

UC Santa Barbara

UC Santa Barbara Previously Published Works

Title

Resampling Forgery Detection Using Deep Learning and A-Contrario Analysis.

Permalink

<https://escholarship.org/uc/item/934608dg>

Authors

Flenner, Arjuna
Peterson, Lawrence
Bunk, Jason
[et al.](#)

Publication Date

2018

Peer reviewed

Resampling Forgery Detection Using Deep Learning and A-Contrario Analysis

A. Flenner, L. Peterson; NAVAIR; China Lake, CA

J. Bunk, T. M. Mohammed, L. Nataraj, B.S. Manjunath; Mayachitra Inc.; Santa Barbara, CA

Abstract

The amount of digital imagery recorded has recently grown exponentially, and with the advancement of software, such as Photoshop or Gimp, it has become easier to manipulate images. However, most images on the internet have not been manipulated and any automated manipulation detection algorithm must carefully control the false alarm rate. In this paper we discuss a method to automatically detect local resampling using deep learning while controlling the false alarm rate using *a-contrario* analysis. The automated procedure consists of three primary steps. First, resampling features are calculated for image blocks. A deep learning classifier is then used to generate a heatmap that indicates if the image block has been resampled. We expect some of these blocks to be falsely identified as resampled. We use *a-contrario* hypothesis testing to both identify if the patterns of the manipulated blocks indicate if the image has been tampered with and to localize the manipulation. We demonstrate that this strategy is effective in indicating if an image has been manipulated and localizing the manipulations.

Introduction

Image tampering techniques have become sophisticated and in many instances experts have difficulty identifying if an image has been modified. Furthermore, with the advent of new cameras, smartphones and tablets, the amount of digital images has grown exponentially and the tools for digitally manipulating these images, such as Photoshop, Gimp, Snapseed, and Pixlr, have evolved significantly making it easy to modify many images in a short time span. Due to these advancements, the field of digital image forensics needs to develop tools that can quickly verify image authenticity and localize the regions where an image has been manipulated. In this paper, we combine our previous works on resampling detection [1] and *a-contrario* analysis [2, 3] to assign a tamper score and localize the image tampering. We demonstrate that our algorithm is effective at detecting many different types of image tampering.

Many categories of image forgeries, including copy-move, splicing, and object removal, often implement resampling as part of the forgery workflow. Resampling is required in order to blend the inserted or modified regions of the image with the rest of the image. For this reason, resampling detection is capable of detecting many different types of image manipulations and many resampling detector have been recently proposed [4, 5, 6, 7, 8, 9, 10]. However, all of these resampling detection techniques are not completely accurate for certain image regions and false alarms occur when anti-detection methods are implemented. There is a need to carefully control the false alarm rate at the image level.

In [1] we designed a detector to improve on previous re-

sampling detectors. Our detector found image artifacts imposed by classic upsampled, downsampled, clockwise rotation, counter-clockwise rotation, and shearing methods. We combined these five different resampling detectors with a JPEG compression detector and for each of the six detectors we output a heatmap corresponding to a confidence score of where the image was manipulated. We observed in our previous work that our heatmaps were noisy, and in [1] we smoothed the heatmaps to localize the detection and determine the detection score. In this work, motivated by [2, 3], we develop a new detection scheme that combines two ideas: *a-contrario* statistical hypothesis testing and image segmentation.

The *a-contrario* methodology was originally formulated by Desolneux, Moison, and Morel in order to rigorously implement the Helmholtz theory of perception [11]. Motivated by the lack of principled methods to determine an appropriate decision threshold for many basic computer vision tasks they formulated a structured statistical test to detect image primitives such as line segments and contrasted boundaries.

We were inspired to use the *a-contrario* procedure since it only requires a model, the background model, for unstructured data, i.e. heatmaps in which no resampling has occurred, and we do not need to postulate a model for heatmaps that indicate resampling has occurred. Our procedure starts by scanning the heatmap for all possible detection events, where a detection event is a location in the heatmap plus a group of pixels within the heatmap that makes it a valid detection. Since the background model is the model for unstructured data, then any detection within the event set is a false detection according to the *a-contrario* model and the goal of *a-contrario* modeling is to provide an upper bound on the number of these false detections.

Previous Work

The image forensics field recognizes many different categories of image forgeries with copy-move, splicing, and object removal as some of the most popular. In many of these forgeries, resampling is a necessary workflow element. Due to the diversity of image tampering methods, there are numerous techniques to identify if an image has been manipulated and we briefly review some of the techniques below with a special emphasis on those that overlap with our method. We then review *a-contrario* analysis.

Resampling and Image Forgery Detection

Resampling an image requires an interpolation method and linear or cubic interpolations are very popular and this fact was exploited by the authors of [4]. They implemented an Expectation-Maximization (EM) algorithm to detect periodic correlations in-

troduced by interpolation. However the periodic JPEG blocking artifacts also introduce periodic patterns that confuses their resampling detector. The variance of the second difference operator was used by [6] to detect resampling on images that are scaled using linear or cubic interpolations. Their method is most efficient at detecting up-scaling and it is very robust to JPEG compression with detection possible even at very low quality factors (QF). Downscaled images can be detected but not as robustly as up-scaled images. In [7], the Radon transform and a derivative filter was exploited to improve the quality of the results and to address other forms of resampling. In [8], a simpler method than [4] was derived by using a linear predictor residue instead of the computationally expensive EM algorithm. This resampling body of work motivated us in [1] where we combined the linear predictor strategy with deep learning based models in order to detect tampered image patches.

Other resampling detectors include [10, 12, 13, 14, 9, 15, 16]. In [10] periodic properties of interpolation were found using the second-derivative of the image and these properties were used for detecting image manipulation. Resampling on JPEG compressed images was detected in [12, 13] by added noise before passing the image through the resampling detector and they showed that adding noise improved resampling detection. A normalized energy feature was implemented in [14, 9] and a support vector machine (SVM) was subsequently used to classify resampled images. Furthermore, recent approaches to reduce the effects of JPEG artifacts were developed in [15, 16].

We tested our resampling detector on a broad range of general image tampering techniques. Most image tamper detection strategies are directed toward a specific type of image forgery such as copy-move or object insertions and removals and we review some common approaches. For copy-move forgeries a common approach is to match image features within the image. In order to detect copy-move forgeries, an image is first divided into overlapping blocks and some sort of distance measure or correlation is used to determine blocks that have been cloned. For example, in [17] copy-move forgeries were detected using SIFT features. Many similar methods to detect copy-move have been proposed [18, 19, 20, 21].

Another strategy to detect copy move forgeries is to match a transformation of image regions rather than image regions themselves. In [22], Fridrich *et. al* use DCT coefficients of image regions to obtain help find duplicate DCT blocks while Popsecu and Farid used PCA [23] to detect duplicated regions and Mahdian and Saic use a combination of blur invariant moments and PCA [24]. A matching image regions to detect copy-move forgeries becomes more difficult if the moved region undergoes some transformation such as scalings that makes region matching difficult. Bayram *et. al* [25] addresses this issue by using a combination of Fourier Mellin transforms, which are invariant to rotation, scale and translation, and Bloom filters. Another issue in locating copy-move forgeries is the computational time to find matching patches. In [26], the patch-match algorithm is used to efficiently compute an approximate nearest neighbor field over an image. They added robustness to their algorithm by using invariant features such as Circular Harmonic transforms and show that they can detect duplicated blocks that have undergone geometrical transformations and then perform keypoint matching.

In [27], an image splicing detection technique has been pro-

posed using visual artifacts. A novel image forgery detection method is presented in [28] based on the steerable pyramid transform (SPT) and local binary pattern (LBP). The paper [29] includes the recent advances in image manipulation and discusses the process of restoring missing or damaged areas in an image. In [30], the authors review the different image forgery detection techniques in image forensic literature.

In computer vision, deep learning shows outstanding performance in different visual recognition tasks such as image classification [31], and semantic segmentation [32]. For this reason, there has been a growing interest to detect image manipulation by applying different computer vision and deep-learning algorithms [33, 34, 35, 36]. In [32], two fully convolution layers have been exploited to segment different objects in an image. The segmentation task has been further improved in [37, 38]. These models extract hierarchical features to represent the visual concept, which is useful in object segmentation. Since, the manipulation does not exhibit any visual change with respect to genuine images, these models often do not perform well in segmenting manipulated regions.

Other deep learning methods include detection of generic manipulations [33, 34], resampling [39], splicing [35] and bootleg [40]. In [41] the authors propose Gaussian-Neuron CNN (GNCNN) for steganalysis detection. A deep learning approach to identify facial retouching was proposed in [42]. In [43], image region forgery detection has been performed using a stacked auto-encoder model. In [33], a new constrained convolutional layer is proposed to learn the manipulated features from an image. In [1] an unique network exploiting convolution layers along with LSTM network was presented.

A-Contrario Analysis

Our approach to determining an image manipulation score and localizing manipulated regions is motivated by *a-contrario* change detection algorithms. Many change detection algorithms propose to find differences between two images by registering the two images and then finding the magnitude of the difference between the two images. Registration error plus illumination differences introduce mistakes in such change detection algorithms, and thus controlling false alarms are a priority. In this work, we replace the difference between the two images with a heatmap generated using our resampling detector. Since our procedure is a modified change detection algorithm, we review the different *a-contrario* change detection methods. Furthermore, for simplicity of discussion and to tie in with this work, we refer to the image difference in the change detection algorithms as a heatmap.

A-contrario methods were popularized in image processing by Desolneux, Moisan and Morel, and this procedure was initially applied to identifying edges, alignments, one dimensional histogram modes, and highly contrasted boundaries [44, 45, 46, 47, 48, 11]. Many problems have been addressed using the *a-contrario* methods other than change detection. These applications include, but are not limited to, automated color segmentation [49], mode detection of two dimensional histograms [2], change detection [3], clustering [50], and matching local features [51]. Fundamental to *a-contrario* methods is the identification of a background probability model \mathcal{H}_0 , a set of detection events \mathcal{E} , and a multiple hypothesis testing procedure that bounds the number of false alarms.

The earliest work on *a-contrario* methods for change detection is by Lisani and Morel [52] where they use the binomial distribution as the background model and a sliding window to determine events. Their method is sensitive to the window size and cannot locate changes of arbitrary shapes. The adapted windows technique of Dibos *et al.* [53] uses *a-contrario* methods to detect changes in moving video. They are interested in real time processing, and therefore they use a sliding window method. The use of sliding windows reduces processing time, but it also has less precise boundary information. Veit *et al.* [54] used the *a-contrario* methods with the binomial distribution as the background model and level sets with highly contrasted boundaries to determine possible events. Using contrasted boundaries allows one to find arbitrary shapes in the image, but the number of events is limited. The *a-contrario* method in [3] built on these ideas but used every possible image level set to determine the set of events. The paper by Buades *et al.* also explores using the *a-contrario* methods for change detection applications [55]. They also use sliding windows and histogram analysis is used to determine the heatmap.

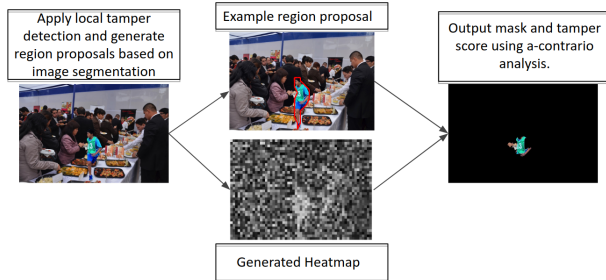


Figure 1. The algorithm workflow. First, a heatmap that indicates resampling is generated using the deep learning models in Figure 2. *a-contrario* analysis, illustrated in the last two images, is then used to localize the resampling and determine a total image score.

Methods and Results

Our procedure to determine manipulated regions consists of two primary steps summarized in Fig. 1. First, a neural net classifier is applied to resampling features to determine six different heatmaps. The heatmap generation was first reported in [1] and we leave the details of the computations to that paper. Second, *a-contrario* analysis is applied to each heatmap and an image score is obtained using information from all the heatmaps. The complete *a-contrario* algorithm is given in [2] and only an outline of the algorithm will be presented here.

Resampling Heatmap Generation

Our initial step is to generate heatmaps that indicate the five different resamplings required for upsampling, downsampling, shearing, clockwise rotation and counterclockwise rotation. Furthermore, we included a JPEG compression classifier that determines if the input has been compressed at a JPEG quality factor of 85 or less. The five resampling and one JPEG classifier generate six different heatmaps. In order to train our six heatmaps we use two different procedures. The first procedure is to extract resampling features from the image and use a deep convolutional neural network to train a classifier on these features. The second

procedure is to train a classifier on the image itself and Fig. 3 summarizes the two procedures labeled model I and model II. In particular, we use model I for rescaling down, rotate clockwise, rotate counterclockwise, and shearing while model II was used for rescaling up and jpeg quality detection. We chose the model by training both models for all manipulation types and picking the model that gave the best area under the curve (AUC) value on a receiver operating characteristics (ROC) curve.

In order to localize the manipulations, we do not train the classifiers on the entire image, but rather train the classifiers on 64x64 image patches. In order to generate features we apply a 3x3 Laplacian filter to these patches and output the linear predictive error magnitude of the patches. Correlations in the linear predictor error are then found using the Radon transform followed by a 2-D FFT. We trained the classifiers using uncompressed images from the UCID dataset [56] and RAISE data set [57] where we simulated the different resampling methods and JPEG compression. Both data sets were disjoint from our final testing data set. Note that the classifiers are not mutually exclusive and we did not use a multiclass model but trained each classifier separately. More details about patch size selection and algorithm design can be found in [1]. Note that since the classifier classifies image blocks with a pixel stride of nine the resulting heatmap's width and height is smaller than the original image.

Tamper Score and Mask determination using *a-contrario* analysis

Once the heatmap is generated, *a-contrario* analysis is used to simultaneously determine an image score and to localize the tamper areas. This procedure requires a background probability model for when no tampering occurs, which we will call \mathcal{H}_0 , a set of possible manipulation events \mathcal{E} , and a decision function based on the background model whose domain are the events.

Recall that the model \mathcal{H}_0 serves as a model when no structure exists within the data, and the set of detection events are possible locations and grouping that correspond to data structures. For example, in line segment detection, locations are every pixel in the image and structures are straight lines in the image. Given any event $E \in \mathcal{E}$, we can calculate the probability of E occurring under the hypothesis \mathcal{H}_0 . Given any probability threshold, we should expect to find false alarms. The definition of meaningful events bounds the expected number of false alarms by the number one:

Definition: Given a probability model \mathcal{H}_0 , the events in \mathcal{E} are called meaningful events if the \mathcal{H}_0 -expected number of events observed is less than or equal to one.

Note that the cardinality of \mathcal{E} can be large, so this technique must correct for multiply comparisons. Furthermore, due to the

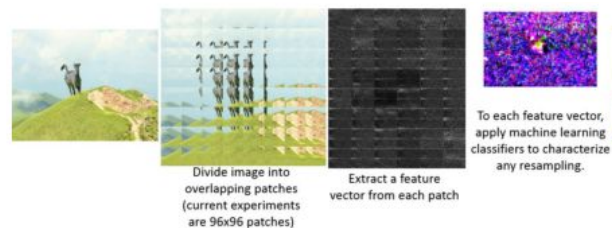


Figure 2. Patch Feature Extraction based on the Radon Transform

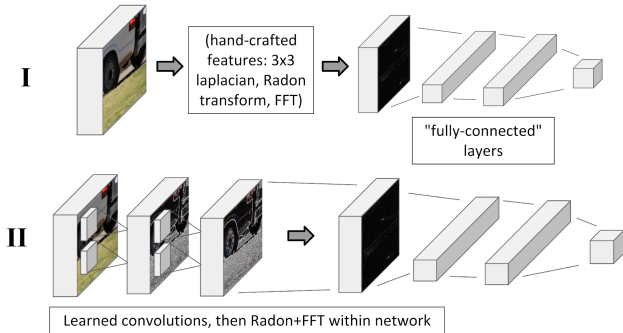


Figure 3. Deep Neural networks for detecting resampling in small patches.

large number of event calculations required in a typical applications, it is common to use a large deviation bound in computations and we used the Höfding bound for Bernoulli random variables in our work [58].

The *a-contrario* procedure has two inputs. The first input is one of the six heatmaps where each tested image block is mapped to a tampering confidence score between zero and one where one means high confidence that the image was manipulated. Let M indicate this mask. The second input is the original image I . The *a-contrario* procedure thresholds the original mask M to create a binary 0-1 mask and then refines the mask using region proposals derived from the original image. Let TM be the mask after applying the threshold. Two region proposals were used in the analysis: level set and deep mask proposals. Given the two inputs, the ini-

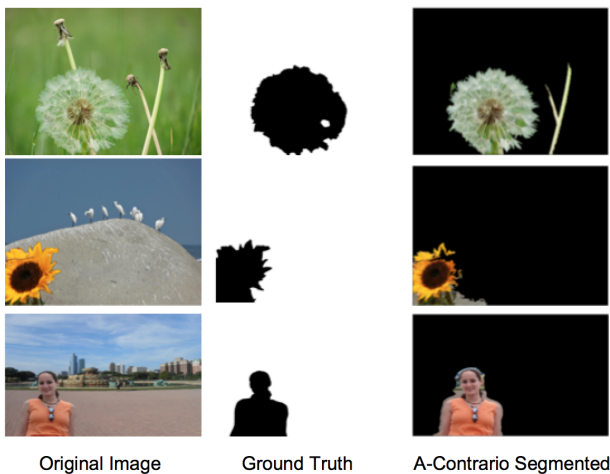


Figure 4. Example results of the *a-contrario* segmentation.

tial step is to threshold the heatmap resulting in a binary mask of zeros and ones with one indicating the block was tampered. We then assume a Bernoulli model where p is the probability of a heatmap pixel is one. The value of p is determined by counting the number of pixels in the heatmap that are one and dividing by the total number of pixels. In other words, in order to determine if a set is spatially grouped, we assume that the pixels of TM are uniformly distributed on the image and look for regions that do not agree with that hypothesis. This forms our background model \mathcal{H}_0

Given this binary mask, we assume tampering occurs when pixels in TM cluster spatially. Ideally, we would search every possible connected group of pixels to find clusters of manipulated

patches, but this is computationally in-feasible. We therefore use the fast level set transform [59] and deep mask [60] to segment the image image. The output of deepmask is a set of connected pixels that correspond to a semantically segmented object in the image. The output of the fast level set transform is a connected region where the boundary of the region has constant pixel value. We use both methods since deepmask is not trained on all possible regions that we may want to segment, therefore just using deepmask results in missed detection, and the fast level set transform will not always find semantically reasonable regions.

The segmented image regions are used to limit the number of connected regions required to search. An event in \mathcal{E} is therefore the region determined by the image together with the count of pixels that are one and the count of pixels that are zero inside the these regions.

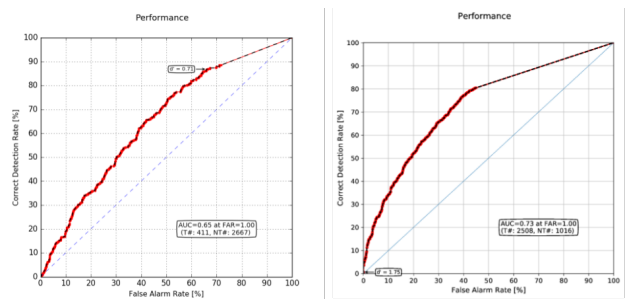


Figure 5. ROC curve for *a-contrario* analysis on the right compared with the method in [1] on the left.

Data: Resampling Heatmap M , Original Image I

Result: Segmented Image

$\{S\} \leftarrow FLST(I) \cup DeepMask(I)$

$TM \leftarrow M > c$

$L \leftarrow$ empty list.

forall $S \in \{S\}$ **do**

$r(S) = \sum_{(i,j) \in S} TM(i,j)$

$NFA(S) = \#S \Pr(R > r(s))$

if $NFA(S) < 1$ **then**

$L.append(S)$

end

end

Cull L to a list of disjoint sets.

Output L .

Algorithm 1: Pseudocode to generate a score and mask from each heatmap.

Let S be a region proposal, i.e. a segmented region of the image, obtained either through the level set analysis or through the deep mask algorithm. We use a Bernoulli process to determine if the region has been tampered by counting the number of tampered blocks within the region using the formula $r(S) = \sum_{(i,j) \in S} TM(i,j)$. We then calculate the decision function $NFA(S) = \#S \Pr(R > r(s))$ where $\#S$ is the total number of region proposals and the probability is the tail of the binomial distribution with probability of success p . Note that $NFA(S)$ stands for the number of false alarms and using the *a-contrario* model all meaningful events are unlikely to occur due to chance since $NFA(S)$ is an upper bound on the expected number of false

alarms. Each meaningful region satisfies $NFA(S) < 1$ and is flagged as a potentially manipulated region. The regions are then further refined by determining a disjoint set of regions that have the minimal $NFA(S)$ values. See [3] for a discussion on how to obtain the disjoint regions.

Following this procedure, a score and heatmap is generated for each of the six heatmaps creating a six dimensional vector with each element between zero and one. A final image score was derived by averaging the non-zero elements of this vector. If all the elements were zero, the output score was zero. The final mask is the union of the masks obtained from the non-zero elements. Example mask results are shown in Fig. 4.

We evaluate the proposed model on NIST Nimble 2017 dataset for the Media Forensics challenge. This dataset includes mainly three types of manipulation: (a) copy-clone, (b) removal, and (c) splicing. The images are tampered in a sophisticated way to beat current state-of-the-art detection techniques. The results in Fig. 5 show the efficacy of the proposed model for different images in Nimble 2016 dataset. Compared with the results from [1], where the same feature set was used but a different image scoring method was implemented, the area under the curve metric on the ROC curve increased by .08 using the *a-contrario* method.

Conclusions and Future Work

In this paper we described a technique to determine and image tamper score and a mask using resampling features, deep neural networks, and *a-contrario* analysis. We demonstrated that the *a-contrario* procedure can increase the AUC measurement over previous work. We note that the *a-contrario* procedure is applicable to all heatmaps bounded between zero and one and does not require the resampling heatmaps presented in this paper.

There is room to improve the *a-contrario* procedure presented here through parameter tuning and better integration of the output scores from the heatmaps. In particular, we applied a threshold to the heatmaps using a value of 0.75 and did not attempt to optimize this threshold. Furthermore, we used a simple mean value to combine the information from the heatmaps. As mentioned above, the heatmaps are not independent and a more careful statistical analysis should obtain a better final output score.

Acknowledgments

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. The paper is approved for public release, distribution unlimited.

References

- [1] J. Bunk, J. H. Bappy, T. M. Mohammed, *et al.*, "Detection and localization of image forgeries using resampling features and deep learning," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, 1881–1889, IEEE (2017).
- [2] A. Flenner, G. A. Hewer, and C. S. Kenney, "Two dimensional histogram analysis using the helmholtz principle," *Inverse Problems & Imaging* **2**(4), 485–525 (2008).
- [3] A. Flenner and G. Hewer, "A helmholtz principle approach to parameter free change detection and coherent motion using exchange-

- able random variables," *SIAM Journal on Imaging Sciences* **4**(1), 243–276 (2011).
- [4] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on signal processing* **53**(2), 758–767 (2005).
- [5] S. Prasad and K. Ramakrishnan, "On resampling detection and its application to detect image tampering," in *2006 IEEE International Conference on Multimedia and Expo*, 1325–1328, IEEE (2006).
- [6] A. Gallagher, "Detection of linear and cubic interpolation in JPEG compressed images," in *Computer and Robot Vision, 2005. Proc. The 2nd Canadian Conference on*, 65–72 (2005).
- [7] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *Information Forensics and Security, IEEE Transactions on* **3**, 529–538 (2008).
- [8] M. Kirchner, "On the detectability of local resampling in digital images," *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X* **6819**(1), 68190F, SPIE (2008).
- [9] X. Feng, I. J. Cox, and G. Doerr, "Normalized energy density-based forensic detection of resampled images," *IEEE Transactions on Multimedia* **14**(3), 536–545 (2012).
- [10] S.-J. Ryu and H.-K. Lee, "Estimation of linear transformation by analyzing the periodicity of interpolation," *Pattern Recognition Letters* **36**, 89–99 (2014).
- [11] A. Desolneux, L. Moisan, and J. Morel, "Meaningful alignments," *Int. J. of Comput. Vision* **40**(1), 7–23 (2000).
- [12] L. Nataraj, A. Sarkar, and B. S. Manjunath, "Improving re-sampling detection by adding noise," in *SPIE, Media Forensics and Security*, **7541** (2010).
- [13] L. Nataraj, A. Sarkar, and B. S. Manjunath, "Adding gaussian noise to "denoise" JPEG for detecting image resizing," in *ICIP*, (2009).
- [14] X. Feng, I. J. Cox, and G. Doerr, "An energy-based method for the forensic detection of re-sampled images," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, 1–6, IEEE (2011).
- [15] S. A. Golestaneh and D. M. Chandler, "Algorithm for jpeg artifact reduction via local edge regeneration," *Journal of Electronic Imaging* **23**(1), 013018–013018 (2014).
- [16] Y. Kwon, K. I. Kim, J. Tompkin, *et al.*, "Efficient learning of image super-resolution and compression artifact removal with semi-local gaussian processes," *IEEE transactions on pattern analysis and machine intelligence* **37**(9), 1792–1805 (2015).
- [17] I. Amerini, L. Ballan, R. Caldelli, *et al.*, "A sift-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security* **6**(3), 1099–1110 (2011).
- [18] J. Li, X. Li, B. Yang, *et al.*, "Segmentation-based image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security* **10**(3), 507–518 (2015).
- [19] P. Kakar and N. Sudha, "Exposing postprocessed copy-paste forgeries through transform-invariant features," *IEEE Transactions on Information Forensics and Security* **7**(3), 1018–1028 (2012).
- [20] M. Jaberri, G. Bebis, M. Hussain, *et al.*, "Accurate and robust localization of duplicated region in copy-move image forgery," *Machine vision and applications* **25**(2), 451–475 (2014).
- [21] O. M. Al-Qershi and B. E. Khoo, "Passive detection of copy-move forgery in digital images: State-of-the-art," *Forensic science international* **231**(1), 284–295 (2013).
- [22] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," in *in Proceedings of Digital Forensic Research Workshop*, Citeseer (2003).

- [23] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," Tech. Rep. TR2004-515, Department of Computer Science, Dartmouth College (2004).
- [24] B. Mahdian and S. Saic, "Detection of copy-move forgery using a method based on blur moment invariants," *Forensic science international* **171**(2), 180–189 (2007).
- [25] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 1053–1056, IEEE (2009).
- [26] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy-move forgery detection," *IEEE Transactions on Information Forensics and Security* **10**(11), 2284–2297 (2015).
- [27] V. Manu and B. Mehtre, "Visual artifacts based image splicing detection in uncompressed images," in *Computer Graphics, Vision and Information Security (CGVIS), 2015 IEEE International Conference on*, 145–150 (2015).
- [28] G. Muhammad, M. H. Al-Hammadi, M. Hussain, *et al.*, "Image forgery detection using steerable pyramid transform and local binary pattern," *Machine Vision and Applications* **25**(4), 985–995 (2014).
- [29] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE signal processing magazine* **31**(1), 127–144 (2014).
- [30] M. D. Ansari, S. P. Ghrera, and V. Tyagi, "Pixel-based image forgery detection: A review," *IETE journal of education* **55**(1), 40–46 (2014).
- [31] B. Zhou, A. Lapedriza, J. Xiao, *et al.*, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 487–495 (2014).
- [32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440 (2015).
- [33] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 5–10 (2016).
- [34] B. Bayar and M. C. Stamm, "Design principles of convolutional neural networks for multimedia forensics," in *The 2017 IS&T International Symposium on Electronic Imaging: Media Watermarking, Security, and Forensics*, IS&T Electronic Imaging (2017).
- [35] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*, 1–6, IEEE (2016).
- [36] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, *et al.*, "Exploiting spatial structure for localizing manipulated image regions," in *Proceedings of the IEEE International Conference on Computer Vision*, (2017).
- [37] S. Zheng, S. Jayasumana, B. Romera-Paredes, *et al.*, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 1529–1537 (2015).
- [38] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for scene segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(12) (2017).
- [39] B. Bayar and M. C. Stamm, "On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection," in *Proceedings of The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE (2017).
- [40] M. Buccoli, P. Bestagini, M. Zanoni, *et al.*, "Unsupervised feature learning for bootleg detection using deep learning architectures," in *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*, 131–136, IEEE (2014).
- [41] Y. Qian, J. Dong, W. Wang, *et al.*, "Deep learning for steganalysis via convolutional neural networks," in *SPIE/IS&T Electronic Imaging*, International Society for Optics and Photonics (2015).
- [42] A. Bharati, R. Singh, M. Vatsa, *et al.*, "Detecting facial retouching using supervised deep learning," *IEEE Transactions on Information Forensics and Security* **11**(9), 1903–1913 (2016).
- [43] Y. Zhang, L. L. Win, J. Goh, *et al.*, "Image region forgery detection: A deep learning approach," in *Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016: Cyber-Security by Design*, **14**, 1 (2016).
- [44] A. Desolneux, L. Moisan, and J. Morel, *From gestalt theory to image analysis: a probabilistic approach*, Springer Verlag (2007).
- [45] A. Desolneux, L. Moisan, and J. M. Morel, "Edge detection by Helmholtz principle," *Journal of Mathematical Imaging and Vision* **14**, 271–284 (2001).
- [46] A. Desolneux, L. Moisan, and J. M. Morel, "Computational gestalts and perception thresholds.," *J. Physiol Paris* **97**(2-3), 311–324 (2003).
- [47] A. Desolneux, L. Moisan, and J. M. Morel, "Maximal meaningful events and applications to image analysis," *Ann. of Statist.* **31**(6), 1822–1851 (2003).
- [48] A. Desolneux, L. Moisan, and J. M. Morel, "A grouping principle and four applications," *IEEE Trans. on Pattern Analy. and Mach. Intell.* **26**, 1105–1111 (2004).
- [49] J. Delon, A. Desolneux, J. L. Lisani, *et al.*, "Automatic color palette," *Inverse Probl. and Imaging* **1**(2), 265–287 (2007).
- [50] P. Musé, F. Sur, F. Cao, *et al.*, "An a contrario decision method for shape element recognition," *Int. J. Comput. Vision* **69**(5), 860–872 (2006).
- [51] A. Rabin, J. Delon, and Y. Gousseau, "A statistical approach to the matching of local features," *SIAM J. Imaging Sciences* **2**(3), 931–958 (2009).
- [52] J. L. Lisani and J. M. Morel, "Detection of major changes in satellite images," *Proc. IEEE Int. Conf. on Image Processing* **1**, 941–943 (2003).
- [53] F. Dibos, G. Koepfler, and S. Pelletier, "Adapted windows detection of moving objects in video scenes," *SIAM J. Imaging Sciences* **2**(1) (2009).
- [54] T. Veit, F. Cao, and P. Boutheymy, "An a contrario decision framework for region-based motion detection," *Int. J. of Comput. Vision* **68**(2), 163–178 (2006).
- [55] A. Buades, J. Lisani, and L. Rudin, "Adaptive change detection," *Systems, Signals and Image Processing - IWSSIP 2009* **1**, 1–4 (2009).
- [56] G. Schaefer and M. Stich, "Ucid: An uncompressed color image database," in *Electronic Imaging 2004*, 472–480, International Society for Optics and Photonics (2003).
- [57] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, *et al.*, "Raise: a raw images dataset for digital image forensics," in *Proceedings of the 6th ACM Multimedia Systems Conference*, 219–224, ACM (2015).
- [58] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American statistical association* **58**(301), 13–30 (1963).

- [59] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing* **9**(5), 860–872 (2000).
- [60] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 1990–1998 (2015).

age processing, computer vision, biomedical image analysis, and large-scale multimedia databases.

Author Biography

Dr. Arjuna Flenner received his Ph.D. in Physics at the University of Missouri-Columbia located in Columbia MO in the year 2004. His major emphasis was mathematical Physics. After obtaining his Ph.D., Arjuna Flenner obtained a job as a research physicist for NAVAIR at China Lake CA. He won the 2013 Dr. Delores M. Etter Navy Scientist and Engineer award for his work on Machine Learning.

Lawrence A. Peterson has extensive research and development experience as a civilian engineer / scientist with the U. S. Navy for over 37 years. He currently heads the Image and Signal Processing Branch in the Physics and Computational Sciences Division, Research Directorate at the Naval Air Warfare Center, Weapons Division (NAWCWD), China Lake, CA. Additional development experience includes signals and imagery exploitation, multi-modal target detection and integrated guidance applications, software development for radar and ESM systems, IR/EO and RF propagation modeling and simulation.

Jason Bunk received his B.S. degree Computational Physics from the University of California, San Diego in 2015, and the M.S. degree in Electrical and Computer Engineering from the University of California, Santa Barbara in 2016. He is currently a Research Staff Member at Mayachitra Inc., Santa Barbara, CA. His recent research efforts include applying deep learning techniques to media forensics, and active learning with neural networks for video activity recognition.

Tajuddin Manhar Mohammed received his B.Tech (Hons) degree in Electrical Engineering from Indian Institute of Technology (IIT), Hyderabad, India in 2015 and his M.S. degree in Electrical and Computer Engineering from University of California Santa Barbara (UCSB), Santa Barbara, CA in 2016. After obtaining his Masters degree, Tajuddin Manhar Mohammed obtained a job as a Research Staff Member for Mayachitra Inc., Santa Barbara, CA. His recent research efforts include developing deep learning and computer vision techniques for image forensics and cyber security.

Lakshmanan Nataraj received his B.E degree in Electronics and Communications Engineering from Anna university in 2007, and the Ph.D. degree in the Electrical and Computer Engineering from the University of California, Santa Barbara in 2015. He is currently a Research Staff Member at Mayachitra Inc., Santa Barbara, CA. His research interests include malware analysis and image forensics.

B. S. Manjunath (F05) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1991. He is currently a Professor of Electrical and Computer Engineering with the University of California at Santa Barbara, Santa Barbara, CA, USA, where he directs the Center for Multi-Modal Big Data Science and Healthcare. He has authored or co-authored about 300 peer-reviewed articles. He is a Co-Editor of the book entitled *Introduction to MPEG-7* (New York: Wiley, 2002). His current research interests include im-