detection. In *Proceedings ELMAR-2013*, pages 49–54, 2013.

[43] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[44] B. Wen, Y. Zhu, R. Subramanian, T. Ng, X. Shen, and S. Winkler. Coverage — a novel database for copy-move forgery detection. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 161–165, 2016.

[45] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. 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)*, June 2019.

[46] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Large-scale evaluation of splicing localization algorithms for web images. *Multimedia Tools and Applications*, 76(4):4801–4834, 2017.

[47] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[48] Lilei Zheng, Ying Zhang, and Vrizlynn Thing. A survey on image tampering and its detection in real-world photos. *Journal of Visual Communication and Image Representation*, 12 2018.

[49] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Two-stream neural networks for tampered face detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1831–1839, 2017.