A feature-based approach for image tampering detection and localization

Luisa Verdoliva, Davide Cozzolino, Giovanni Poggi
DIETI, University Federico II of Naples, Italy

***Abstract*—We propose a new camera-based technique for tampering localization. A large number of blocks are extracted off-line from training images and characterized through features based on a dense local descriptor. A multidimensional Gaussian model is then fit to the training features. In the testing phase, the image is analyzed in sliding-window modality: for each block, the log-likelihood of the associated feature is computed, reprojected in the image domain, and aggregated, so as to form a smooth decision map. Eventually, the tampering is localized by simple thresholding. Experiments carried out in a number of situation of interest show promising results.**

***Index Terms*—Forgery detection and localization, image forensics, local descriptors.**

I. Introduction

Local image descriptors have by now reached a prominent status in image processing. They have been used with success for such diverse and challenging tasks as image mining and retrieval, texture classification, face recognition, fingerprint liveness detection, steganalysis, image quality assessment. In forgery detection [1], the key idea is that suitable features can capture the deviations from the normal behavior induced by typical image forgeries, such as copy-moves or splicings. It is worth underlining that these deviations are often not perceivable by a human being, since modern image editing tools, if used with proper skill, allow one to manipulate images leaving little or no obvious artifacts, smoothing the boundary between host image and forgery to avoid abrupt transitions.

Recently, following an approach used in steganalysis [2], we have devised a powerful descriptor-based forgery detection technique [3] to tackle phase 1 of the first IEEE IFS-TC Challenge [4]. A high-pass filtering is first carried out to compute a residual image where the useless high-level information is removed and anomalies can be better detected. Then, synthetic features are computed by means of a histogram of occurrence. Given the good results obtained in detection, we designed also a sliding-window version of the same algorithm [5] to address phase 2 of the Challenge, devoted to forgery localization. Specifically, this latter algorithm was designed to detect traces left by splicings at the boundary with the host image. However, by deeper investigating, we realized that the proposed descriptor was discovering much more than the anomalies related to unnatural boundaries. In fact, it was revealing more general deviations from the typical appearance of a natural image.

We considered then the following non-exclusive hypotheses:

1) the algorithm was detecting the different camera (device, model, or brand) that generated the splicing;
2) the algorithm was detecting some forms of image processing.

Indeed, it is well-known [6] that various types of artifacts exist, specific of a manufacturer or a model or even an individual camera which, in suitable hypothesis, enable one to identify the source of a given image. Likewise, when an image is tampered with, the different processing history of its regions can be traced back. In particular, the potential of using the conditional joint distribution of residuals, evaluated by first-order differences [7], was already explored in [8] to discover traces left by median filtering. This work showed the strong detection power of this features even when a post-processing (JPEG compression) was considered.

We then set to analyze systematically this problem, devising eventually the algorithm for image tampering detection and localization, described in this paper. The proposed method requires the availability of the source camera or else of a good number of pristine images taken by it, which are the typical hypotheses of camera-based methods, like those based on sensor noise [9]. Once local statistics are learnt from the training images, the test image is analyzed in sliding-window modality to discover deviations from the model, and local distance measures are aggregated to build a decision map [5]. Unlike techniques based on sensor noise, the proposed algorithm is not influenced by the scene content, and is computationally efficient.

In next Section we discuss related work, then, in Section III, we describe the proposed algorithm. Sections IV and V are devoted to the experimental validation, conducted first in more controlled conditions, to establish some basic properties of the approach, and then for realistic forgery localization tasks. Finally, Section VI draws conclusions.

II. Related work

A large number of methods have been proposed in recent years for forgery detection and localization. Most of them look for traces of image tampering, whatever their nature, as hints of possible image manipulation.

Much work has been devoted to copy-moves [10], obtaining a pretty good accuracy with sparse descriptors [11], and much better when more complex dense descriptors are used [12], [13]. Splicings, however, are certainly harder to detect than copy-moves. Much of the current literature aims to explore some specific types of processing the forgery could have been subject to. Some methods are based on the artifacts caused by JPEG compression [14], [15], others on blur inconsistency

[16], or on revealing traces caused by resampling [17], which is a necessary operation whenever the forgery needs to be rotated or rescaled by a certain factor.

Much more general methods are those based on camera-artifacts since they do not make specific assumptions on the type of forgery, but take advantage of some peculiar characteristics of the camera under analysis. They are based, for example, on the analysis of color filter arrays and interpolation filters [18], on artifacts caused by demosaicking [19], or on the absence of the specific sensor noise (PRNU) that characterizes each camera [9], [20].

Rather than considering camera artifacts, some researchers rely on the statistics which characterize pristine natural images. Regions that have been subject to some kind of processing are then detected based on the deviation from these statistics. In [21], following [8], a feature-based procedure is outlined in order to tell apart regions subject to median filtering from region treated by other forms of processing. An analogous approach is used in [22], where a noncausal Markov model is considered in order to capture the underlying statistical characteristics of the signal. Feature-based classification and localization is also performed in [23], where blurring is detected by using features already considered for the evaluation of natural image statistics in the context of image quality assessment [24]. The key idea is that these statistics change when blurring takes place.

In all these techniques a two-class (pristine/forged) training procedure is necessary and each method focuses on a particular type of manipulation. The method proposed in this work is itself feature-based, but is not tailored to a specific type of tampering and requires training only on pristine images.

III. Proposed method

In the following we describe how features are extracted, and how they are used for detection and localization.

A. Feature extraction

We follow the three-step model already devised in [2], [3] comprising

1) computation of residuals through high-pass filtering;
2) quantization of the residuals;
3) computation of a histogram of co-occurrences.

The final histogram is the feature vector associated with the whole image, which can be used for classification. To compute the residual image we use a linear high-pass filter of the third order, which assured us a good performance in the context of forgery detection [3], defined as

$$r_{ij} = x_{i,j-1} - 3\,x_{i,j} + 3\,x_{i,j+1} - x_{i,j+2}$$

where x and r are origin and residual images, and i, j indicate spatial coordinates.

More then in the residual themselves, we are interested in their co-occurrences, which provide information on higher-order phenomena and are based on larger support areas. Of course, residuals must be first quantized and, in order to obtain a manageable number of bins in the histogram, a very small number of quantization values must be considered. As suggested in [2] we perform quantization and truncation as:

$$\widehat{r}_{ij} = \mathrm{trunc}_T(\mathrm{round}(r_{ij}/q))$$

with q the quantization step and T the truncation value. To limit the matrix size we use $T = 2$ and $q = 1$. At this point we compute co-occurrence on four pixels in a row, that is

$$C(k_0, k_1, k_2, k_3) = \\ \sum_{i,j} I(q_{i,j} = k_0, q_{i+1,j} = k_1, q_{i+2,j} = k_2, q_{i+3,j} = k_3)$$

The homologous column-wise co-occurrences are pooled with the above based on symmetry considerations, obtaining eventually a 625-bin histogram, which is reduced to little more than 300 by further symmetry arguments.

As a final step, to reduce the weight of outliers in the subsequent training and classification phases, we pass the resulting features through a square-root non-linearity. Starting from normalized histograms (unitary sum) the final features happen to have unitary L2 norm.

B. Detection/identification

We consider first the simpler detection problem, which requires to classify a whole image or image region as genuine (hypothesis H0) or tampered (hypothesis H1). We assume to know the camera which took the photos and have a large enough collection of images taken with the same camera or even the freedom to take new ones. On the contrary, we know nothing on the camera used to produce the possible forgery.

Rather than training a two-class classifier using blocks drawn from the most heterogeneous sources for hypothesis H1, we consider a model-based approach. Following the methodology used in [25], we fit the available H0 samples through a multidimensional gaussian and carry out a threshold test. To this end, we estimate the mean vector and covariance matrix of the features $\mathbf{h}_n$

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{h}_n$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{h}_n - \boldsymbol{\mu})(\mathbf{h}_n - \boldsymbol{\mu})^T$$

Then for each new feature under test, say $\mathbf{h}'$ we compute the log-likelihood w.r.t. the Gaussian model (neglecting constants)

$$L(\mathbf{h}') = (\mathbf{h}' - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{h}' - \boldsymbol{\mu})$$

and compare it with a threshold. Setting the threshold might be a challenging problem, but we do not analyze it here, and will compute performance as a function of this parameter.

C. Localization

In localization we assume to know already that a region of the image has been tampered with and want to delineate as accurately as possible its position and shape. A simple solution based on the detection procedure described above

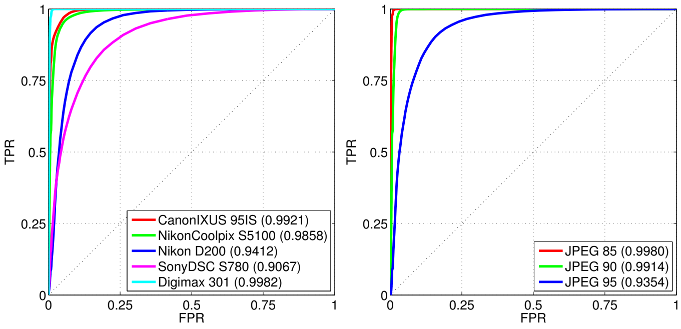

Fig. 1: Camera-based (left) and processing-based (right) detection performance using a Canon EOS 450D as target camera.

consists in using it in a sliding-window modality, with the window size $W \times W$ trading off reliability for resolution. We consider partially overlapping blocks, taken with step $1 \leq S < W$, and aggregate all decisions a map, summing -1 or +1 in the map for all pixels of the block depending on the decision, pristine or tampered, respectively. To improve the performance, rather than just the sign, we associate a real-valued strength to the block under test, depending on the reliability of the decision, so as to give more importance to clear-cut situations. By aggregating these strengths, a real valued map is generated, based on which all decisions are eventually made. In our case, it is straightforward to associate the strength with the log-likelihood itself. A threshold is eventually needed to single out the suspect tampered region but, again, we do not address this problem here, resorting to ROC curves to analyze performance.

IV. Preliminary tests on tampering detection

Although our main focus is on tampering localization, we carry out some preliminary tests for the more controlled detection case. These tests, in particular, will shed some light on the validity of the two conjectures sketched in the Introduction. We analyze them in turn.

A. *Camera-based detection*

For our experiments we have 6 cameras available, of 6 different models, produced by four manufacturers: Canon EOS 450D, Canon IXUS 95IS, Nikon D200, Nikon Coolpix S5100, Digimax 301, Sony DSC S780. For each camera we have a relatively large number of images, always more than 100, which are cropped to size 768×1024, for simplicity, aligned with the JPEG grid.

As first experiment we consider one target camera for hypothesis H0, and several more for hypothesis H1. From each available test image, not included in the training set, we extract 140 blocks of size 128×128 pixels, drawn with step 64 pixels on rows/columns. Each block is then independently classified as pristine or tampered. In Fig.1(left) we show the receiver operating curves (ROC) obtained for the Canon EOS 450D by varying the decision threshold. Results are always very good, and almost perfect in several cases. Although good results can be obtained with other approaches as well, it is worth underlining that our method is very general, does not depend on specific attributes of digital photos (e.g., CFA, quantization tables, etc.) nor is tailored to specific brands or models.

This experiment makes clear that the descriptor is indeed capturing some subtle camera-related feature and hints (a field proof is given in next section) that a splicing coming from a different camera can be very likely detected and localized. Let us therefore turn to the second conjecture: can we detect tampering based on processing history?

B. *Processing-based detection*

To test the second conjecture, we now consider the same camera for both host images and forgeries, but assume that the forgery has been subject to some type of processing before splicing, such as JPEG compression, resizing, etc., which is typically the case, both for the different history of the images and because the inserted material is often manipulated to have a natural appearance in the new context. In this case, quite a large number of combinations could be considered, but we focus only on JPEG compression, postponing to next Section a more detailed analysis. Results shown in Fig.1(right) are also in this case very good. When the test blocks are compressed with QF 85 or even 90, the ROC is almost perfect, characterized by an area under curve (AUC), shown in parentheses in the legend, exceeding 0.99. Performance becomes appreciably worse for QF 95, but remains still good (with AUC=0.93), considering also that, for the camera under test, blocks that are nominally uncompressed are actually compressed with quality factor 98.

This second experiment fully confirms also our second conjecture, so we can conclude that the proposed approach is able to accurately identify deviations from the model, even subtle deviations, due both to the different source camera and to the different processing history of the forgery.

V. Tampering localization experiments

Let us now consider some more realistic experiments with forged images, following the same path of the preceding Section. In each host image, of size 768×1024 pixel, we insert a random square forgery of size 192×192 in random positions in the image.

A. *Camera-based localization*

In a first experiment, the host image is taken by the target camera while the forgery comes from another unknown camera. Blocks of size 128×128 are drawn from the image with step 16 pixels on rows/columns. for each one we compute the log-likelihood of its feature w.r.t. the gaussian model fitted to the target camera. These quantities are then reprojected on the image and aggregated. The final map is eventually thresholded, and ROCs are computed as the threshold value λ varies. Fig.2 shows the ROCs obtained for two different host cameras (a Canon EOS 450D and a Nikon D200), again quite satisfactory. Note that all performance indicators are computed pixel-wise, therefore the curves depart from the ideal behavior not because forgeries are not localized, which never happens

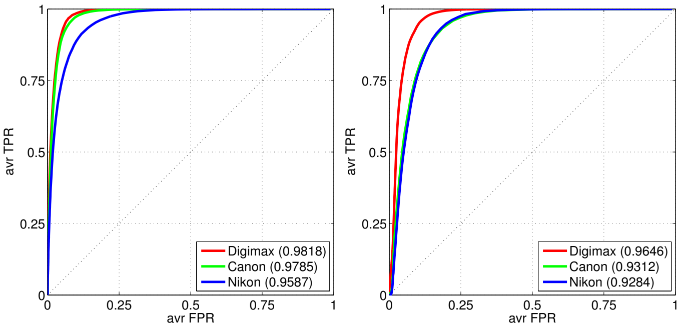

Fig. 2: Camera-based localization performance (Canon EOS 450D on the left and Nikon D200 on the right).

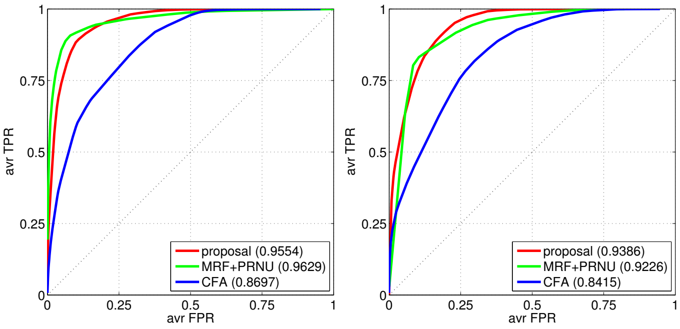

Fig. 4: Performance comparison (Canon EOS 450D on the left and Nikon D200 on the right).

in these experiments, but because of the obvious inaccuracy in detecting the exact shape of the forgery. Some examples are shown in Fig.4, last column: the forgeries are correctly localized, but their shape is recovered only approximately. Better decision strategies, as in [27], can certainly improve upon this basic procedure.

B. Processing-based localization

We now repeat the above experiment, but the forgery is generated by the same camera that took the host image. However, before insertion, the forgery undergoes some kind of processing which changes its statistics. In Fig.3 we report some results for two host cameras and four common processing tasks, that is blurring, compression, resizing, rotation. For each case, we show several ROCs obtained by changing the main parameter of interest, e.g., the quality factor in JPEG compression. Results are again very good in all cases. To observe a significant drop in performance we must consider very challenging situations, such as rotation with a very small angle, or JPEG compression with quality factor 95, and even in these cases the AUC remains near or beyond 0.80. It is worth underlining that a human being would hardly detect such high-quality forgeries by visual inspection. Some interesting phenomena, yet to investigate, concern resizing, less detectable for scales below 1 than above it, and rotation, where performance depends more strongly than expected on the angle.

C. Performance comparison

We conclude this analysis carrying out an experimental comparison with other two well-known camera-based techniques, which exploit the photo response non-uniformity (PRNU) noise, and the color filter array (CFA), respectively. In particular, for PRNU-based localization we use the algorithm recently proposed in [20], using a Bayesian framework and modern optimization techniques, while the CFA-based technique has been proposed in [26]. We built a quite varied training set, using three cameras (Canon IXUS-95IS, Nikon Coolpix-S5100, Digimax-301), and considering all combination of forgeries with and without JPEG compression, resizing, rotation, and blurring. Results are reported in Fig.4. The proposed method performs slightly better than the PRNU-based technique, and significantly better than the CFA-based one. Moreover, while the proposed method has negligible complexity, the PRNU-based technique, based on MRF modeling and optimization, has a significant run time.

Finally, we show some examples of realistic forgeries to enable a comparison of the proposed and PRNU-based techniques by visual inspection. Note that the threshold for the proposed method has been set to a fixed value equal to 1.75, while for the PRNU-based technique we considered the setting of the original paper [20]. The first two forgeries have not been processed, but come from a camera different from the host, the following two come from the host camera, but have been resized before splicing, the fifth and sixth come from a different camera with resizing, the last one from the host camera after blurring.

The proposed technique has a very good detection performance, with no false alarms, while the reference PRNU-based method occasionally (rarely) exhibits some false alarms and misses, as in some of the examples of Fig.4, selected on purpose. For the proposed technique errors concern only the limited ability to follow the shape of the forgery, affecting the pixel-based ROCs.

VI. Conclusion and Future Work

We proposed a new technique for forgery localization based on a simple modeling of natural image statistics. Despite its simplicity, it provides very good performance in a wide range of experimental conditions. Future research will include a more thorough investigation of the many design choices of the technique (e.g. features, model, decision strategy, etc.) and its optimization w.r.t. the main parameters. This analysis could help in understanding which characteristics of the camera are effectively captured by this approach. Further experimental analysis is also necessary to study robustness to subsequent processing and possible countermeasures.

References

[1] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognition*, vol. 45, pp. 4292–4299, 2012.

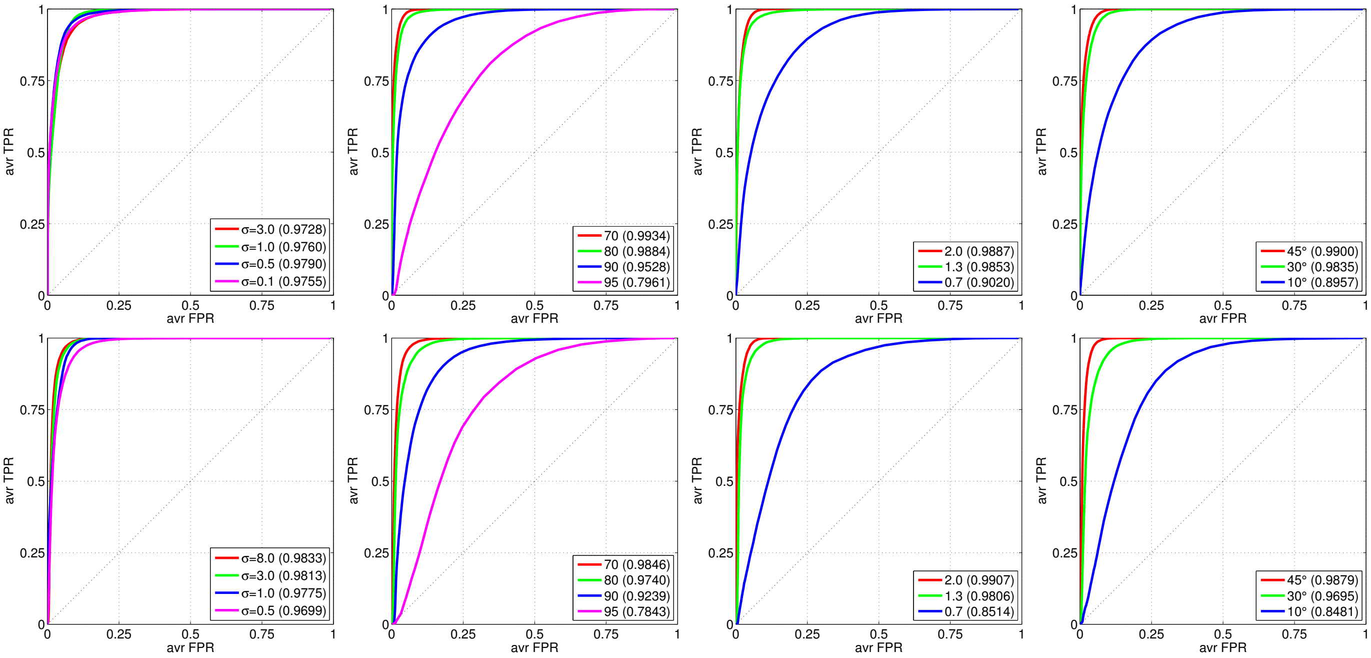

Fig. 3: Processing-based localization performance (Canon EOS 450D on the first row and Nikon D200 on the second row). From left to right: blurring, JPEG compression, resizing, rotation.

[2] J. Fridrich, and J. Kodovský, “Rich models for steganalysis of digital images,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868-882, june 2012.

[3] D. Cozzolino, D. Gragnaniello and L. Verdoliva, “Image forgery detection through residual-based local descriptors and block-matching,” *IEEE International Conference on Image Processing (ICIP)*, pp. 5297-5301, oct. 2014.

[4] http://ifc.recod.ic.unicamp.br/fc.website/index.py?sec=0.

[5] D. Cozzolino, D. Gragnaniello and L. Verdoliva, “Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques,” *IEEE International Conference on Image Processing (ICIP)*, pp. 5302-5306, oct. 2014.

[6] O. Çeliktutan, B. Sankur and I. Avcibaş, “Blind Identification of Source Cell-Phone Model,” *IEEE Trans. on Information Forensics and Security*, vol. 3, no. 3, pp. 553–566, sep. 2008.

[7] T. Pevný, P. Bas, and J. Fridrich, “Steganalysis by subtractive pixel adjacency matrix,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215-224, june 2010.

[8] M. Kirchner and J. Fridrich, “On Detection of Median Filtering in Digital Images,” *SPIE, Electronic Imaging, Media Forensics and Security XII*, vol. 7541, pp. 101–112, jan. 2010.

[9] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, “Determining image origin and integrity using sensor noise,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[10] V. Christlein, C. Riess, J. Jordan, and E. Angelopoulou, “An Evaluation of Popular Copy-Move Forgery Detection Approaches,” *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[11] X. Pan, and S. Lyu, “Region duplication detection using image feature matching,” *IEEE Trans. on Information Forensics and Security*, vol. 5, no. 4, pp. 857–867, dec. 2010.

[12] S.-J. Ryu, M. Kirchner, M.-J. Lee and H.-K. Lee, “Rotation invariant localization of duplicated image regions based on Zernike moments,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1355–1370, aug. 2013.

[13] D. Cozzolino, G. Poggi and L. Verdoliva, “Copy-Move Forgery Detection based on PatchMatch,” *IEEE International Conference on Image Processing (ICIP)*, pp. 5312-5316, oct. 2014.

[14] Y.-L. Chen, and C.-T. Hsu, “Detecting Recompression of JPEG Images via Periodicity Analysis of Compression Artifacts for Tampering Detection,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 396–406, Jun. 2011.

[15] T. Bianchi, and A. Piva, “Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts,” *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 1003–1017, Jun. 2012.

[16] Y. Sutcu, B. Coskun, H.T. Sencar and N. Memon, “Tamper Detection Based on Regularity of Wavelet Transform Coefficients,” *IEEE International Conference on Image Processing (ICIP)*, pp. 397–400, 2007.

[17] M. Kirchner, “Fast and Reliable Resampling Detection by Spectral Analysis of Fixed Linear Predictor Residue,” *Proceedings of the Multimedia and Security Workshop*, pp. 11–20, ACM Press, 2008.

[18] A. Swaminathan, M. Wu and K.J. Ray Liu, “Digital Image Forensics via Intrinsic Fingerprints,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 101–117, March 2008.

[19] A.C. Kot, “Accurate Detection of Demosaicing Regularity for Digital Image Forensics,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 899–910, Dec. 2009.

[20] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, “A Bayesian-MRF approach for PRNU-based image forgery detection,” *IEEE Trans. on Information Forensics and Security*, vol. 9, no. 4, pp. 554–567, 2014.

[21] H.-D. Yuan, “Blind Forensics of Median Filtering in Digital Images,” *IEEE Trans. on Information Forensics and Security*, vol. 6, no. 4, pp. 1335–1345, dec. 2011.

[22] X. Zhao, S. Wang, S. Li, J. Li and Q. Yuan, “Image splicing detection based on noncausal Markov model,” *IEEE International Conference on Image Processing (ICIP)*, pp. 4462–4466, sep. 2013.

[23] Z. Chen, Y. Zhao and R. Ni, “Forensics of blurred images based on no-reference image quality assessment,” *IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pp. 437–441, 2013.

[24] A. Mittal, A.K. Moorthy and A.C. Bovik, “No-Reference Image Quality Assessment in the Spatial Domain,” *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, dec. 2012.

[25] A. Mittal, R. Soundararajan and A.C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, march 2013.

[26] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, “Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts,” *IEEE Trans. on Information Forensics and Security*, vol. 7, pp. 1566–1577, 2012.

[27] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, “Guided filtering for PRNU-based localization of small-size image forgeries,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6273-6276, 2014.

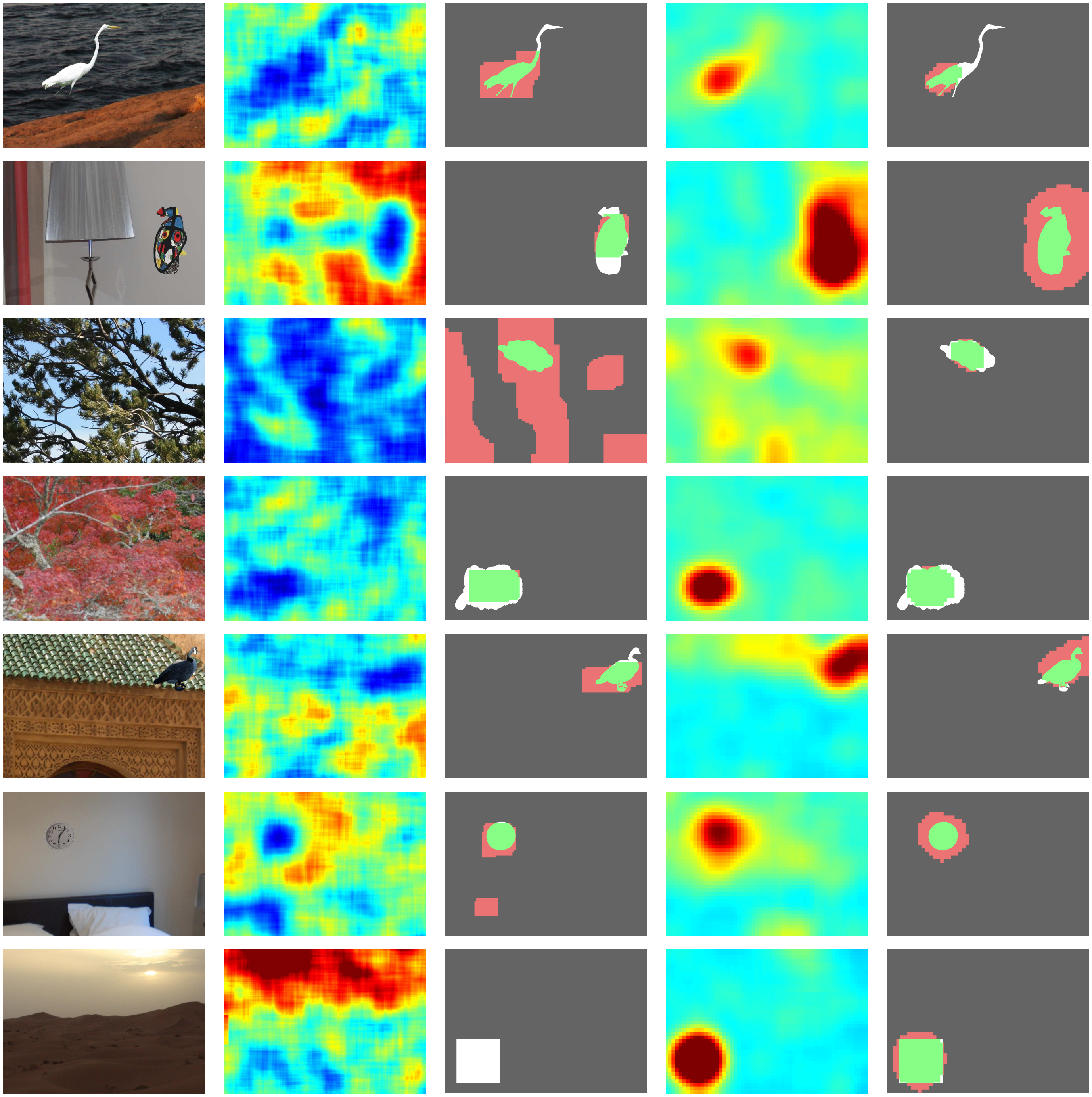

Fig. 5: Forgery localization results for some selected examples. From left to right: forged image, PRNU correlation index field and color-coded detection mask, proposed aggregation map and color-coded detection mask. Gray: genuine pixel declared genuine, red: genuine pixel declared tampered (error), white: tampered pixel declared genuine (error), green: tampered pixel declared tampered.