



HAL
open science

A Contrario Mosaic Analysis for Image Forensics

Quentin Bammev

► **To cite this version:**

Quentin Bammev. A Contrario Mosaic Analysis for Image Forensics. Advanced Concepts for Intelligent Vision Systems (ACIVS), Springer, Aug 2023, Kumamoto, Japan. 10.1007/978-3-031-45382-3_19 . hal-04185508

HAL Id: hal-04185508

<https://hal.science/hal-04185508v1>

Submitted on 22 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

A Contrario Mosaic Analysis for Image Forensics

Quentin Bammey¹[0000–0003–2280–2349]

Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli

Abstract. With the advent of recent technologies, image editing has become accessible even without expertise. However, this ease of manipulation has given rise to malicious manipulation of images, resulting in the creation and dissemination of visually-realistic fake images to spread disinformation online, wrongfully incriminate someone, or commit fraud. The detection of such forgeries is paramount in exposing those deceitful acts. One promising approach involves unveiling the underlying mosaic of an image, which indicates in which colour each pixel was originally sampled prior to demosaicing. As image manipulation will alter the mosaic as well, exposing the mosaic enables the detection and localization of forgeries. The recent introduction of positional learning has facilitated the identification of the image mosaic. Nevertheless, the clues leading to the mosaic are subtle and frail against common operation such as JPEG compression. The pixelwise estimation of the mosaic is thus often imprecise, and a comprehensive analysis and aggregation of the results are necessary to effectively detect and localize forged areas. In this work, we propose MIMIC: Mosaic Integrity Monitoring for Image *a Contrario* forensics. an *a contrario* method to analyse a pixelwise mosaic estimation. We show that despite the weakness of these traces, the sole analysis of mosaic consistency is enough to beat the state of the art in forgery detection and localization on uncompressed images. Moreover, results are promising even on slightly-compressed images. The *a contrario* framework ensures robustness against false positives, and the complementary nature of mosaic consistency analysis to other forensic tools makes our method highly relevant for detecting forgeries in high-quality images.

Keywords: Positional learning · Image Forgery Detection · Demosaicing · *a contrario* · Media forensics

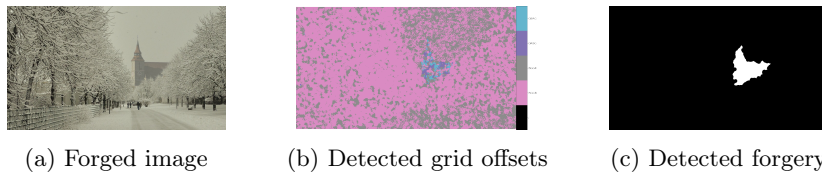


Fig. 1: MIMIC automatically detects forgeries based on the analysis of the underlying mosaic of an image. An *a contrario* detection automatically filters the grid estimates in search for significant inconsistencies, while keeping false positives under control.

1 Introduction

The once-reliable status of photographic images as evidence is now uncertain, owing to the proliferation of digital photography and the development of sophisticated photo editing tools. Although image modifications are frequently intended to enhance an image’s aesthetic appeal, they can alter the meaning of the image. The addition, modification, or concealment of objects can give an image a completely new and potentially misleading meaning, especially as the modifications can now appear convincingly authentic.

However, images contain traces and artefacts left by the various operations of the image signal processing pipeline (ISP), from the camera sensors to the compressed version of the image. Those traces act as a signature to the image; as modifications made to an image will alter the original traces. As such, the resulting inconsistencies can be detected to show that the image has been forged. One such trace that can be analysed is the image mosaic.

Most cameras do not capture colours directly, instead, a colour filter array (CFA) is used to sample each pixel’s value in a single colour. By applying filters of different colours to adjacent sensors, the pixels are sampled in different colours. The missing colours are interpolated with a demosaicing algorithm to provide a true colour image. We focus on the Bayer CFA, used in nearly all commercial cameras. This matrix samples half of the pixels in green, a quarter in red, and a quarter in blue, in a quincunx pattern. Depending on the offset, the image can be sampled in one of four patterns: $\begin{smallmatrix} R & G \\ G & B \end{smallmatrix}$, $\begin{smallmatrix} B & G \\ G & R \end{smallmatrix}$, $\begin{smallmatrix} G & R \\ B & G \end{smallmatrix}$, or $\begin{smallmatrix} G & B \\ R & G \end{smallmatrix}$. These patterns are phases of the 2-periodic CFA, offset by 0 or 1 in both directions. As demosaicing involves the reconstruction of missing data, no demosaicing method can be considered perfect, and each method introduces artefacts of some degree. As a result, these artefacts can reveal the image mosaic.

When an image is forged, the underlying mosaic of the image is altered as well. Copy-move forgeries, for instance, will displace the mosaic and might induce dephasing. Common operations in photo editing software, such as cloning and healing, often consist of multiple small copy-moves from smooth regions, the underlying mosaic of the resulting image will thus feature many small blobs of the original mosaic. Splicing from JPEG-compressed or resampled sources will make the underlying mosaic harder to detect, and might even alter its periodicity. Overall, revealing the underlying mosaic of an image provides important clues to the presence of image forgeries.

Of course, this mosaic is not explicitly known. Revealing the sampling colour of each pixel is a difficult enterprise, as the mosaic traces are hidden deep in the highest frequencies of an image. The slightest JPEG compression can dampen or even erase said traces [20], making it even more difficult to un conceal.

The recent advent of positional learning [6,7], coupled with internal fine-tuning, enabled the analysis of demosaicing traces on an image, even after a slight compression. However, even these methods remain locally inaccurate; reliable information on the mosaic can only be obtained when aggregating the method’s output over a larger scale. Simply revealing the estimated mosaic is consequently no longer enough to detect forgeries. To provide reliable detections, a method

must be able to analyse its own estimation so as to distinguish true mosaic inconsistencies from regions where the analysis is not accurate enough.

A *contrario* detection theory [14,15] provides a way to perform such an analysis. Based on the non-accidentalness principle, this theory proposes to detect data based on their unlikelihood under a background hypothesis, by thresholding the results based on a tolerated limit on the number of false alarms (NFA) under the hypothesis. This paradigm has seen successful applications in varied detection tasks [1,21,22,23,24,28,29,30], including forensics [3,8,10,5,10,17,19,32,29].

In this article, we propose MIMIC: Mosaic Integrity Monitoring for Image *a Contrario* forensics, an *a contrario* method to analyse a mosaic estimate and reliably detect image forgeries. Taking as only input the pixelwise mosaic estimate from an existing demosaicing analysis algorithm, the proposed method detects regions in which the estimate is significantly incoherent due to a shifted or even locally erased mosaic. MIMIC beats the forgery detection SOTA on high-quality images, even against a slight compression.

2 Related Works

Demosaicing analysis focused at first on linear estimation or using filters to detect inconsistencies [34,8,7]. Popescu and Farid jointly estimated a linear model and detected sampled pixels [34]. Ferrara looked for the local absence of demosaicing traces by comparing the variance between sampled and interpolated pixels [18]. Kirchner and Milani identified the sampling pattern by performing demosaicing with multiple algorithms [25,31]. Choi compared the counts of intermediate values in each lattice to estimate the correct pattern [11,4].

A *contrario* analysis is more than twenty years old [15], and has recently seen use in image forensics, primarily to attempt to detect inconsistencies in the JPEG [32] or demosaicing patterns [10,8] while controlling the risks of false positives. Blocks are made to vote for the most likely pattern, then the algorithm look for regions where one of the votes is significant enough to void the background hypothesis of an absence of demosaicing or JPEG compression. If two different patterns are significantly detected in different places, then the methods conclude to a forgery. This paradigm, however, is unable to detect regions without demosaicing traces, as is often possible in forged images, and also fails to reliably detect many forgeries even when they are visible in the vote maps.

Positional learning In AdaCFA [6], it is noted that due to translation invariance, convolutional neural networks cannot inherently detect the position of pixels, or information thereon. However, they can do so if the input image itself contains cues on the position of each pixel, as they will then learn to infer the positional information from these cues. In the case of demosaicing, the sampling mosaic of an image is 2-periodic, and demosaicing artefacts thus feature a strong 2-periodic component. If a CNN is trained on demosaiced images to detect information such as the modulo (2, 2) position (horizontally and vertically) of each pixel, it will naturally rely on the demosaicing artefacts to use them as clue to the position of the pixel. Of course, the actual position of the pixel is already known, there

is thus no need to actually detect it. What matters is that the network mimics the underlying mosaic of an image. Being trained on authentic images with integrate mosaics, the network indeed detects the correct positions of pixels in the training set. When used on a real image, however, the correctness of the output depends on the integrity of the image mosaic. If an image is authentic with an integrate mosaic, the network should correctly detect the position of pixels. More interestingly, if the tested image is locally forged, its mosaic will likely be locally altered as well. As a consequence, the network will yield incorrect outputs. If the mosaic has simply been shifted, for instance due to an internal copy-move, the model output will likewise be locally shifted on the forged area. If the mosaic is locally destroyed, for instance due to blurring or the insertion of a compressed and/or resampled object, whose mosaic is no longer visible, the network will simply render noise-like output instead of the actual position. In both cases, the fact that the network yields locally erroneous results is the very proof of a local forgery. This positional learning is introduced with AdaCFA [6] and refined with 4Point [7]. Combined with internal retraining on the very tested image, the internal mosaic of an image can be revealed even on slightly-compressed images.

3 Proposed method: MIMIC

MIMIC extends on 4Point [7], where positional learning is coupled with internal fine-tuning on tested image. A network is trained on demosaiced images to detect two features on each pixel: its diagonal offset, which corresponds to whether the pixel was initially sampled in green or in another colour, and the evenness of the line and column of pixels that are on the main diagonal, which corresponds to whether the pixel (which is at this point known not to be sampled in green) was sampled in red or blue. This formulation is equivalent to detecting the modulo 2 position of the pixels horizontally and vertically, but is more natural in terms of demosaicing. From there, 4Point introduces a simple scheme to give a confidence over the authenticity of regions in the image. The mosaic estimated by the network is satisfying even on slightly-compressed images. However, as artefacts are not always visible and are often dampened or destroyed by even slight JPEG compression, the detected positions is rarely perfect. The result of the network only lead to an estimation of the underlying mosaic, and the estimation itself is not error-free. A thorough analysis of the result is thus needed to distinguish true clues of forgeries from mere estimation errors of the network.

Starting from the same base network and training, we introduce a more robust way to analyze the output of the network. Using an *a contrario* framework, we look for regions where the estimate is significantly more erroneous than in the rest of the image. Indeed, while the mosaic may be harder to detect on some images due to the processing of an image, this difficulty should not naturally vary within an image, as such, a locally erroneous estimate is a sign of forgery.

3.1 Block votes

The network outputs two features for each pixel, namely its diagonal offset O_d and the evenness of its line offset O_l , both between 0 and 1 with extremes signifying more confidence in the result.

To bypass differences that may arise between differently-sampled pixels, the results are aggregated into 2×2 blocks, which correspond to a mosaic tile. The estimations on the four pixels of each block are averaged and compared to the midpoint, so as to return three binary outputs per block:

- $\delta \cdot \begin{smallmatrix} \text{G} & \text{G} \\ \text{G} & \text{G} \end{smallmatrix}$ represents the diagonal of the block, more precisely whether the underlying mosaic tile of the block has green-sampled pixels in the top-right and bottom-left corners ($\begin{smallmatrix} \text{G} \\ \text{G} \end{smallmatrix}$ kind, with $\delta \cdot \begin{smallmatrix} \text{G} & \text{G} \\ \text{G} & \text{G} \end{smallmatrix} = 0$) or in the top-left and bottom-right corners ($\begin{smallmatrix} \text{G} \\ \text{G} \end{smallmatrix}$ kind, with $\delta \cdot \begin{smallmatrix} \text{G} & \text{G} \\ \text{G} & \text{G} \end{smallmatrix} = 1$)
- $\delta \begin{smallmatrix} \text{R} & \text{G} & \text{B} & \text{G} \\ \text{G} & \text{B} & \text{G} & \text{R} \end{smallmatrix}$ (resp. $\delta \begin{smallmatrix} \text{G} & \text{R} & \text{B} & \text{G} \\ \text{B} & \text{G} & \text{R} & \text{G} \end{smallmatrix}$) refine on the diagonal to estimate the full pattern. They represent whether, assuming the block tile is of the $\begin{smallmatrix} \text{G} \\ \text{G} \end{smallmatrix}$ kind (resp. $gxxg$), whether it is more likely to be on a $\begin{smallmatrix} \text{R} & \text{G} \\ \text{G} & \text{B} \end{smallmatrix}$ or $\begin{smallmatrix} \text{B} & \text{G} \\ \text{G} & \text{R} \end{smallmatrix}$ tile (resp. $\begin{smallmatrix} \text{G} & \text{R} \\ \text{B} & \text{G} \end{smallmatrix}$ or $\begin{smallmatrix} \text{G} & \text{B} \\ \text{R} & \text{G} \end{smallmatrix}$).

We now know the detected diagonal and pattern of each 2×2 block in the image. The detected diagonal and pattern of the whole image is then defined as the mode of the blocks' diagonals and patterns. Let $D_g \in \{\begin{smallmatrix} \text{G} \\ \text{G} \end{smallmatrix}, \begin{smallmatrix} \text{G} \\ \text{G} \end{smallmatrix}\}$ and $P_g \in \{\begin{smallmatrix} \text{R} & \text{G} & \text{B} & \text{G} \\ \text{G} & \text{B} & \text{G} & \text{R} \end{smallmatrix}, \begin{smallmatrix} \text{G} & \text{R} & \text{B} & \text{G} \\ \text{B} & \text{G} & \text{R} & \text{G} \end{smallmatrix}\}$ denote the global diagonal and pattern of the image.

3.2 A *contrario* automatic forgery detection

Algorithm 1: Error map computation

```

1 function compute_errormap(P)
   Input P: patterns or diagonal of each block, size (X, Y).
   Output E: Error map of P, same size.
   Output Pg: Global detected pattern or diagonal.
2   Pg = mode(P)
3   E = 0X,Y
4   for x from 0 to X and y from 0 to Y do
5     if Px,y ≠ Pg then
6       Ex,y = 1
7   return E, Pg

```

In each 2×2 block of the image, we have derived an estimation of the diagonal tile and full pattern of the underlying mosaic. These estimations can be compared to the global estimations to look for forgeries. So as to avoid false positives that are solely due to misinterpretation of the detected mosaic map, we propose an *contrario* framework to automatically detect significantly deviant regions.

As proposed in [8], we could focus on regions that present a significant grid that is different from the grid of the global image. Yet, this would not enable us to detect areas with multiple small patches of different grids (as is frequently the case on inpainted images); nor would we see the localised absence of demosaicing.

Instead, we detect regions where the detection is significantly erroneous, i.e. where the network makes more mistakes than in the rest of the image. We apply the method separately on the detected diagonals and patterns. Let E_d

(resp. E_p) be a binary map which equals 1 for each block whose detected diagonal (resp. pattern) is different from D_g (resp. P_g). This is a map of the “wrong” blocks. The computation of those maps is described in Algorithm 1.

For the rest of the subsection, E represents either E_d or E_p . The empirical probability of any block on the image being wrong is denoted by p_0 , and is computed as the mean of E . We want to find regions in which the error density is significantly higher than p_0 .

Algorithm 2: NFA computation

```

1 function get_rectangle_nfa( $E, d, x_0, x_1, y_0, y_1$ )
  Input  $E$ : 1 if the block is erroneous, 0 if it is correct. Size  $(X, Y)$ .
  Input  $d$ : Downsampling coefficient, given by the radius of the linear
            estimation filters.
  Input  $x_0, x_1, y_0, y_1$ : Coordinates of the rectangle whose NFA to compute
  Output  $\epsilon$ : NFA of the rectangle
2    $p_0 = \frac{1}{X \cdot Y} \sum_{x=0}^X \sum_{y=0}^Y E_{xy}$ 
3    $n_{\text{tests}} = 2 * (X \cdot Y)^2$ 
4    $k = \sum_{x=x_0}^{x_1} \sum_{y=y_0}^{y_1} E_{xy}$ 
5    $n = (x_1 - x_0) \cdot (y_1 - y_0)$ 
6    $NFA = n_{\text{tests}} \cdot I_{p_0} \left( \frac{k}{d^2} - 1, \frac{n - k}{d^2} \right)$ 
7   return NFA

```

Let us assume that, in a given rectangle, k out of the n blocks contained in the rectangle are incorrect. Under the background hypothesis that the probability of error is p_0 , and assuming that the blocks are independent, the probability of having at least k wrong blocks in the area is the survival function of the binomial distribution $\text{Binom}_{sf}(k, n, p_0)$. Yet a first obstacle to this simple strategy arises, as the grid values of different blocks are not independent, as the neural network uses inputs that overlap between neighbouring blocks. To achieve independence, we simulate down-sampling and divide k and n by d^2 , where d is the distance between two independent outputs. We set d to 17, the radius of the CNN. To account for the fact that in the binomial integers are then replaced by floating values, we use the Beta distribution to interpolate the binomial. The probability of having at least k wrong blocks in this area is thus evaluated by

$$p_{k,n,p_0} = I_{p_0} \left(\frac{k}{d^2} + 1, \frac{n - k}{d^2} \right),$$

where I_x is the regularized incomplete Beta function. Under the *a contrario* framework, the number of tests is the possible number of rectangles in the image, that is, the number of blocks squared, multiplied by a factor 2 since we work separately on the patterns and diagonal. The NFA associated with the detection,

whose computation is described in Alg. 2, is consequently defined by

$$\text{NFA}_{k,n,p_0} = 2n_{\text{blocks}}^2 I_{p_0} \left(\frac{k}{d^2} - 1, \frac{n-k}{d^2} \right). \quad (1)$$

Algorithm 3: Forgery detection

```

1 function get_forgery_mask( $E, d, s$ )
    Input  $P$ : Patterns or diagonals of each block, size  $(X, Y)$ .
    Input  $p_b$ : 0.5 if  $P$  represents diagonal, 0.25 if it represents patterns.
    Input  $d$ : Downsampling coefficient, given here by the size of the filters.
    Input  $s$ : Stride at which to search for rectangles. Here,  $s = 16$ .
    Output  $\mathcal{D}$ : Forgery mask, each pixel represents the NFA detection score of
                its corresponding  $2 \times 2$  block.

2  $\mathcal{D} = +\infty$ 
3  $E, P_g = \text{compute\_errormap}(P)$ 
4  $E_c = \text{morphological\_closing}(E, \text{disk}_2)$ 
5 labels = label_connected( $E$ )
6 for label from 0 to max(labels) do
7      $\mathcal{M} = \text{labels}_{x,y} = \text{label}$ 
8     if  $\frac{1}{|E_c|} \sum_{x \in E_c} \mathbf{1}_{E_c}(x) > 1 - p_b$  then
9          $\epsilon = 0$  for  $\mathcal{R}$  rectangle within the bounding box of  $\mathcal{M}$  at
            step  $s$  do
10             $\epsilon_{\mathcal{R}} = \text{get\_rectangle\_nfa}(E, d, \mathcal{R})$ 
11             $\epsilon = \min(\epsilon, \epsilon_{\mathcal{R}})$ 
12             $\mathcal{M} = \text{morphological\_closing}(\mathcal{M}, \text{disk}_8)$ 
13             $\mathcal{D}_{|\mathcal{M}} = \min(\mathcal{D}_{|\mathcal{M}}, \epsilon)$ 
14 return  $\mathcal{D}$ 

```

Ideally, to detect forgeries in an image, we would compute the NFA of all the rectangles in the image. The score of a pixel would be the minimal score among the rectangles containing that pixel, and the score of an image would be that of the most significant rectangle. However, the number of rectangles scales quadratically with the number of pixels in an image. Hence, checking all possible rectangles is not possible. Even if a forgery is detected, some rectangles bigger than the forgery itself may still be significant, and the detection will therefore be too large; conversely, if part of a forgery is detected, we should detect nearby parts of the same forgery as well, even if they are not as significant as the detected part. As a consequence, we propose to first detect and separate all potential forgeries, and then to decide on their significance, so as to improve the localization of the forgeries. The method used is described in Alg. 3.

Still separately on the diagonal and full patterns, we use the map E of 2×2 blocks whose diagonal/pattern is erroneous. We apply a morphological closing to this map with a disk of size 2 to connect inconsistent blocks, and segment the resulting map into connected components. Components where the global pattern

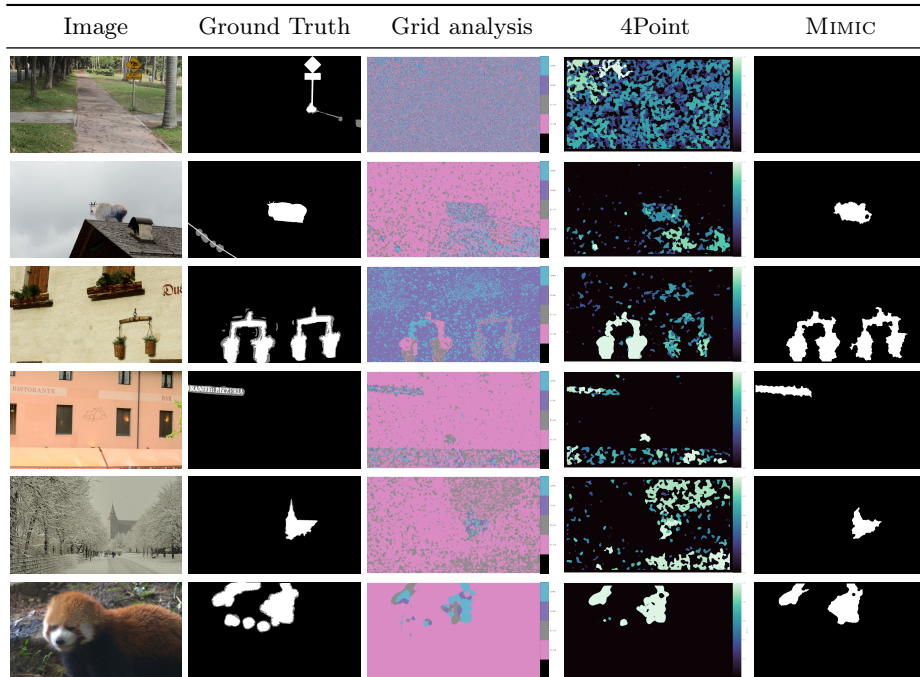


Fig. 2: Visual results on Korus forged images. In most cases, MIMIC detects the forgeries, even with inaccurate grid estimates. Although it uses the same grid estimate as 4Point, its detections are much more precise, in addition to being automatic. Its main caveat is that it misses some detections when they are too thin or diagonal, such as in the second and last columns.

(respectively diagonal) represent more than 25% (respectively 50%) of the blocks are immediately rejected and not tested for forgeries.

Each of the remaining components is tested to determine whether it is a forgery. On each component, we test all the rectangles contained within the bounding box of the component, with a step of 16 pixels. The selected striding represents a compromise between precision and computation time, as a lower stride means more rectangles need to be checked.

Finally, we keep the NFA of the most significant rectangle. We set the score of the whole component to this NFA, thus solving the final two issues addressed above: only blocks that were in the component are given this NFA, and blocks out of a significant rectangle but still in the component are kept. Forgery detection is performed separately on the full pattern and on diagonals, then the detected forgeries are merged. The final NFA map is the pointwise minimum of score maps of the patterns and diagonals NFA.

The NFA of a region is an upper bound on the expected number of regions that would be falsely detected as forged under the background hypothesis. We set the threshold to $\epsilon = 10^{-3}$, and the final, binary map keeps pixels whose NFA is below this threshold. Under the background hypothesis, the false detection

rate therefore is expected to be below one for 1000 images. Of course, this rate only concerns the risk of false positives that are due to a misinterpretation of the estimated mosaic, and does not provide further guarantees against significant errors within the estimated mosaic, which could be due to the image structure, such as the presence of textured areas, or post-processing such as resampling which would modify the mosaic traces. Still, this enables us to filter out regions which are only marginally less precisely detected than the rest of the image. The proposed *a contrario* method thus only select regions which are significantly more erroneous than the rest of the image, regardless of the reason, and provides us with a mathematically rigorous way to automatically interpret the estimated mosaic to yield an automatic detection.

4 Results

| Method | Uncompressed | JPEG $Q = 95$ | JPEG $Q = 90$ |
|-------------------|--------------|---------------|---------------|
| MIMIC | 0.724 | 0.311 | 0.196 |
| 4Point [7] | <u>0.709</u> | <u>0.307</u> | <u>0.151</u> |
| AdaCFA [6] | 0.692 | 0.005 | 0.003 |
| DDem [10] | 0.401 | 0.129 | 0.093 |
| Shin[35] | 0.104 | 0.001 | 0.001 |
| Choi[11,4] | 0.603 | 0.156 | 0.070 |
| Ferrara[18,37] | 0.071 | 0.000 | 0.000 |
| Dirik[16,37] | -0.002 | 0.000 | 0.001 |
| Park[33] | 0.116 | 0.001 | 0.000 |
| Noiseprint[13] | -0.001 | 0.004 | 0.001 |
| Splicebuster [12] | 0.003 | 0.004 | 0.001 |
| ManTraNet [2,36] | 0.000 | -0.001 | 0.002 |

Table 1: Results on the CFA Grid exomask (Grid) dataset of the Trace database, on uncompressed images and after compression with quality factors 95 and 90. The methods are grouped, after the proposed method MIMIC are methods based on demosaicing analysis, then more generic methods that do not specifically target demosaicing artefacts. Our analysis of the mosaic improves on the results of 4Point, especially on stronger compression ($Q = 90$). As already established in the literature [6,10,7,9], generic methods that do not specifically target demosaicing artefacts are entirely blind to shifts in the mosaic.

We test MIMIC on the Trace CFA Grid [9] and on the Korus [26,27] datasets. The Trace CFA Grid dataset, on which results are shown in Tab. 1 contains 1000 forged images that can only be detected by their demosaicing traces, thus enabling a comparison between demosaicing analysis methods and evaluation of the sensitivity to demosaicing artefacts of more generic methods. The Korus dataset, on which results are shown in Tab. 2 and visually in Fig. 2, features 220 forged images from four cameras.

| Method | Overall | Canon 60D | Nikon D7000 | Nikon D90 | Sony $\alpha 57$ |
|-------------------|--------------|--------------|--------------|--------------|------------------|
| Proposed | 0.472 | 0.000 | 0.595 | 0.630 | 0.662 |
| 4Point [7] | <u>0.353</u> | 0.00 | <u>0.401</u> | <u>0.378</u> | <u>0.624</u> |
| AdaCFA [6] | 0.167 | 0.002 | 0.049 | 0.044 | 0.574 |
| Shin[35] | 0.143 | 0.021 | 0.003 | 0.012 | 0.511 |
| Choi[11,4] | 0.238 | 0.004 | 0.176 | 0.251 | 0.251 |
| Ferrara[18,37] | 0.321 | -0.016 | 0.498 | 0.461 | 0.339 |
| Dirik[16,37] | 0.153 | 0.036 | 0.241 | 0.275 | 0.062 |
| Park[33] | 0.338 | 0.018 | 0.540 | 0.491 | 0.302 |
| Noiseprint[13] | 0.202 | 0.153 | 0.322 | 0.236 | 0.148 |
| Splicebuster [12] | 0.238 | 0.153 | 0.329 | 0.222 | 0.155 |
| ManTraNet [2,36] | 0.169 | 0.121 | 0.229 | 0.193 | 0.143 |

Table 2: Results on the Korus dataset of forged images. No demosaicing artefacts are found on the Canon 60D images by any of the demosaicing-based method, thus we can safely conclude they do not feature demosaicing artefacts. On all the other images, as well as overall, the proposed method significantly improves over the existing state of the art on this dataset.

Results are presented using Matthew’s Correlation Coefficient (MCC), a metric ranging from 1 (perfect detection) to -1 (opposite detection). Any input-independent method has a zero MCC expectation. The MCC can only be measured on binary detections, which is only the case for MIMIC. For the other methods, we threshold the output on the threshold that maximizes the score, giving those methods a slight advantage compared to a real case scenario, where adjusting the threshold to the data would not be possible.

Experiments on the Trace dataset show that MIMIC beats the state of the art even when the images are JPEG-compressed at quality levels 95 and 90, while generic methods are shown to be insensitive to demosaicing artefacts and inconsistencies. On the Korus dataset, one quarter of the images do not feature traces of demosaicing, as validated by all tested methods. The proposed method is thus unable to detect forgeries on this part of the dataset. Despite that, MIMIC still yields the best overall score on this dataset. We further note that on the images without traces of demosaicing, MIMIC does not make any false detection.

5 Conclusion

In this paper, we proposed MIMIC (Mosaic Integrity Monitoring for Image *a Contrario* forensics), an *a contrario* method that extends on the demosaicing analysis network 4Point [7] to refine its analysis and automatically detect image forgeries. MIMIC identifies forgeries as regions where the network output is significantly more erroneous than in the rest of the image. An *a contrario* framework helps limit the risk of false positives.

Demosaicing artefacts are frail and subtle, yet they can provide highly-valuable information to detect forgeries. On high-quality images, their sole analysis yields better results on forgery detection than any other state-of-the-art method. These results are furthermore fully complementary to non-demosaicing-specific methods, which are not sensible to demosaicing artefacts.

Acknowledgements This work has received funding by the European Union under the Horizon Europe vera.ai project, Grant Agreement number 101070093, and by the ANR under the APATE project, grant number ANR-22-CE39-0016. Centre Borelli is also a member of Université Paris Cité, SSA and INSERM. I would also like to thank Jean-Michel Morel and Rafael Grompone von Gioi for their insightful advice regarding this work.

References

1. Aguerrebere, C., Sprechmann, P., Muse, P., Ferrando, R.: A-contrario localization of epileptogenic zones in spect images. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (2009)
2. Bammey, Q.: Analysis and Experimentation on the ManTraNet Image Forgery Detector. *Image Processing On Line* **12** (2022)
3. Bammey, Q.: Jade owl: Jpeg 2000 forensics by wavelet offset consistency analysis. In: 8th International Conference on Image, Vision and Computing (ICIVC). IEEE (2023)
4. Bammey, Q., Grompone von Gioi, R., Morel, J.M.: Image Forgeries Detection through Mosaic Analysis: the Intermediate Values Algorithm. *IPOL* (2021)
5. Bammey, Q., von Gioi, R.G., Morel, J.M.: Automatic detection of demosaicing image artifacts and its use in tampering detection. In: *MIPR* (2018)
6. Bammey, Q., von Gioi, R.G., Morel, J.M.: An adaptive neural network for unsupervised mosaic consistency analysis in image forensics. In: *CVPR* (2020)
7. Bammey, Q., von Gioi, R.G., Morel, J.M.: Forgery detection by internal positional learning of demosaicing traces. In: *WACV* (2022)
8. Bammey, Q., Gioi, R.G.v., Morel, J.M.: Reliable demosaicing detection for image forensics. In: 2019 27th European Signal Processing Conference (EUSIPCO). pp. 1–5 (2019). <https://doi.org/10.23919/EUSIPCO.2019.8903152>
9. Bammey, Q., Nikoukhah, T., Gardella, M., von Gioi, R.G., Colom, M., Morel, J.M.: Non-semantic evaluation of image forensics tools: Methodology and database. In: *WACV* (2022)
10. Bammey, Q., Von Gioi, R.G., Morel, J.M.: Demosaicing to detect demosaicing and image forgeries. In: 2022 IEEE International Workshop on Information Forensics and Security (WIFS) (2022)
11. Choi, C.H., Choi, J.H., Lee, H.K.: CFA pattern identification of digital cameras using intermediate value counting. In: *MM&Sec* (2011)
12. Cozzolino, D., Poggi, G., Verdoliva, L.: Splicebuster: A new blind image splicing detector. In: *WIFS* (2015)
13. Cozzolino, D., Verdoliva, L.: Noiseprint: A cnn-based camera model fingerprint. *IEEE TIFS* (2020)
14. Desolneux, A., Moisan, L., Morel, J.: From gestalt theory to image analysis. *interdisciplinary applied mathematics*, vol. 35 (2007)
15. Desolneux, A., Moisan, L., Morel, J.M.: Meaningful alignments. *IJCV* (2000)
16. Dirik, A.E., Memon, N.: Image tamper detection based on demosaicing artifacts. In: *ICIP* (2009)
17. Ehret, T.: Robust copy-move forgery detection by false alarms control. *arXiv preprint arXiv:1906.00649* (2019)
18. Ferrara, P., Bianchi, T., De Rosa, A., Piva, A.: Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE TIFS* **7** (2012)

19. Gardella, M., Musé, P., Morel, J.M., Colom, M.: Noisesniffer: a fully automatic image forgery detector based on noise analysis. In: IWBF. IEEE (2021)
20. Gardella, M., Nikoukhah, T., Li, Y., Bamme, Q.: The impact of jpeg compression on prior image noise. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2689–2693 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9746060>
21. Grompone von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32** (2010)
22. Grompone von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a Line Segment Detector. *IPOL* **2** (2012)
23. Grompone von Gioi, R., Randall, G.: Unsupervised Smooth Contour Detection. *Image Processing On Line* **6** (2016)
24. Gómez, A., Randall, G., Grompone von Gioi, R.: A Contrario 3D Point Alignment Detection Algorithm. *Image Processing On Line* **7** (2017)
25. Kirchner, M., Fridrich, J.: On detection of median filtering in digital images. In: *MFS II* (2010)
26. Korus, P., Huang, J.: Evaluation of random field models in multi-modal unsupervised tampering localization. In: *IEEE WIFS* (2016)
27. Korus, P., Huang, J.: Multi-scale analysis strategies in prnu-based tampering localization. *IEEE Trans. on Information Forensics and Security* (2017)
28. Lezama, J., Randall, G., Morel, J.M., Grompone von Gioi, R.: An Unsupervised Point Alignment Detection Algorithm. *Image Processing On Line* **5** (2015)
29. Li, Y., Gardella, M., Bamme, Q., Nikoukhah, T., Morel, J.M., Colom, M., Grompone von Gioi, R.: A contrario detection of h.264 video double compression. In: *2023 IEEE International Conference on Image Processing (ICIP)* (2023)
30. Lisani, J.L., Ramis, S.: A Contrario Detection of Faces with a Short Cascade of Classifiers. *Image Processing On Line* **9** (2019)
31. Milani, S., Bestagini, P., Tagliasacchi, M., Tubaro, S.: Demosaicing strategy identification via eigenalgorithms. In: *ICASSP* (2014)
32. 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: *CVPRW* (2019)
33. Park, C.W., Moon, Y.H., Eom, I.K.: Image tampering localization using demosaicing patterns and singular value based prediction residue. *IEEE Access* **9**, 91921–91933 (2021). <https://doi.org/10.1109/ACCESS.2021.3091161>
34. Popescu, A.C., Farid, H.: Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing* (2005)
35. Shin, H.J., Jeon, J.J., Eom, I.K.: Color filter array pattern identification using variance of color difference image. *Journal of Electronic Imaging* (2017)
36. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In: *IEEE CVPR* (2019)
37. Zampoglou, M., Papadopoulos, S., Kompatsiaris, Y.: Large-scale evaluation of splicing localization algorithms for web images. *Multimedia Tools and Applications* **76** (2017)