Single-image splicing localization through autoencoder-based anomaly detection

Davide Cozzolino and Luisa Verdoliva
DIETI, University Federico II of Naples, Italy
Email: {davide.cozzolino,verdoliv}@unina.it

***Abstract*—We propose a new method for single-image splicing localization. Lacking any side information, such as training data or prior knowledge on the source camera and the image history, we cast the problem as an anomaly detection task. Expressive local features, extracted from the noise residual of the image, feed an autoencoder which generates an implicit model of the data. By iterating discriminative feature labeling and autoencoding, the implicit model fits eventually the pristine data, while the spliced region is recognized as anomalous. Experiments on a suitable test set of spliced images show that the proposed method outperforms the previous state-of-the-art. In addition, it exhibits a good robustness against typical social net post-processing, showing promises for real-world applications.**

I. Introduction

A huge number of photos are posted every day on social media. For the large majority, they are genuine images used by people to share a part of their lives with friends. However, more and more often images are manipulated and diffused with malicious purposes, like, for example, influencing the public opinion on sensitive political or religious issues. On the other hand, manipulating images has never been easier, thanks to the abundance of material on the web and to powerful media editing tools. Image splicing, also known as photo composition, is the most common form of forgery. It consists in inserting some alien material in a host image. Even though some care is required to guarantee coherent illumination, perspective, and scale between the two images, results can be extremely realistic. A large number of cases can be found on the web [1], and often the manipulations may be hardly perceived by visual inspection. This is apparent for example in Figure 1, where two images of the IEEE Image Forensics Challenge database are shown.

The ability to reveal such forgeries is becoming even more important in a number of applicative fields. Consequently, there is a growing research activity in the scientific community especially on the forgery localization task. Some of the proposed methods rely on machine learning [2], [3], reporting quite a good performance. However, such methods depend intrinsically on the availability and quality of training data, not always granted. Moreover, they cannot generalize beyond cases learned in the training set, and therefore cannot handle properly new situations. Others, e.g. [4], [5], [6] consider a one-class approach, for example based on the knowledge of a certain number of photos taken by a specific camera. In both cases, strong hypotheses are made often not met in practice.

Fig. 1: Two examples of spliced images [9]. In both cases, the person on the left has been added to the original photo.

Single-image methods, which do not require training images, have been already proposed, but they are generally based on very specific hypotheses. For example, [7] exploits the artifacts arising from double JPEG compression, while other papers [8] rely on the fingerprints left by internal camera processes, depending on specific color filter arrays and interpolation filters. Such methods are therefore intrinsically sensitive only to some specific manipulations. In addition, they typically rely on a mathematical model of the pristine image or on some specific assumptions on the in-camera processing. Hence, when the image undergoes a global distortion, due to common forms of post-processing, like resizing and re-compression, these basic hypotheses do not hold anymore and performance drops dramatically.

A more robust approach is followed in [10]. Assuming that the spliced region has a different noise power than the target image, the analysis of local noise power is used to detect candidate forgeries. However, the spliced region and the target image differ under many more respects than just noise. Once the image content is removed, the so-called noise residual contains specific micro-patterns, due to both in-camera and out-camera processing, that may enable reliable analyses. Accordingly, in [11] this wealth of information is compacted in expressive local features [12], which are assumed to follow two different distributions, multivariate Gaussian for the target image and uniform for the splicing. Splicing localization and model learning are then pursued jointly by means of the expectation-maximization algorithm. This latter method exhibits promising performance, but it also presents some weaknesses. In particular, the reliance on specific models for the class distributions and the need to set a decision threshold to obtain the binary localization map.

978-1-5090-1138-4/16/$31.00 ©2016 IEEE

In this work we use the same local features as in [11] but recast the problem in terms of anomaly detection, adopting the approach proposed in [13] for a related problem. The explicit distribution of the data, which requires parameter estimation, is replaced by an autoencoder which, in its hidden layer, provides an implicit data model, learned from the data themselves. Features drawn from pristine areas of the image conform this implicit model and are reconstructed in the output layer with very small error. On the contrary, features coming from the spliced area, depart from the model (they are anomalous) and are therefore reconstructed with large error. The iterative analysis and clustering of reconstruction errors allows eventually to localize the forgery.

In the following, Section II recalls the basic structure and properties of an autoencoder. Section III describes our unsupervised discriminative learning approach. Section IV presents experimental results. Finally, Section V draws conclusions.

II. Autoencoders

An autoencoder is an artificial neural network which can be trained to learn a suitable representation (coding) of the input data such to guarantee some desired properties. Fig.2 shows an example autoencoder in its simplest form, a feedforward non-recurrent net, with an input layer, an output layer and an hidden layer connecting them. Although many alternative structures are possible, a stable feature of autoencoders is that the input and output layers have the same size. This allows one to compare input and output, say $\mathbf{x}$ and $\widehat{\mathbf{x}}$, and compute a loss function based on their distance, so us to train the net to reproduce the input vector as faithfully as possible. Hence, since the autoencoder does not need labeled data for training, it implements unsupervised learning.

In more detail, the encoder maps the input vector, $\mathbf{x} \in \mathbb{R}^K$, to its hidden (or latent) representation, $\mathbf{h} \in \mathbb{R}^H$, as

$$\mathbf{h} = \phi_1 \left(\mathbf{W_1 x} + \mathbf{b_1} \right) \tag{1}$$

where $\phi_1(\cdot)$ is the activation function, $\mathbf{W_1}$ is a $K \times H$ weight matrix and $\mathbf{b_1}$ is a bias vector. The decoder, in turn, maps the latent representation to the output reconstruction as

$$\widehat{\mathbf{x}} = \phi_2 \left(\mathbf{W_2 h} + \mathbf{b_2} \right) \tag{2}$$

where $\phi_2, \mathbf{W_2}$ and $\mathbf{b_2}$ have obvious meaning.

Each training input $\mathbf{x}^{(i)}$ is hence mapped to its latent representation $\mathbf{h}^{(i)}$, and then reproduced as $\widehat{\mathbf{x}}^{(i)}$. The parameters of the autoencoder $\theta = \{\mathbf{W_1}, \mathbf{b_1}, \mathbf{W_2}, \mathbf{b_2}\}$ are learned by minimizing the average reconstruction error between input and output measured typically by the average Euclidean distance

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}^{(i)} - \widehat{\mathbf{x}}^{(i)}||^2 \tag{3}$$

The optimization is normally carried out through gradient descent. A typical use of the autoencoder is to provide a compact representation of the input. By adopting a bottleneck structure, $H < K$, the network is forced to represent the input with a smaller number of variables while preserving the information content as much as possible. If linear activation functions are used, the autoencoder approximates a principal component analysis (PCA), providing a low-dimensional linear representation of data. More often, nonlinear activation functions are used, such as rectified linear unit (ReLU), hyperbolic tangent, or a sigmoid function. In this case the autoencoder goes beyond PCA, capturing multi-modal aspects of the input distribution [14].

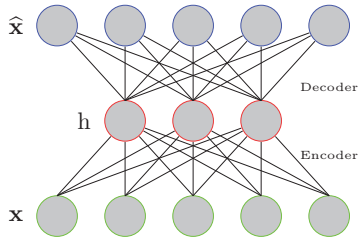


Fig. 2: Autoencoder with a single hidden layer.

Nonlinear autoencoders with more hidden units than inputs (called overcomplete) are also useful in applications [15]. This is the case of the denoising autoencoders, trained to recover clean versions of noisy input, and of the sparse autoencoders, trained with additional constraints to induce a sparse representation of the input. In the absence of additional constraints, an overcomplete autoencoder could be expected to learn just the identity function. However, this is not the case, and experimental results have shown that such autoencoders can still learn useful representations [16].

III. Proposed method

We cast splicing localization as an anomaly detection problem, where features extracted from the spliced region are regarded as anomalies. Autoencoders have been already used for anomaly detection in the literature typically solving a one-class classification problem [17]. The network is first trained off-line on the positive examples. Then, in the on-line phase, positive examples are reproduced with good accuracy, while anomalies give rise to large errors, allowing their reliable detection. Unfortunately, this one-class paradigm requires a training phase, something we rule off in pursue of a fully blind algorithm.

Recently, an autoencoder-based algorithm has been proposed [13] for fully unsupervised outlier removal in the context of image retrieval on the web. The key observation is that inliers (positive examples) obtained in response to a query image tend to share similar statistics, while outliers have a much more scattered distribution. Therefore, a bottleneck autoencoder trained on such examples will preferentially learn the dominant inlier distribution, and detect outliers. The performance is then boosted by re-training the autoencoder only on the positive examples and iterating until convergence. Note that this procedure is conceptually similar to adopting the EM algorithm for joint segmentation and classification, as already done in some papers for blind forgery localization [7], [8], [11].

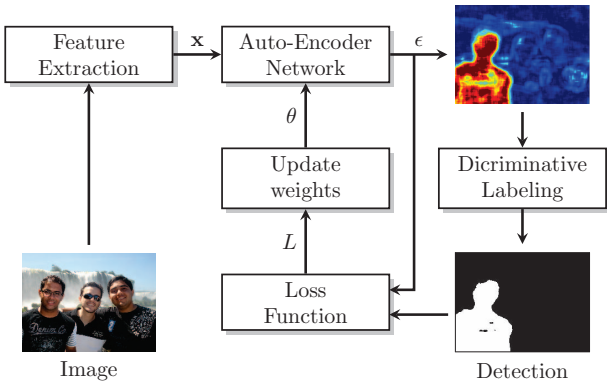


Fig. 3: Block diagram of the proposed algorithm.

We follow this approach for splicing localization, working on features extracted in sliding-window modality from the image under test. The block diagram of Fig.3 summarizes the procedure. In the following, we describe in more detail the main processing steps.

A. Feature extraction

We considered the features already used in [11], originally proposed for steganalysis in [12], which have shown very good performance in detecting different type of image manipulations [2], [5], [18]. Here we briefly recall their main processing steps referring to [12], [11] for further details. First, image residuals are computed through high-pass filtering (third order derivative), then they are quantized and truncated. The residual is then analyzed in sliding-window modality, taking patches of $w \times w$ pixels with stride s in both directions. Within each patch, co-occurrences of small patterns along the vertical and horizontal direction are computed as done in [11]. The histogram of co-occurrences is processed through a square-root non-linearity [11] and eventually normalized to zero-mean and unit norm to obtain a length-K feature. All these features, $X = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}\}$, represent the input to the algorithm.

With respect to [11] we modified some choices in order to better adapt to the new context. In particular, we kept separate counts for h/v filtering and h/v co-occurrences, relying on the net to exploit symmetries, and reduced the support for co-occurrence computation to 3 pixels to limit feature length. With these choices, features of 108 components are obtained.

B. Autoencoder

We use a simple feedforward autoencoder with a single hidden layer. Considering the limited number of features available for network training, we set $\mathbf{W}_1 = \mathbf{W}_2^T$ to reduce the number of free parameters. Several activation functions were tested, and finally we chose the hyperbolic tangent for the encoding phase (ϕ_1) and the the identity for the decoding one (ϕ_2). In the training phase, a 50% dropout was applied to neurons of the hidden layer to improve robustness.

Among the many design choices, the most important turned out to be the number of hidden neurons. Initially, we focused on bottleneck autoencoders ($H < K$) as in [13]. However, we observed a large number of false alarms, depending on the image and on the network initialization. By increasing the size of the hidden layer, including overcomplete autoencoders ($H > K$), better and more stable results were achieved.

One possible explanation is that the features associated with the splicing, unlike the image outliers in [13], are not scattered. Instead, they form a compact cluster that weighs significantly in the training of the autoencoder. A bottleneck autoencoder devotes part of its scarce resources to encode this "wrong" model, and cannot improve the representation of good features initially classified as splicing. Redundancy in the hidden layer helps improving the representation of wrongly classified genuine features, at which point the splicing features emerge as the real anomalies, allowing the network to converge towards a better solution. As of today, this is only a conjecture to justify our compelling experimental evidence and more studies are certainly necessary.

C. Iterative procedure

Initialization. The initial weights of the autoencoder are drawn from a uniform distribution in $[-1, +1]$, while the biases are set to 0 (remember that features have been normalized to zero mean and unit norm). The features extracted from the image are then autoencoded, and the reconstruction errors $\epsilon^{(i)} = ||\mathbf{x}^{(i)} - \widehat{\mathbf{x}}^{(i)}||^2$ are computed.

Discriminative Labeling. Features are labeled as pristine, $l^{(i)} = 0$, or splicing, $l^{(i)} = 1$, by comparing the associated reconstruction error with a threshold ϵ_T. This latter is chosen as in the Otsu algorithm [19] to minimize the sum of the intra-class variances of the reconstruction error

$$\sigma_w^2 = p_0 \cdot \sigma_0^2 + p_1 \cdot \sigma_1^2 \tag{4}$$

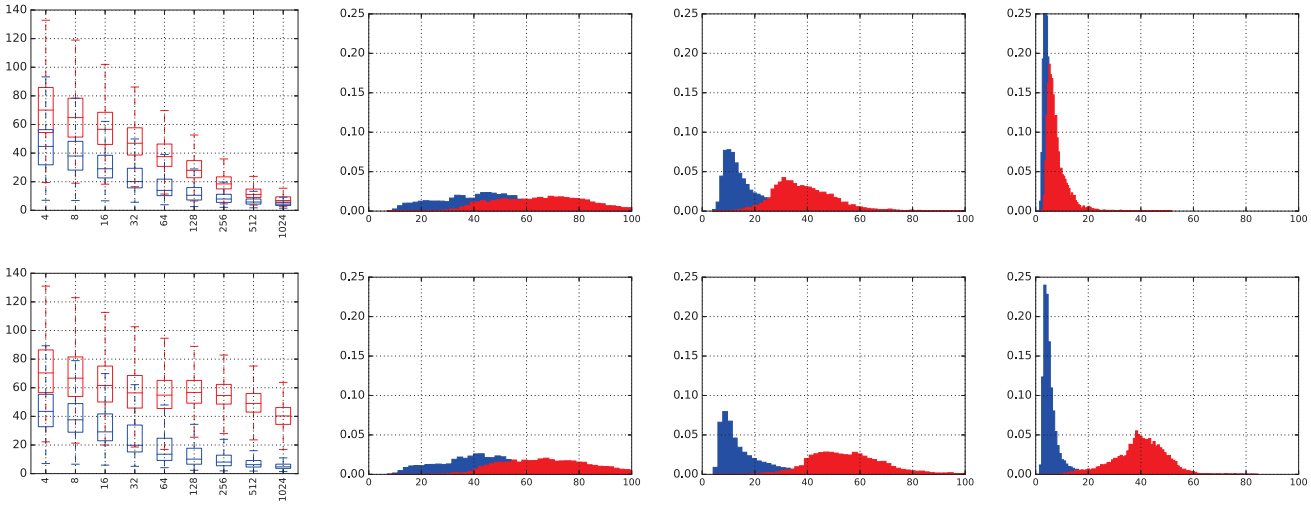
where p_l and σ_l^2 are the fraction of features labeled in class l, and the variance of their reconstruction error. The threshold is found by linear search.

Reconstruction Learning. The parameters of the autoencoder are now optimized by gradient descent. However, only features classified as pristine are used to this end, so as to reinforce learning for pristine class only, and increase the separation between errors of the two classes. Following again [13] the adopted loss function is

$$L(\theta) = \frac{1}{N_0} \sum_{i:l^{(i)}=0} \epsilon^{(i)} + \lambda \frac{\sigma_w^2}{\sigma_t^2} \tag{5}$$

with N_0 the cardinality of class 0 and σ_t^2 the variance of all reconstruction errors. The first term is the average reconstruction error for the pristine features, while the second term pushes towards a larger separation between the two classes as the parameter λ grows.

Stopping. The procedure ends when the final binary decision map is stable, with no more than 5 pixels switching label in the last 100 iterations.



(a) error vs no. iterations (b) 4 iterations (c) 64 iterations (d) 1024 iterations

Fig. 4: Different behaviour of the proposed method without (top) and with (bottom) discriminative learning. (a) boxplot of reconstruction errors vs. number of iterations. (b)-(d) detailed histograms after 4, 64, and 1024 iterations, respectively.

IV. Experimental results

Before turning to extensive experiments, it is instructive to study the behavior of the iterative procedure, with and without the discriminative learning phase, on a sample image. Based on some preliminary experiments we set the patch dimension to $w = 96$, the stride to $s = 8$, and the weight λ to 5. With reference to the image of Fig.1 (right), in Fig.4(b)-(d) we show the histograms of the reconstruction error for pristine and splicing features after 4, 64, and 1024 iterations. Without discriminative learning (top row) errors approach gradually zero for both classes as the autoencoder learns the mixture data distribution. Accordingly, the two histograms become closer and closer. This appears also clearly in the synthetic boxplot shown in part (a). On the contrary, when discriminative learning is included (bottom row), only pristine features are well reconstructed after 1024 iterations, and the histograms are well separated allowing reliable decisions. For the same image, Fig.5 shows the reconstruction error maps and the corresponding localization maps at 4, 16, 64, 256, 1024 iterations. Results are definitely good, although a small false alarm area remains, easily discarded by visual inspection.

For a more extensive analysis, we created a synthetic dataset of images of size 768×1024 pixels. The images are acquired from 7 devices, 6 smartphones (Huawei P7 mini, Nokia Lumia 925, LG D855, Samsung GT-I9505, Samsung GT-S7580, Apple iPhone 5s) and a camera (Samsung ES15). We consider about 25 host images for each device, and in each one insert a square splicing of size going from 96×96 to 192×192 pixels, taken from images of the other devices. Therefore, we have a total of 420 images, each one with a single splicing. All images are eventually JPEG compressed with quality factor 100.

A first experiment explores how performance depend on the hidden layer size. For each image we run the proposed algorithm, compute true positive (TPR) and false positive (FPR) rates, and average them over all images. Although the algorithm sets the detection threshold ϵ_T automatically for each image, we vary the actual threshold in the range $[0.1\epsilon_T, 3\epsilon_T]$ and report results in terms of ROC curves. Fig.6 shows the curves obtained for $H = 25, 50, 100, 300$ and 500. As already said, overcomplete autoencoders grant a competitive advantage over bottleneck autoencoders, and in the following experiments we use $H = 500$, the optimal value among those investigated. Note also that the automatic threshold (star point) given from the algorithm is very close to the threshold (circle point) that optimizes the F-measure.

Following, Fig.7 shows a comparative analysis of performance. As reference algorithms we consider [7] based on double JPEG compression, [10] based on noise level analysis, and [11] (Splicebuster) based on similar features as our own, but with explicit data modeling and EM optimization. First, we consider the case when the images have not been subject to any post-processing. All methods work pretty well, but the proposed method outperforms all others. In particular, with respect to Splicebuster, a smaller FPR is observed, indicating fewer false alarms. Then, we repeated the experiment after JPEG compressing all images with quality factor 90, or else resizing them with a scale factor 0.9. Although only a mild post-processing has been carried out, the performance of [10] degrades significantly, and [7] becomes basically useless. On the contrary both Splicebuster and the proposal keep working well, with a minor performance impairment. In Tab.I we report synthetic results in terms of average F-measure taking for each method the best threshold over all the dataset. Besides

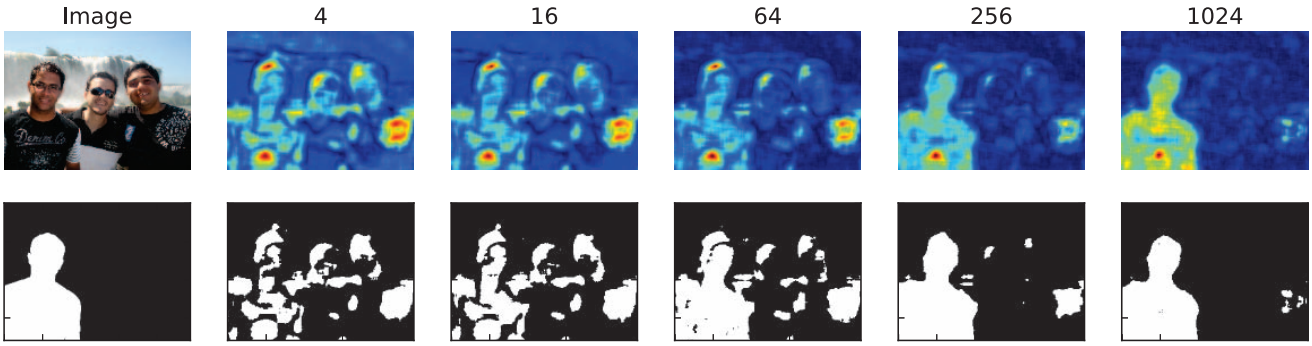


Fig. 5: Top row: the original image and the reconstruction error maps at iterations $4, 16, \cdots, 1024$. Bottom row: the ground truth and localization maps at iterations $4, 16, \cdots, 1024$.

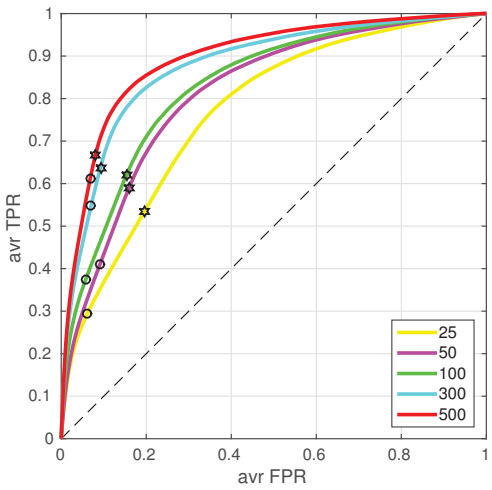


Fig. 6: Pixel-level ROCs on the synthetic database. Stars mark the points obtained with the Otsu threshold, while circles mark points with maximum F-measure.

confirming the better performance of the proposed method, the table shows that the Otsu threshold used in the proposal is near-optimal, with a minimal loss w.r.t. the optimal point.

In Fig.8 we present results on a more realistic dataset, DSO-1, composed of 100 forged images [9]. This is a subset of the IEEE Image Forensics Challenge database for which ground truths are available and is composed only of splicings. In this case the proposed method has results comparable with Splicebuster, exchanging a higher localization ability for a lower false alarms detection. Finally, in Fig.9 we show results applying all the methods on the forged image of Fig.1 (left).

V. Conclusion

We proposed a new method for blind image splicing localization. We regard features coming from the spliced area as anomalies, and iterate autoencoder-based modeling and discriminative labeling to tell them apart. Experimental results look promising, not only in ideal conditions, but also in the presence of post-processing. However, a deep investigation

Method	basic	compressed	rescaled
Bianchi and Piva [7]	0.306	0.067	0.068
Lyu et al. [10]	0.373	0.119	0.137
Cozzolino et al. [11]	0.283	0.228	0.266
Proposed (Otsu th. ϵ_T)	0.415	0.372	0.382
Proposed (optimal threshold)	0.418	0.378	0.389

TABLE I: Average F-Measure of proposed method and state-of-the-art references.

of the many degrees of freedom of the autoencoder structure is necessary, as well as on the possibility to extract features directly from the data by means of a suitable neural network. A final major topic is forgery detection. Here, we neglected this problem, assuming to work in a post-detection phase. These will be main directions for future research.

VI. Acknowledgement

This material is based on research sponsored by the Air Force Research Laboratory and the Defense Advanced Re search Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

References

[1] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, "Detection image splicing in the wild (web)," in *IEEE International Conference on Multimedia and Expo Workshops*, 2015, pp. 1–6.

[2] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques," in *IEEE International Conference on Image Processing*, 2014, pp. 5302–5306.

[3] W. Fan, K. Wang, and F. Cayre, "General-purpose image forensics using patch likelihood under image statistical models," in *IEEE International Workshop on Information Forensics and Security*, 2015, pp. 1–6.

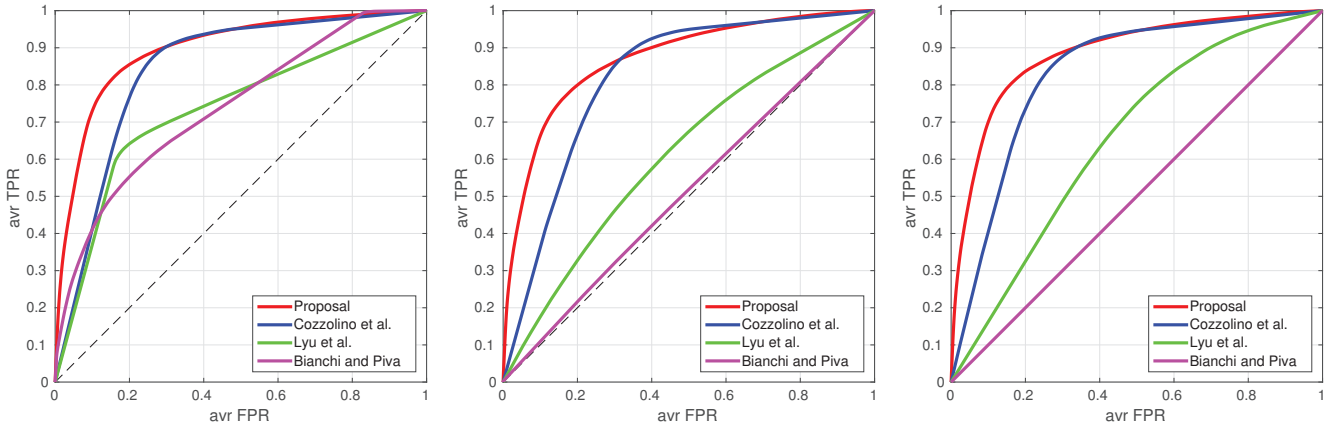


Fig. 7: Pixel-level ROCs on the synthetic database (left), on the processed database when all the images have been JPEG compressed with quality factor 90 (middle) and when the images have been resized with a scale factor 0.9 (right).

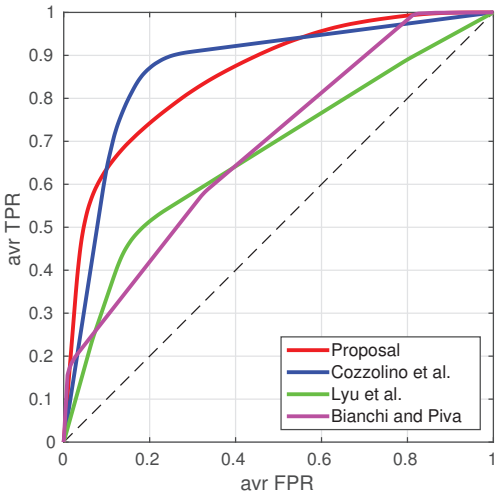


Fig. 8: Pixel-level ROCs on the DSO-1 Database.

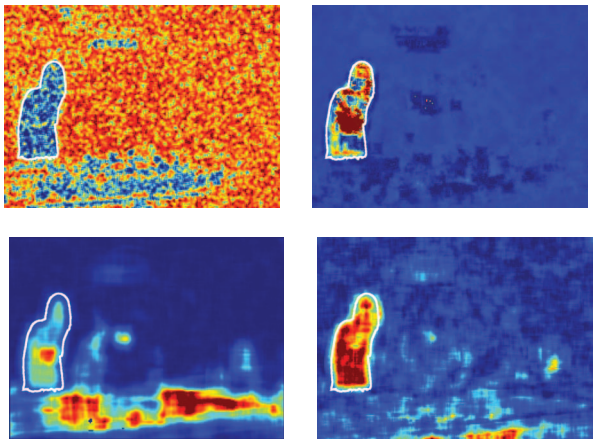


Fig. 9: First row: heat maps obtained for Bianchi and Piva [7] (left) and Lyu et al. [10] (right). Second row: heat maps for Cozzolino et al. [11] (left) and proposed method (right).

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

[5] L. Verdoliva, D. Cozzolino, and G. Poggi, "A feature-based approach for image tampering detection and localization," in *IEEE Workshop on Information Forensics and Security*, 2014, pp. 149–154.

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

[7] T. Bianchi and A. Piva, "Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.

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

[9] T. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1182–1194, july 2013.

[10] S. Lyu, X. Pan, and X. Zhang, "Exposing Region Splicing Forgeries with Blind Local Noise Estimation," *International Journal of Computer Vision*, vol. 110, no. 2, pp. 202–221, 2014.

[11] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *IEEE International Workshop on Information Forensics and Security*, 2015, pp. 1–6.

[12] 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.

[13] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *IEEE International Conference on Computer Vision*, 2015, pp. 1511–1519.

[14] N. Japkowicz, S. Hanson, and M. Gluck, "Nonlinear autoassociation is not equivalent to PCA," *Neural Comput.*, vol. 12, pp. 531–545, 2000.

[15] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[16] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[17] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *International Conference and Data Warehousing and Knowledge Discovery*, 2002.

[18] H. Li, W. Luo, X. Qiu, and J. Huang, "Identification of various image operations using residual-based features," *IEEE Transactions on Circuits and Systems for Video Technology, in press*, 2016.

[19] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Sys., Man., Cyber.*, vol. 9, pp. 62–66, 1979.