IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 10, 2020, accepted July 6, 2020, date of publication July 16, 2020, date of current version July 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009877

A Full-Image Full-Resolution End-to-End-Trainable CNN Framework for Image Forgery Detection

FRANCESCO MARRA[1], (Member, IEEE), DIEGO GRAGNANIELLO[1], (Member, IEEE), LUISA VERDOLIVA[2], (Senior Member, IEEE), AND GIOVANNI POGGI[1], (Member, IEEE)

[1]DIETI, Università degli Studi di Napoli Federico II, 80125 Napoli, Italy
[2]DII, Università degli Studi di Napoli Federico II, 80125 Napoli, Italy

Corresponding author: Francesco Marra (francesco.marra@unina.it)

This work was supported in part by the Google Faculty Research Award, in part by the Air Force Research Laboratory, in part by the Defense Advanced Research Projects Agency under Grant FA8750-16-2-0204, and in part by the PREMIER project, funded by the Italian Ministry of Education, University, and Research within the Progetti di Ricerca di Interesse Nazionale (PRIN) 2017 Program.

ABSTRACT Due to limited computational and memory resources, current deep learning models accept only rather small images in input, calling for preliminary image resizing. This is not a problem for high-level vision problems, where discriminative features are barely affected by resizing. On the contrary, in image forensics, resizing tends to destroy precious high-frequency details, impacting heavily on performance. One can avoid resizing by means of patch-wise processing, at the cost of renouncing whole-image analysis. In this work, we propose a CNN-based image forgery detection framework which makes decisions based on full-resolution information gathered from the whole image. Thanks to gradient checkpointing, the framework is trainable end-to-end with limited memory resources and weak (image-level) supervision, allowing for the joint optimization of all parameters. Experiments on widespread image forensics datasets prove the good performance of the proposed approach, which largely outperforms all baselines and all reference methods.

INDEX TERMS CNN, digital image forensics, forgery detection.

I. INTRODUCTION

In this work, we propose a new framework for image forgery detection based on convolutional neural networks (CNN). This may not look particularly exciting: deep learning is by-now common practice to solve all kinds of vision-related problems. However, image forensics has some peculiarities that set it apart from standard computer vision problems. We can summarize them in the need to look, at the same time, at the whole image but also at its tiniest details. Consider the example of Fig.1. This well-crafted splicing does not show obvious artifacts that allow detection by visual inspection, but a suitable textural analysis reveals differences that may be due only to the insertion of alien material in the host image. Indeed, many state-of-the-art forensic tools rely on the statistical analysis of local micro-patterns, observed at their native (full) resolution. However, *local* analysis alone are necessarily suboptimal. Clues emerging from the whole image, and at multiple scales, should be combined and processed jointly to make a reliable decision. Therefore, our goal is to design CNN-based forensic tools that meet the contrasting requirements of full-resolution and full-image training and analysis.

FIGURE 1. Example of carefully crafted splicing. Visual inspection does not allow detection, but pixel-level analysis performed with Noiseprint extraction technique [1] expose suspicious textural differences.

The associate editor coordinating the review of this manuscript and approving it for publication was Joewono Widjaja.

It should be realized that this problem is indeed peculiar of multimedia forensics. Typical CNN classifiers for computer vision problems rely on *macroscopic* features, which bear high-level semantic clues on the scene. For example, a face detector may look for the presence of specific facial features with suitable spatial relationships. Such large-scale information persists nicely after resizing the image. And in fact, target images of wildly different sizes are routinely resized to match the input CNN layer. Actually, resizing is even used on purpose, during training, to gain robustness to scale changes. In the context of image forensics, instead, resizing may destroy the very same information classifiers rely upon, the pixel-level *micro-patterns* that characterize different digital histories. By analyzing such patterns one can identify camera models, individual devices, or discover the traces of out-camera processing. A huge scientific literature testifies on the importance of such high-frequency features. Hence, image resizing and resampling should be definitely avoided when performing forensic tasks.

So, one could naively think of using a network with an input size as large as the target image. Besides the lack of generality (images can be of any size) a more fundamental issue concerns computational and memory resources. Acquisition devices are continuously improving their resolution, with commercial smart-phone cameras delivering photos with many millions of pixels. Deep learning hardware capabilities do not increase at the same rate. Due to computation and memory limitations, state-of-the-art architectures accept only small images in input, especially when very deep networks are used. Therefore, the highly informative image samples cannot be directly fed to a network and analyzed as a whole.

Eventually, when high-resolution must be preserved, a simple solution is to perform patch-wise feature extraction, followed by some forms of feature aggregation to exploit the full-image information. This approach makes full sense, and largely predates deep learning. Yet, even with good CNN-based feature extractors and classifiers, it is inherently suboptimal for several reasons: *i)* poor feature extraction; *ii)* poor global decision; *iii)* need of over-detailed ground truth.

First of all, since the patch-wise feature extractor is trained without taking into account full-image information, the best it can do is to learn good features for *local* decisions, which are not necessarily the best ones in view of future aggregation. Then, the global classifier, trained after freezing the patch-level processing, operates only on intermediate features, hence is necessarily suboptimal with respect to a classifier trained end-to-end on the original data. Last, patch-wise training requires a detailed, handcrafted, ground truth. Therefore, the large datasets necessary to train deep learning models require a huge man-power and are inevitably affected by errors, with a sure impact on the eventual performance.

All these considerations motivate our work, and allow us to define the final goal more clearly. We want to design deep learning models for image forgery detection which are:

1) full-image: make decisions based on information gathered from all over the image;
2) full-resolution: do not perform any harmful image resizing;
3) end-to-end trainable: optimize jointly all model parameters for image-level classification, based only on image-level (weak) supervision.

To achieve this goal, we propose a framework comprising three blocks in cascade performing, respectively, patch-wise feature extraction, image-wise feature aggregation, and global decision. By itself, this structure is not new. However, unlike in the current literature, where feature extractor and classifier are trained independently of one another, we train all blocks jointly, based on image-wise labels, allowing information to flow backward through the whole network. Therefore, the global decision takes into account features extracted from the whole image, whatever its size, and based on local micro-patterns. To perform end-to-end training with the full-image and full-resolution constraints, however, we must overcome the problem of insufficient memory resources. So we trade memory for computation in an advantageous way, by means of the gradient checkpointing strategy [2], solving the memory problems at the cost of a very limited increase of processing time. Eventually, the proposed framework allows one to optimize jointly the local information extraction, the global feature aggregation, and the whole-image classification, whatever the input image size.

We implemented several versions of this general framework, through appropriate selection of the major architectural blocks. After training on suitable synthetic datasets, we performed extensive experiments on realistic datasets widespread in the image forensics community, focusing on local manipulations, such as splicings, copy-moves, and inpainting, likely indicators of malicious attacks. Results fully support our approach which largely outperforms both baseline methods and state-of-the-art references, including methods requiring strong supervision.

In the following, we analyze related work (Section II), describe the proposed approach (Section III), report on the results of numerical experiments (Section IV), and finally draw conclusions (Section V).

II. RELATED WORK

Forgery detection is a central topic in image forensics, and there is a large bulk of relevant literature. In addition, it is necessary to consider both forgery detection and localization, since these tasks are tightly related. Indeed, detection methods can be used for localization through sliding-window analysis, and localization method may allow detection by suitable post-processing. So, to limit the scope, in the following analysis we take a historical perspective, but focus especially on recent CNN-based methods. Moreover, we neglect global manipulations, such as histogram equalization or gamma correction [3], [4], which are not necessarily related to a malicious forgeries, as well as methods devoted only to copy-move forgery detection [5]–[8].

Early contributions were mostly model-based, looking for statistical anomalies related to the color filter array (CFA) [9], [10], double JPEG compression [11], [12], or sensor noise [13], [14]. Most of these methods assume *a priori* the presence of a forgery and pursue localization through pixel-level analysis, generating a heat-map. Then, a global score can be easily computed from the latter and used for detection. Model-based approaches are elegant and do not require extensive training, but work only in quite restrictive hypotheses.

The advent of data-driven solutions granted a quantum leap in performance and ensured higher generality. Methods based on machine learning extract suitable hand-crafted features from the image, both in the spatial domain [15]–[19] and in the transform (DCT, wavelet) domain [20]–[22], which are used to train a classifier. Extracting features from the whole image allows direct and reliable image forgery detection. Instead, localization can be obtained by working in sliding-window modality and using a suitable local score. The most discriminative features rely on high-order image statistics which help revealing spatial inconsistencies originated by the presence of forgeries. To this end, high-pass residual images are often used, obtained by means of derivative filters [23] or image denoisers.

In recent years, methods based on deep learning have become dominant. Some early papers, inspired by the success of residual-based machine learning methods, propose CNN architectures with a first layer of high-pass filters, either fixed [24], [25], or trainable [26], meant to extract residual feature maps. In [27] it is even shown that successful methods based on hand-crafted features can be recast as CNNs and fine tuned for improved performance. In [28] these low-level features are augmented with high-level ones in a two-stream CNN architecture. Recent findings [29], [30], however, show that such constrained first layer is only useful with small networks and datasets. Given a suitably large training set, general-purpose very deep architectures provide the same good results in favourable cases, but ensure higher robustness to compression and training/test misalignments.

Several papers, to begin with [24], followed more recently by [31] and [4], train explicitly the net to distinguish between homogeneous and heterogeneous patches, the latter characterized by the presence of both pristine and forged areas. The rationale is to catch the patterns that characterize transitions regions, anomalous with respect to the background, so as to localize possible forgeries. This idea is followed also in [32], where an hybrid CNN-LSTM architecture is trained end-to-end to produce a binary mask for forgery localization. These methods, however, require detailed ground truth maps to train the net, which may not be available or precise.

For architectural constraints, most of these methods carry out a patch-based analysis, working on relatively small patches, with further steps needed to compute a global score at image-level. In [24], for example, the CNN extract features patch-wise and later aggregates them in a global feature vector used to feed a SVM classifier. This may impact on detection performance. A more fundamental limit concerns the need of strongly aligned training and test sets. Some methods, *e.g.*, [4], [32], carry out experiments on a single database split into training and test, others [28] require fine-tuning on target data. All this highlights the limited generalization ability of supervised learning, as also shown in [33].

A more promising line of research is to revisit the anomaly detection approach under a data-driven paradigm. Anomalies are detected by means of single-image analysis, with a sort of blind source identification. In [34] this was accomplished in a fully unsupervised fashion by using an autoencoder architecture. More recent proposals [1], [35], [36] use camera-model features, gathered off-line by dedicated CNNs, or leverage metadata information [37] for direct detection. A strong pro of this approach is that training is performed only on pristine images, with no need of aligned datasets and ground truths, which ensures good robustness and adaptability to unseen manipulations. In [1] and [37], in particular, this is achieved by using a Siamese training on pairs of patches extracted from pristine images, with a suitable consistency metric.

Besides its technical content, this short review of ideas makes clear that there is high and growing interest for new solutions in this field, to face the threats posed by increasingly sophisticated fake multimedia tools.

III. PROPOSED METHOD

Our aim is to design a deep network to detect the presence of localized forgeries in a target image, irrespective of the image size and the forgery size. Of course, images can have wildly different sizes, depending also on the context, but the trend is towards higher and higher resolutions. For Christmas 2019, Xiaomi released a 500$ smartphone featuring a 108-Mpixel camera. On the other hand, due to computation/memory bottlenecks, deep networks accept rather small images in input, for example 256×256-pixel. Hence, a strong size mismatch typically occurs between target image and network input. For most image analysis applications, this mismatch is not a big problem and two solutions can be considered:

1) images are rescaled to fit the network input, or
2) images are processed patch-wise, and results are fused off-line to make a global decision.

In the following paragraphs, we first explain why such solutions are not viable for image forgery detection, then describe the proposed architecture, and finally show how it can be trained end-to-end based on the gradient checkpointing method.

A. THE NEED FOR FULL-IMAGE FULL-RESOLUTION PROCESSING

The first solution listed before is to rescale the image to fit the network first layer. However, this is not advisable when dealing with forgery detection. In some cases, the forged

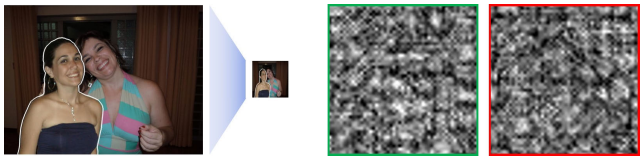

FIGURE 2. Strong image resizing corrupts the textural patterns used in forensics. Here the pixel-level analysis performed with Noiseprint [1] does not show anymore such strong differences.

region could be so small to become practically undetectable after strong downsampling. A more fundamental problem, however, is that some sophisticated forgeries may only be detected based on the statistical analysis of micro-textures. These precious high-frequency components are strongly corrupted when the image is resized or resampled. Fig.2 shows a clear example, in which the markedly different textures highlighted in Fig.1, after resizing become very similar to one another and basically useless for forensic analysis.

The second solution is to perform patch-level detection, with no resampling, followed by some form of information fusion to make a global decision. Indeed, given an ideal patch-level classifier, the fusion problem has an obvious solution, and the presence of a forgery can be declared if at least one forged patch is detected. However, real-world detectors are far from ideal, they always have non-zero missing-detection and false-alarm rates. For example, assuming a rather optimistic 1% patch-level false-alarm rate, and independent decisions, a 100-patch pristine image would present a false-alarm rate beyond 63%. Therefore, the fusion problem is not at all trivial with real-world detectors, as our experiments will confirm. In addition, the patch-level detector itself should be designed taking into account image-level performance.

These considerations motivate the need for a full-image full-resolution detector. In this way, precious microtextures can be preserved and, at the same time, information coming from all patches can be processed jointly to make a reliable decision. A naive implementation of this idea, with a CNN input size matching the image size, would require huge computational and memory resources, not to speak of the number of images needed for reliable training. Instead, we propose a suitable architecture that, through reasonable structural constraints, satisfies the needs of forensics detection with limited resources.

B. PROPOSED FRAMEWORK

The proposed framework is represented pictorially in Fig.3. It consists of three blocks performing, respectively, patch-level feature extraction, feature aggregation, and decision. These blocks are deliberately left unspecified at this time because their precise implementation is not the core of the proposal. In the following, we try several CNNs as feature extractors, as well as several forms of pooling and several classifiers, selecting eventually the architecture that performs best in our test. However, this is only for the purpose of carrying out experiments on real-world datasets and prove the potential of this approach. Better implementations will be certainly possible within the same general framework.

1) PATCH-LEVEL FEATURE EXTRACTION

After dividing the image in overlapping patches, these are processed to extract discriminative features. As feature extractors, we adopt some state-of-the-art deep networks, taking the output of the penultimate layer as feature vector, and discarding the final class probabilities. However, considering the peculiarities of image forgery detection, we modify the input layer to accommodate some additional inputs, the image noiseprint [36], besides the image color bands. Noiseprints are high-pass image residuals, extracted through a dedicated network, in which camera-related artifacts are emphasized. Therefore, they highlight possible spatial anomalies and may help detecting local manipulations.

2) FEATURE AGGREGATION

The feature extractor produces a large number of features, which are aggregated image-wise to obtain a single descriptor for the classification task. To this end, we consider several forms of pooling, maximum, minimum, average, and average of squares:

$$\begin{aligned} F_{max} &= \max_{i=1,..,N_p} F_i \\ F_{min} &= \min_{i=1,..,N_p} F_i \\ F_{mean} &= \frac{1}{N_p}\sum_{i=1}^{N_p} F_i \\ F_{msq} &= \frac{1}{N_p}\sum_{i=1}^{N_p} F_i^2 \end{aligned} \tag{1}$$

where $F_i = [F_{i,1}, \ldots, F_{i,C}]$ is the C-component feature extracted from the i-th patch, N_p is the number of (possibly overlapping) patches, and all operations on features are component-wise. The most appropriate type of pooling depends on the problem of interest. When the information is spread over the whole image, an average pooling is reasonable, while min or max pooling are more appropriate when the discriminative information is concentrated in a localized region. In any case, we also use the combination of multiple types of pooling, leaving the final choice to experiments. After aggregation all explicit spatial dependencies are discarded.

Note that the type of pooling impacts on how information back-propagates from the output to update the parameters of the feature extractor. In more detail, let F_{agg} denote the aggregated feature, $\mathcal{L}$ the loss function of the framework, and θ a generic parameter of the CNN. Then, the gradient of $\mathcal{L}$ with respect to θ reads

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{c=1}^{C} \frac{\partial \mathcal{L}}{\partial F_{agg,c}} \frac{\partial F_{agg,c}}{\partial \theta} \tag{2}$$

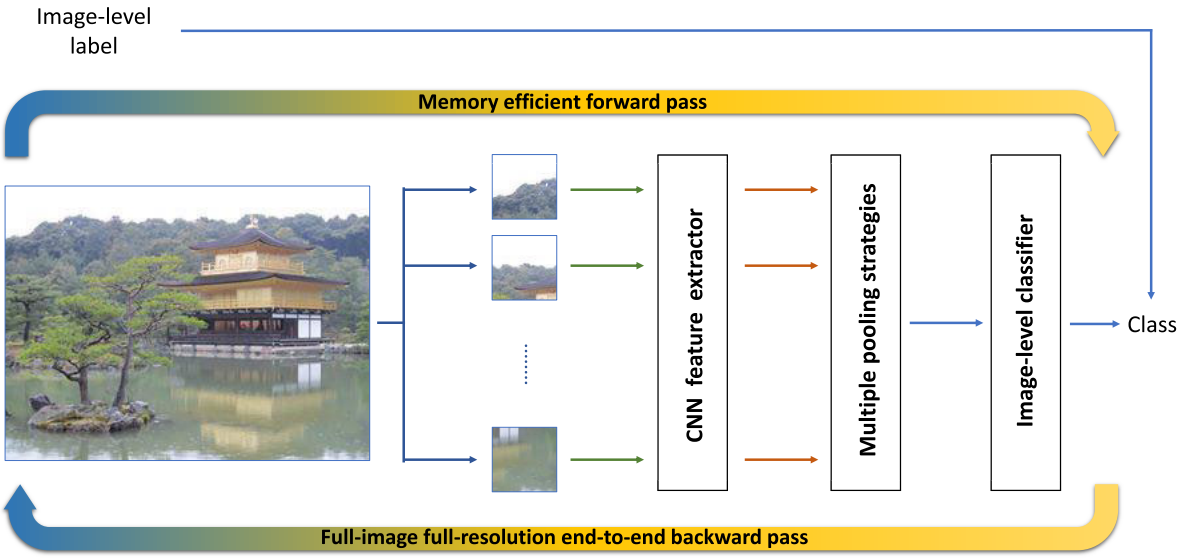

FIGURE 3. Proposed end-to-end-trainable framework for image forgery detection, comprising extraction, aggregation, and classification blocks. During the forward pass, a memory-efficient implementation allows us to process the entire full-resolution image as a whole. During the backward pass, all framework blocks are optimized jointly and the network learns how to extract and aggregate the most discriminant information towards the correct classification of the whole image.

with

$$\frac{\partial F_{agg,c}}{\partial \theta} = \begin{cases} \dfrac{\partial F_{i,c}}{\partial \theta} \cdot \delta_{i,i_{\max}(c)}, & \text{max pooling} \\ \dfrac{\partial F_{i,c}}{\partial \theta} \cdot \delta_{i,i_{\min}(c)}, & \text{min pooling} \\ \dfrac{1}{N_p} \displaystyle\sum_{i=1}^{N_p} \dfrac{\partial F_{i,c}}{\partial \theta}, & \text{average pooling} \\ \dfrac{1}{N_p} \displaystyle\sum_{i=1}^{N_p} 2F_{i,c} \dfrac{\partial F_{i,c}}{\partial \theta}, & \text{av.square pooling} \end{cases} \tag{3}$$

In the above equation, $\delta_{i,j}$ equals 1 when $i = j$ and 0 otherwise, while $i_{\max}(c)$ and $i_{\min}(c)$ point to the feature vectors with the largest, respectively smallest, c-th component. Therefore, with max or min pooling, only some "active" patches contribute to the gradient, and are updated during training. Instead, with average and average of square pooling all patches are involved. Of course, when multiple forms of pooling are used at the same time, the gradient is obtained as the weighted sum of the individual terms.

3) DECISION

After aggregating the local information in a single descriptor F for the whole image, this is classified by means of a few fully-connected layers. This is the typical classifier used in deep networks, and usually two layers provide a good trade-off between complexity and accuracy.

C. END-TO-END TRAINING

If we focus only on the post-training operations, the proposed architecture does not look much different from conventional approaches based on patch-wise feature extraction, pooling, and classification. In the literature, however, these blocks are trained independently of one another. The feature extractor is trained on a large number of labeled patches to minimize some patch-wise loss. Once the training is over, the network is frozen. Subsequently, it can be used to extract all the features of an image and generate an image level feature through pooling. Then, given a large number of image-level features and image-level labels, the final classifier is trained. On the contrary, our framework is trainable *end-to-end*. That is, we train the whole framework, top to bottom, on full-size images, with a single label associated with each one: forged or pristine. The loss back-propagates through the net up to individual patches, allowing the feature extractor to learn the most discriminative patterns for the final decision, and adapting the classifier jointly with the extractor itself.

To better underline the difference with respect to patch-wise CNN training, consider that in a large image with a localized forgery most patches are actually pristine, and only a few ones truly forged. In our end-to-end training, all these patches share the same image-level label (forged). Therefore, the net is forced to learn how to manage such contrasting indications to make the correct decision. As a side benefit, there is no need to have a pixel-wise ground truth for training, since the only relevant label applies to the whole image. Also, images of any size can be used for training, with forgeries of any size (especially if max/min pooling is used).

Going into technical details for each training batch of images, the framework performs *i)* an inner loop on the

patches of each image, computing the back-propagation at the end of the loop, and *ii)* an outer loop on the images of the batch, that sums up gradients computed for each inner loop and finally updates the weights once at the end of the batch. Due to the arbitrary size of input images, each inner loop involves a different number of patches, impacting on the computational effort, which may vary significantly from batch to batch. This is a minor issue, though, with respect to memory requirements. In fact, to back-propagate the loss, gradients must be computed for all processed patches, causing an increase of the occupied memory, which grows linearly with the image size. For deep networks and large images, this memory is simply unavailable.

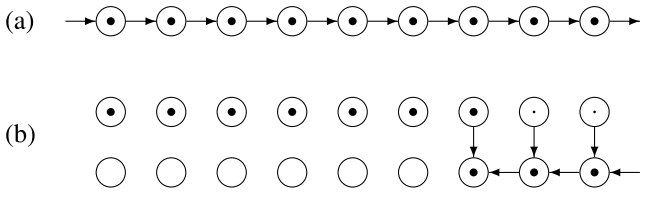

FIGURE 4. Conventional CNN training with backpropagation. During the forward pass (a) all the activations are stored (black dots). During the backpropagation (b), the activations are erased (small dots) as soon as they are used to back-propagate the gradients.

The situation is described pictorially in Fig.4, where a circle represents a layer, and a black dot at the center indicates that activations are stored. In the forward pass (a), in fact, all activations at each layer are computed and stored. Then, in the backward pass (b), they are used to propagate gradients from the last layer, where the loss is computed, to the input. After usage, they are erased (small dots). It should be realized that deep nets can include hundreds of layers, with several feature maps at each layer, whose size is typically proportional to the input size. Therefore, to process a large input image at once, a huge number of variables should be stored, exceeding the available memory.

To manage this problem we resort to the gradient checkpointing strategy, originally proposed in [2], which trades off memory for computation. This solution is described pictorially in Fig.5. During the forward pass (a), all activations are deleted immediately after use, except for those in a few ''checkpoint'' layers (red dots). In the backward pass (b)-(e), gradients are computed one group at a time (in the figure we show two groups of 4 layers). Since activations are necessary to this end, they are recomputed, but only from the last checkpoint on, (b). This allows backpropagating the gradient until the checkpoint layer itself (c). At this point all variables at layers beyond the checkpoint are deleted, and the process goes on with a new group of layers (d)-(e).

With a judicious choice of the number of checkpoints, memory occupation can be significantly reduced and become manageable. Of course, each activation is computed twice, but the computational overhead is limited, because the forward pass is lighter than the backward pass. Note that gradient checkpointing has been recently made available in PyTorch as

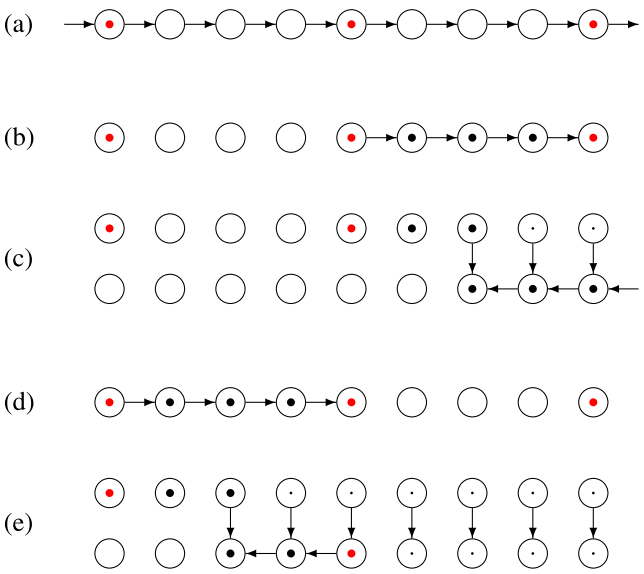

FIGURE 5. CNN training with gradient checkpoints. After the forward pass (a), activations are stored only at checkpoint layers (red). The backward pass (b)-(e) proceeds one group of layers at a time. Activations at intermediate layers must be recomputed each time a group is processed.

well as in other platforms. With this solution, we were able to train our network end-to-end seamlessly, with a increase of the training time that never exceeded 20%.

IV. EXPERIMENTAL ANALYSIS

Here, we design and perform numerical experiments to validate the proposed approach. In the following subsections, we first describe the training procedure, then present the results of some preliminary experiments carried out to make key design choices, and finally compare the proposed method with both baselines and state-of-the-art references on several challenging datasets widespread in the community.

A. TRAINING

In order to train our networks, we generated a suitable synthetic dataset. Background images are taken from the Vision dataset, proposed originally [38] for camera model identification, which comprises 7565 images acquired by 35 different devices with the native high-quality JPEG compression. To generate manipulated images, we spliced on them objects drawn from a set of 81 objects manually cropped from the uncompressed images of the UCID dataset [39]. Details on all datasets used in this work are reported in Tab.1.

We used all images from 25 devices of the Vision dataset for training, and kept the others for validation, with an approximate 70%-30% split, so as to avoid any possible bias. For each pristine image, we created on the fly a manipulated image by inserting in a random position one of the UCID objects, selected at random, with random scaling and rotation. Scaling is such that the size of spliced objects goes from about 1% to about 10% of the image size. Eventually, both pristine and manipulated images are flipped or rotated, and

TABLE 1. Features of datasets used for training and testing.

dataset	manipulations	counter forensic	# prist. / forged	resolution	format
Vision / UCID	automatic splicing	-	7565 / ∞	960×720 – 4640×3480	JPG
Dresden / FAU	automatic splicing	-	4992 / 14976	2560×1920 – 4352×3264	JPG
DSO-1	splicing	color/contrast adjustment	100 / 100	2048×1536	PNG
Korus	splicing, copy-move	-	220 / 220	1920×1080	TIF
NC2017	splicing, copy-move, computer-generated, inpainting	color/contrast adjustment, PRNU editing, JPEG quantization, cloning	2470 / 1051	436×600 – 3648×5472	PNG, BMP JPG
MFC2018	splicing, copy-move, computer-generated, inpainting	color/contrast adjustment, PRNU editing, JPEG quantization, cloning, noising, dithering, social network laundering	12246 / 1935	352×512 – 5470×7586	PNG, BMP JPG, TIF
MFC2019	splicing, copy-move, CG, inpainting, GAN, face manipulation	color/contrast adjustment, PRNU editing, JPEG quantization, cloning, noising, dithering, social network laundering	8646 / 5732	160×120 – 5320×7968	PNG, BMP JPG, TIF

FIGURE 6. Examples from the synthetic Vision/UCID training set. Spliced objects are delimited by a red contour for the sake of clarity.

JPEG compressed with QF going from 75 to 100, obtaining a significant augmentation. Fig.6 shows a few examples of manipulated Vision/UCID images (without rotations).

In the training procedure we used the Adam optimizer with minibatches of 10+10 images and a learning rate of 0.001. Training took about three days with an Nvidia Tesla P100 GPU. With the same hardware, testing takes about half a second for a 3072 × 4096-pixel image, including the noiseprint extraction, which decreases to 0.01 seconds if the image tiles are already stored in the GPU memory. The trained net is available online at https://github.com/FrancescoMarra/E2E-ForgeryDetection.

B. PRELIMINARY EXPERIMENTS

The proposed framework aims at the detection of localized manipulations, such as splicing, copy-move, and object removal through content-aware inpainting. Towards this goal, we instantiated the proposed framework by means of some key design choices. In particular, we

- augmented input RGB bands with the corresponding noiseprint bands;
- used Xception [40] as feature extractor;
- performed aggregation by including all types of pooling;
- used two fully connected layers, of size FC1=512 and FC2=256, to perform the final classification.

We arrived at these choices as a result of a large number of preliminary experiments, whose description would be dispersive and tedious. However, we can study experimentally the impact of each individual choice on the performance of the proposed architecture. To this end, we generated a new dataset, with the same modalities used for the training set, but completely separated from it. Background images were taken from the Dresden dataset, originally proposed [41] for camera model identification, and manipulated images were created by splicing on them 13 objects taken from the FAU dataset [5] (see again Tab.1 for details on datasets). After performing the splicing, images were JPEG compressed at high (QF≥95), medium (90≥QF≥85), or low quality (80≥QF≥75), and eventually resized at scale=0.75 or left unchanged. Spliced objects can be classified as large, medium, or small, depending on the largest dimension of their bounding box (after image resizing), set to 1024, 384, or 128, respectively. Note that, to carry out the large number of tests required by this analysis, we use a small training set, here, and results indicate main trends but can be improved by a more accurate training.

To assess performance, here and in all subsequent experiments, we classify the whole test set, compute false positive rate (FPR) and true positive rate (TPR) as a function of the detection threshold, going from 0 to 1, and obtain the corresponding receiver operating characteristic (ROC) curve. Eventually, we compute the area under the ROC curve (AUC) as a synthetic measure of performance.

In Tab.2 we report the results of our ablation study. The second row refers to the selected architecture, which uses Xception, takes in input both RGB and noiseprint bands, concatenates vectors given by all pooling types, and uses a size-512 FC1 layer. In all other rows, we modified a single

TABLE 2. Results of ablation studies on the Dresden/FAU dataset.

architecture			AUC
RGB+NP / Xception / all poolings / 512			0.851
RGB	→	RGB+NP	0.845
NP	→	RGB+NP	0.849
Resnet101	→	Xception	0.750
Inception	→	Xception	0.745
max-pooling	→	all poolings	0.800
avg-pooling	→	all poolings	0.808
FC1-size 256	→	512	0.831
FC1-size 1024	→	512	0.838

item of this reference architecture. A number of non-trivial results appear. First of all, Xception is a much better feature extractor than the two alternatives, Resnet101 [42] and InceptionV4 [43]. We had already observed a similar edge in other applications [30] although never so sharp. The likely reason is Xception's better use of resources, with a much smaller number of parameters to optimize for a given network depth. It also clearly emerges that using 4 types of pooling together ensures a significant improvement w.r.t. using only one of them. Using only max-pooling, as suggested by the nature of the problem, is even worse than using average pooling, probably because of its lower robustness to noise. As for the size of the first FC layer, 512 appears to be the best choice, although just slightly. The only controversial choice concerns the input. In fact, using only the RGB bands or only the noiseprint (NP) bands provides results very close to those of RGB+NP, with a statistically insignificant gap. Therefore, we refrain from sharp decisions on the input, and will keep testing several options in real-world cases.

TABLE 3. Results on subsets from the Dresden/FAU dataset.

sub-dataset	AUC
global	0.851
large-size objects	0.855
medium-size objects	0.860
small-size objects	0.875
high-QF JPEG compression	0.886
medium-QF JPEG compression	0.847
low-QF JPEG compression	0.855
original-size	0.884
resized	0.841

We now study the impact of compression, resizing, and splicing size on the performance of the proposed method by collecting results for specific relevant subsets. A quick look at the numbers of Tab.3 makes clear that only minor variations occur across such subsets, with all AUC's in the 0.84–0.89 range. The largest performance gap is observed between original-size and resized images. Also JPEG compression affects somewhat the detection performance, although no significant difference emerges between the medium-QF and low-QF cases. The size of the spliced area, instead, seems to have a minor impact and, contrary to expectation, relatively small-size splicings are detected more easily that large-size ones. Note that, on the average, the AUC on specific subsets is larger than the global AUC, but this is a consequence of the higher homogeneity of the tested images.

C. COMPARATIVE PERFORMANCE ANALYSIS

Having justified our design choices, we now move to compare the performance of the proposed framework with those of suitable baselines and state-of-the-art methods, using not only our relatively simple Dresden/FAU synthetic dataset, but also several realistic and challenging datasets widespread in the forensic community.

1) REFERENCE METHODS

First of all, we consider three natural baselines, all relying on Xception, given its good performance. The first one, Xception-resize, consists simply in resizing the target image to fit the CNN input, with straightforward training procedure. Xception-patchwise, instead, works by analyzing the image patch-by-patch, with no resizing and some spatial overlapping, and finally fusing scores. Accordingly, the net is trained to perform binary patch classification. Eventually, the output probabilities are collected in a heatmap, from which a suitable statistic is extracted (after some tests, we chose the max statistic) and compared with a threshold to make the image-level decision. Xception-pooling, instead, performs patch-level feature extraction, image-level pooling, and classification exactly like the proposed method with RGB input. However, CNN and classifier are trained independently of one another. Therefore, it provides direct insight into the competitive advantage of end-to-end over independent training. Both Xception-patchwise and Xception-pooling need labeled patches for training. Since the detectors look for anomalies, we decided to label as forged only boundary patches, that is, patches including a significant fraction of both background and manipulated areas.

Just like our baselines, methods proposed in the literature can be grouped in two classes. A few ones work at image level, like Xception-resize, while the majority, like Xception-patchwise/pooling, work at patch-level, as they pursue forgery localization, and are converted into image-level detectors through some simple post-processing.

For the first category, we selected the SPAM+SVM method [16], winner of the First IEEE Forensic Challenge and based on the SPAM steganalytic features [23], the CNN+SVM method of [24], which extract features through a constrained CNN, LSTM-EnDec [32], which uses a long-short term memory recurrent neural network to detect pristine/forged spatial transitions, and MantraNet [44], which performs joint image-level detection and pixel-level localization of forgeries, regarded as local image anomalies. For the second category, we consider several forgery localization methods converted into image-level detectors. In particular, we selected the best performing methods resulting from the analysis carried out in [1], that is, CFA [10], which

TABLE 4. Results of all versions of E2E and all references methods on the test datasets. No fine-tuning.

Method	supervision	Dresden/FAU	DSO-1	Korus	NC2017	MFC2018	MFC2019	average
Xception-resize	weak	0.609	0.539	0.527	0.513	0.570	0.516	0.546
Xception-patchwise	strong	0.721	0.643	0.533	0.729	0.711	0.632	0.661
Xception-pooling	strong	0.839	0.702	0.561	0.751	0.635	0.633	0.687
SPAM+SVM [16]	weak	0.506	0.768	0.502	0.767	0.631	0.634	0.635
CNN+SVM [24]	strong	0.593	0.728	0.568	0.798	0.702	0.679	0.678
LSTM-EnDec [32]	strong	0.543	0.590	0.521	0.504	0.535	0.542	0.539
ManTraNet [44]	strong	n/a	0.874	0.555	*0.612	*0.758	*0.580	0.676
CFA [10]	–	0.507	0.584	0.598	0.593	0.539	0.526	0.558
DCT [11]	–	0.505	0.614	0.501	0.683	0.523	0.509	0.556
NOI [9]	–	0.558	0.543	0.507	0.678	0.523	0.726	0.589
NoisePrint [1]	–	0.611	0.821	0.583	0.746	0.684	0.662	0.684
EXIF-SC [37]	–	0.599	0.721	0.496	0.709	0.670	0.655	0.642
E2E-RGB	weak	0.958	0.596	0.607	0.774	0.760	0.737	0.739
E2E-NP	weak	0.874	0.924	0.665	0.766	0.776	0.741	0.791
E2E-RGB+NP	weak	0.914	0.790	0.619	0.762	0.765	0.765	0.769
E2E-Fusion	weak	0.993	0.824	0.655	0.846	0.838	0.787	0.824

ManTraNet results marked with an asterisk are obtained on approximately 20% of the dataset (small images).

exploits features related to the color-filter array, DCT [11], based on the analysis of double-quantized DCT coefficients, NOI [9], looking for spatial inconsistencies in the noise level, EXIF-SC [37], looking for anomalies in the image leveraging the EXIF metadata during the training phase, and Noiseprint [1], which extracts and analyzes an image fingerprint where camera model-related artifacts are emphasized. All these methods compute a heatmap representing the probability that a certain patch has been manipulated. To make the image-level decision we extract several statistics from such heatmaps: mean, maximum, and q-quantile, with $q \in \{5, 10, \ldots, 95\}$, selecting the best one in terms of AUC performance separately for each method. Note that all these latter methods are blind, that is, they require no training on forged images or patches.

2) DATASETS

For performance assessment, besides our synthetic Dresden/FAU dataset, we consider several more datasets, widely used in the forensics community, with markedly different characteristics. DSO-1 [45] features only splicings, with little or no post-processing. In Korus [46], instead, both splicings and copy-moves are present. Both datasets include only large-size high-quality images, not even compressed in the case of Korus. A very different, and much more challenging, scenario is given by the NC2017, MFC2018, and the very recent MFC2019 datasets [47], developed by NIST[1] in the context of the Medifor initiative. Images of these datasets have been manually doctored, often with multiple and possibly overlapping manipulations of various types. In addition, they have wildly different sizes and quality levels, and have been subject to several anti-forensics measures to prevent easy detection and localization of forgeries. For our tests, we kept all images with splicing, copy-move, inpainting, or computer-generated material. The reader is referred to Tab.1 and to the original papers for more details, while some example images are shown in Fig.7.

[1]https://www.nist.gov/itl/iad/mig/media-forensics-challenge-2019-0

FIGURE 7. Examples from the NIST datasets.

3) NUMERICAL RESULTS

In Tab.4 we report the detection AUC for all reference and proposed methods on all test datasets. Next to each method, in column 2, we give the level of supervision it requires, strong (pixel-wise ground truth), weak (only image label), or – (none) for blind methods. In the upper part of the table we group all reference methods, including our three baselines, and in the lower part all versions of the proposed method with end-to-end (E2E) training. Best results are highlighted in red for reference methods and in blue for our proposal. In Fig.8 we also show ROC curves for a subset of methods (for readability) and datasets (for space)

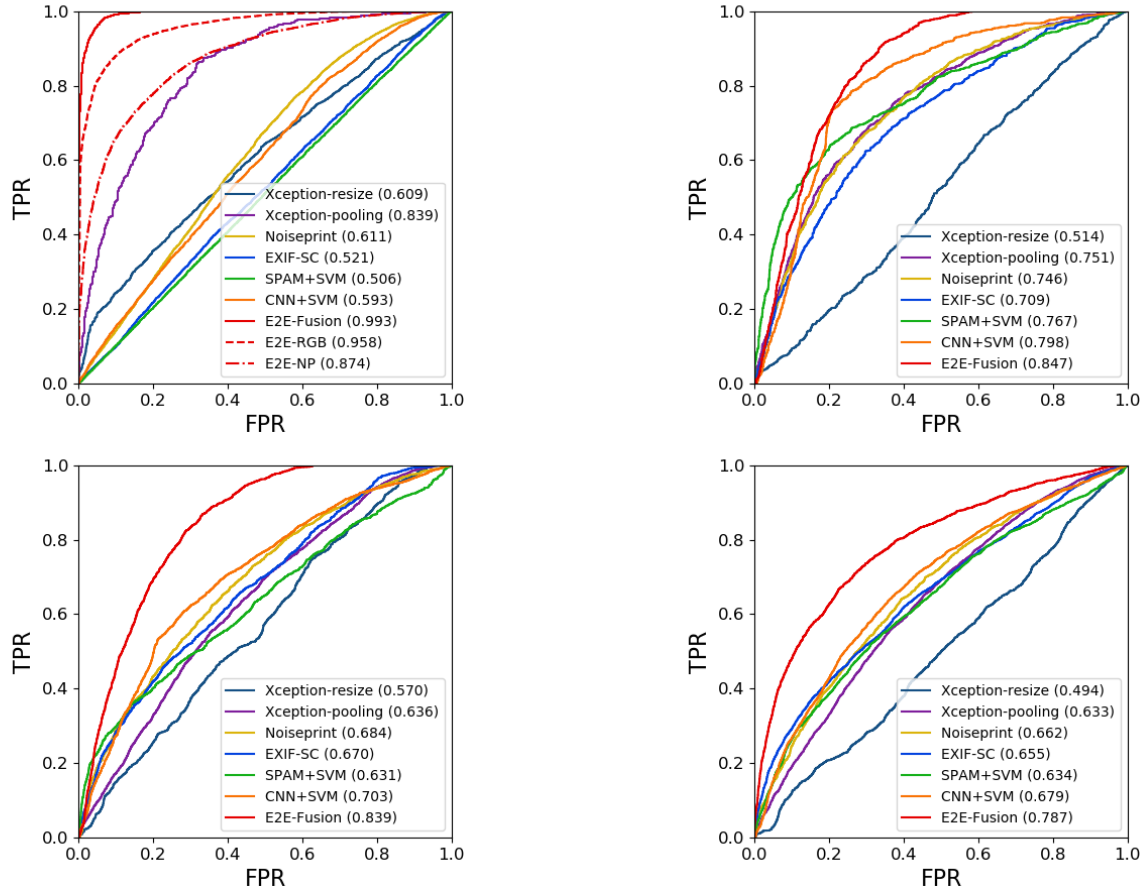

FIGURE 8. ROC curves on Dresden/FAU (top-left), NC2017, MFC2018, MFC2019 (bottom-right) datasets. For the sake of clarity, ROCs are shown only for selected methods: the best proposed (E2E-Fusion), two baselines, and the best references (SPAM+SVM, CNN+SVM, Noiseprint, EXIF-SC). Only for Dresden/FAU we also show other E2E versions. E2E-Fusion is always clearly, and almost uniformly, the best. The resizing-based baseline always the worst.

characterized by very different features. On the Dresden/FAU dataset, disjoint from the training Vision/UCID dataset, but well-aligned with it, the proposed method (E2E-RGB+NP) largely outperforms all references, with a gain of almost 10 percent points over the best one, the strongly supervised Xception-pooling. On this dataset, comprising pretty large images, ManTraNet does not run, as it requires memory exceeding the GPU capacity. For the same reason, it can process only about 20% of the images in the NIST datasets. Therefore its performance should be taken with care. Guided by the outcomes of preliminary experiments, together with the "best" version, with RGB+NP input, we consider also the versions with only RGB and only NP inputs. To our surprise, E2E-RGB provides a further significant performance improvement. Our explanation for this phenomenon is the strong heterogeneity of the input: since RGB bands and noiseprints have quite different statistics, the net may have a hard time processing them jointly. To confirm such hypothesis, we considered a further versions of the proposed method, where the networks trained on RGB-only, NP-only, and RGB+NP inputs are fused afterwards by a trivial average of scores. This strategy proved successful, with the new version, E2E-Fusion, providing almost perfect detection (see also the top-left ROC in Fig.8), thus confirming our conjecture.

Moving to the DSO-1 dataset, we observe again a significant gain, more than 5 percent points, of the best E2E method over the best reference. On this dataset, Noiseprint provides an especially good performance, a phenomenon already observed in [1], and likely related to all images being JPEG compressed at high-quality. Accordingly, also E2E works best with only noiseprints as input, with no fusion. Images of the Korus dataset, instead, are uncompressed. This removes a major source of forensic traces, which impacts all methods, some of which exhibit a 0.5 AUC, equivalent to coin tossing. CFA (relying on color filter array properties) and Noiseprint, keep providing decent results, however they trail all E2E versions, featuring AUC's between 0.60 and 0.66. It is worth underlining that the poor results observed in some cases are also a consequence of our experimental setting. In fact, all data-driven methods, including all versions of E2E, are trained on the Vision/UCID dataset and then tested,

with no fine-tuning, on other datasets completely unrelated with it. In the literature, aligned training and test sets are often considered, with a consequent boost in performance.

Turning to the more challenging NIST datasets, the general behavior does not change, with E2E working generally better than reference methods. The best reference method is not always the same for all such datasets: CNN+SVM for NC2017, ManTraNet for MFC2018 (only on small images), NOI for MFC2019. On the contrary, E2E-Fusion is always the best version of proposed method, and the best overall, with a significant performance gain over the best reference, going from 0.048 (NC2017) to 0.080 (MFC2018).

The final column shows the average over all datasets, which confirms all above observations. We only underline that the Xception-patchwise and Xception-pooling baselines are among the best references, although they require strong supervision, while Xception-resize, as expected, performs quite poorly, not far from coin tossing. This is very likely due to the loss of precious high-frequency details induced by resizing. However, to gather more objective data on this point, we carry out a further experiment by letting the scale of resizing vary on a wide range in a controlled way. Since Xception-resize cannot work on large images without an appropriate (strong) resizing, we consider instead the SPAM+SVM reference, which is a well-established method and can work on images of any dimension. Tab.5 provides the results for the DSO-1 dataset. It clearly appears that even a moderate resizing (scale=0.9) causes a sever performance loss, with AUC going from 0.768 to 0.669, and stronger resizing lead to decisions that are basically random. It is worth emphasizing that even a three-fold reduction may not be enough to perform image-level analysis through standard CNNs.

TABLE 5. Results of SPAM+SVC on DSO-1 vs. resizing scale.

resizing scale	1.000	0.950	0.900	0.700	0.500	0.333
AUC	0.768	0.737	0.669	0.595	0.460	0.478

Going back to the results of Tab.4, a general observation is that the performance of E2E is consistently good in all cases (with a small dip on Korus), including the NIST datasets, despite their great variety and the abundance of counter-forensic measures. This is all the more remarkable, considering that the network was trained on a dataset, Vision/UCID, lacking such a diversity. Therefore, we carried out a further experiment on NC2017 and MFC2018, in which the E2E methods are fine-tuned on their respective development sets, provided by NIST together with the test sets. Detailed results in terms of ROC curves are reported in Fig.9. It is clear that fine-tuning on the development set, certainly more aligned with the test set than Vision/UCID, grants further performance gains. In both cases, all curves show a large improvement with fine-tuning, with the E2E-Fusion AUC growing from 0.846 to 0.932 on NC2017, and from 0.838 to 0.902 on MFC2018. The larger improvement on NC2017 can be attributed to better development-test alignment and lighter

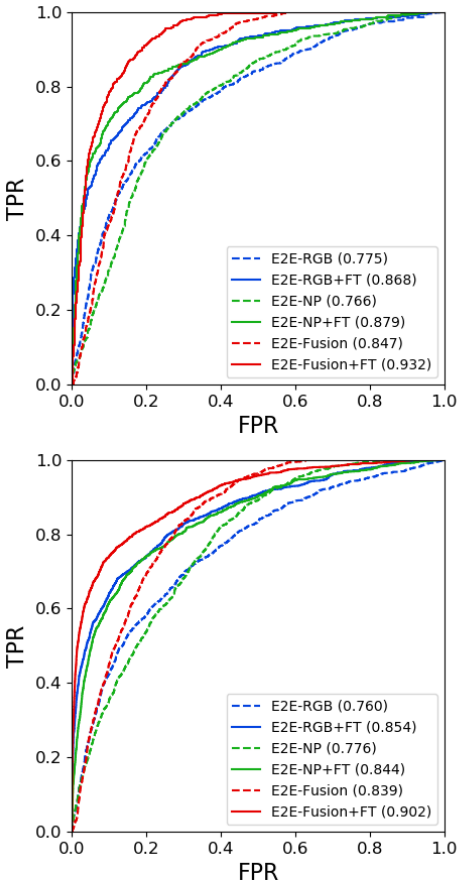

FIGURE 9. ROC curves of all E2E variants on NC2017 (top) and MFC2018 (bottom) without (dashed lines) and with (solid) fine-tuning on the NIST development sets. Fine-tuning provides a significant gain in all cases.

TABLE 6. Analytic results on NC2017 per type of manipulation.

Method	splicing	CM	inpaint.	CG	average
CNN+SVM	0.728	0.769	0.822	0.826	0.798
NoisePrint	0.692	0.722	0.776	0.786	0.746
E2E-RGB	0.829	0.819	0.694	0.949	0.774
E2E-NP	0.774	0.752	0.762	0.902	0.765
E2E-RGB+NP	0.816	0.832	0.693	0.921	0.762
E2E-Fusion	0.860	0.870	0.809	0.932	0.846

TABLE 7. Analytic results on MFC2018 per type of manipulation.

Method	splicing	CM	inpaint.	CG	average
CNN+SVM	0.750	0.685	0.672	0.731	0.702
NoisePrint	0.713	0.644	0.643	0.717	0.684
E2E-RGB	0.808	0.705	0.696	0.730	0.760
E2E-NP	0.805	0.750	0.744	0.817	0.775
E2E-RGB+NP	0.795	0.733	0.734	0.786	0.765
E2E-Fusion	0.860	0.811	0.811	0.799	0.838

counter-forensic actions. In any case, results are extremely satisfactory for such challenging datasets.

Taking advantage of the auxiliary information provided with these datasets, in Tab.6 and Tab.7 we provide also

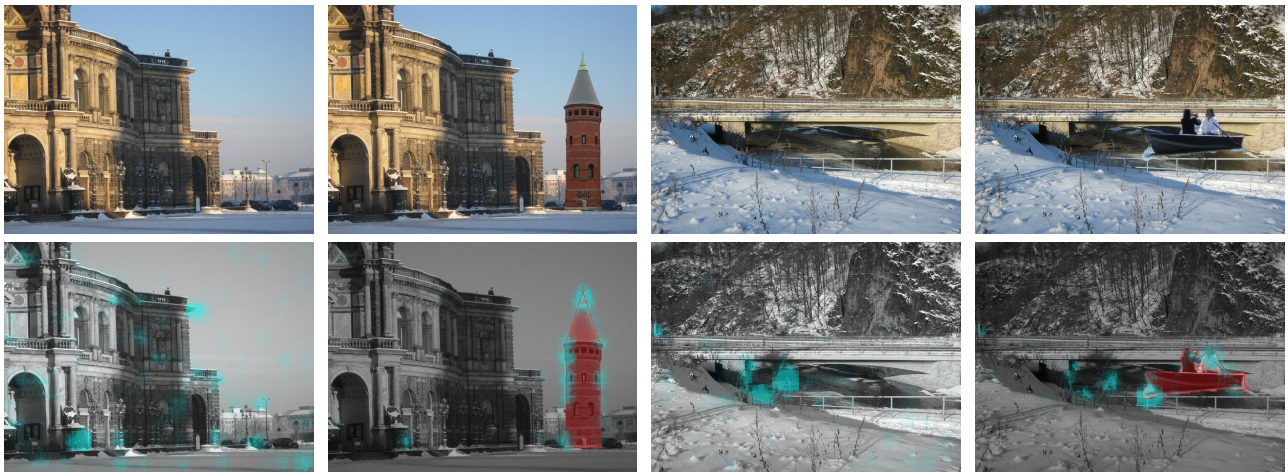

FIGURE 10. Example images (top) and activation maps (bottom) from the Dresden/FAU dataset. Pristine images are on the odd columns, forged images (with hand-made splicings for higher visibility) on the even columns. Active patches are superimposed in cyan to the gray-scale/red-scale version of the images.

analytic results for each type of forgery. Together with all versions of E2E (no fine-tuning), we also include some strong references, CNN+SVM and NoisePrint. Even though E2E is trained only on splicing, it works well also on the other manipulations and, with the exception of inpainting on NC2017, much better than the reference methods. Results are also rather stable across the two datasets, except for a sharp performance drop on computer-generated fakes going from NC2017 to MFC2018. This is probably the effect of the fast pace of progress in the quality of such manipulations.

D. MODEL INTERPRETABILITY

The E2E framework was conceived and trained with the goal of making global decisions. During training, with no information on forgery location, like ground truth masks, the model learns automatically to single out the image details that most contribute to decide on the nature of the whole image, forged or pristine. In the following subsections we provide some insight into how the system exploits and combines local information coming from all over the image, giving an interpretation of the global decision making process.

1) ACTIVATION MAPS

First of all, we try to investigate the impact of each patch of the image on the final decision. To this end, we consider a simplified framework in which only the max pooling is used. Given this hard selection rule, we can easily compute a spatial activation map which counts how many features each patch contributes to the overall feature vector. Such a map, however, would be extremely coarse, due the low resolution of patch-wise analysis. Therefore, we combine it with the a high-resolution map, the Grad-CAM (guided gradient weighted class activation map) obtained by backpropagating the loss gradient to the full-resolution input [48]. In Fig.10 we show some results for images of the Dresden/FAU dataset (hand-made to look more realistic). For this synthetic dataset, we have the pristine version of each manipulated image, so we can analyze the network behavior in both circumstances. In all cases, the network focuses on high-activity regions, often corresponding to object boundaries. When there is no manipulation, the salient regions are scattered all over the image. On the contrary, when a splicing takes place, they tend to concentrate on the boundaries of the spliced object, proving that the system has learned to look at these patches to make its decisions. Therefore, when a forged image is detected, this activation provides hints about the possible site of the manipulation.

2) ROI-BASED ANALYSIS

Moving towards forgery localization, we can obtain some interesting results by leveraging the flexibility of the proposed framework. Indeed, since the system can analyze images of any size, it can also analyze regions of interest (ROI) selected by the user based on the previous activation map or any other criterion. If the ROI contains manipulated material, the system will likely provide a large probability of manipulation (score, from now on). Therefore, the system can be used in supervised modality to test suspicious objects. Also, it can be recast to perform automatic box-like localization. In fact, once features have been computed and stored for all patches, the aggregation and classification phases are extremely simple, with light-speed processing. Therefore, one can easily test a large number of boxes and select automatically as ROI those with the largest scores, obtaining a rough but effective form of localization.

Fig.11 shows some examples taken from the MFC2018 dataset. Together with the original images (top) and activations maps (middle) it also shows (bottom) the scores obtained over the whole image (white number in the top-left corner) and on selected boxes (colored numbers). The green boxes have been selected manually around possible subjects of interest, while the magenta boxes are selected

FIGURE 11. Manipulated images from the NIST datasets (top) corresponding activation maps (middle) and ROI-based localization results (bottom) with hand-made (green) and automatic (magenta) box-shaped ROIs. Detection scores are shown on the top-left of each box.

by our automatic procedure around the local maxima of the score. In the first image, the man on the right has been spliced on the host background. Here, the activation map provides strong hints on the possible manipulation, confirmed by a large image-level score (0.935). However, an even larger score (1.000) is obtained when a ROI is correctly placed around the splicing. The automatic procedure also selects a ROI roughly covering the splicing, with unitary score. Another ROI is selected automatically in a pristine area in correspondence of a local maximum, but is has a rather low score (0.428). In the second image, a further splicing has been added, the woman in the center. Neither the activation map nor the automatic ROI selection procedure highlight this new subject. So, we selected a ROI manually around this splicing, obtaining a rather low score. Exploiting the side information provided with the NIST datasets, we investigated on this splicing, to discover that the inserted object had been acquired with the same camera model as the host image. This fact reduces the discriminating power of the noiseprint input, justifying in hindsight such result. In the third image, the only manipulation is a tiny inpainted region. Here, a supervised selection makes no sense, since the manipulated region does not correspond to any semantic object. However, the manipulation is nicely localized through the automatic procedure, with unitary score, unlike other candidate ROIs characterized by low scores. The last image shows an opposite case, with many large, semantically relevant, objects spliced on the host image. To avoid cluttering the image, we now show only the supervised ROIs and the corresponding scores, which are very large in all cases.

FIGURE 12. Examples of missed detection from the NIST datasets.

To complete this visual inspection of results, it is fair to show, in Fig.12, some counter-examples where the proposed framework fails to detect the manipulation. Reasons for failure are not always obvious. In these cases, they may be related to the absence of texture in the spliced object (right) or the strongly textured host image (right) which may hide

the discriminating information. Note that in the image on the right, a well-placed ROI would allow detection, but there is no semantic hint to select it.

V. CONCLUSION

We proposed a new CNN-based framework for image forgery detection. Thanks to suitable architectural solutions, the framework can be trained end-to-end, based only on weak (image-level) supervision, exploiting information gathered at full-resolution from the whole image. Therefore, all components of the framework, from feature extractor to classifier, are optimized jointly. We proved the effectiveness of this solution by extensive performance analysis on forensic datasets widespread in the community. A large performance gain is observed in all cases with respect to all reference methods. In addition, preliminary analysis show that the framework can also provide clues on forgery localization.

Despite the very promising results, there is still much room for improvement. In particular, better forms of pooling should be considered to preserve long-range spatial relationships in the aggregation phase; image and object semantics should be taken into account to complement the low-level information analyzed by the current framework; and a dedicated forgery localization tool should be designed based on the same approach. Work is under way along these paths.

ACKNOWLEDGMENT

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] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8713484

[2] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," 2016, *arXiv:1604.06174*. [Online]. Available: http://arxiv.org/abs/1604.06174

[3] G. Cao, Y. Zhao, R. Ni, and X. Li, "Contrast enhancement-based forensics in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 3, pp. 515–525, Mar. 2014.

[4] C. Zhang, D. Du, L. Ke, H. Qi, and S. Lyu, "Global contrast enhancement detection via deep multi-path network," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2815–2820.

[5] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[6] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient dense-field copy–move forgery detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2284–2297, Nov. 2015.

[7] X. Bi, C.-M. Pun, and X.-C. Yuan, "Multi-scale feature extraction and adaptive matching for copy-move forgery detection," *Multimedia Tools Appl.*, vol. 77, no. 1, pp. 363–385, Jan. 2018.

[8] C. Lin, W. Lu, X. Huang, K. Liu, W. Sun, H. Lin, and Z. Tan, "Copy-move forgery detection using combined features and transitive matching," *Multimedia Tools Appl.*, vol. 78, p. 30081–30096, Feb. 2019.

[9] B. Mahdian and S. Saic, "Using noise inconsistencies for blind image forensics," *Image Vis. Comput.*, vol. 27, no. 10, pp. 1497–1503, Sep. 2009.

[10] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1566–1577, Oct. 2012.

[11] S. Ye, Q. Sun, and E.-C. Chang, "Detecting digital image forgeries by measuring inconsistencies of blocking artifact," in *Proc. IEEE Multimedia Expo Int. Conf.*, Jul. 2007, pp. 12–15.

[12] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1003–1017, Jun. 2012.

[13] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, May 2008.

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

[15] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in *Proc. SPIE, Electron. Imag., Media Forensics Security*, vol. 7541, pp. 101–112, Oct. 2010.

[16] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery detection through residual-based local descriptors and block-matching," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5297–5301.

[17] X. Zhao, S. Wang, S. Li, and J. Li, "Passive image-splicing detection by a 2-D noncausal Markov model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 185–199, Feb. 2015.

[18] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Nov. 2015, pp. 1–6.

[19] H. Li, W. Luo, X. Qiu, and J. Huang, "Identification of various image operations using residual-based features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 31–45, Jan. 2018.

[20] S. Lyu and H. Farid, "How realistic is photorealistic?" *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 845–850, Feb. 2005.

[21] Y. Q. Shi, C. Chen, W. Su, and G. Xuan, "Steganalysis versus splicing detection," in *Proc. Int. Workshop Digit. Watermarking*, vol. 5041, Dec. 2008, pp. 158–172.

[22] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognit.*, vol. 45, no. 12, pp. 4292–4299, Dec. 2012.

[23] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[24] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[25] Y. Liu, Q. Guan, X. Zhao, and Y. Cao, "Image forgery localization based on multi-scale convolutional neural networks," in *Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2018, pp. 85–90.

[26] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proc. 4th ACM Workshop Inf. Hiding Multimedia Secur.*, 2016, pp. 5–10.

[27] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur.*, 2017, pp. 1–6.

[28] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1053–1061.

[29] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of GAN-generated fake images over social networks," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Apr. 2018, pp. 384–389.

[30] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1–11.

[31] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image splicing localization using a multi-task fully convolutional network (MFCN)," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 201–209, Feb. 2018.

[32] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and encoder–decoder architecture for detection of image forgeries," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3286–3300, Jul. 2019.

[33] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "ForensicTransfer: Weakly-supervised domain adaptation for forgery detection," 2018, *arXiv:1812.02510*. [Online]. Available: http://arxiv.org/abs/1812.02510
[34] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.
[35] L. Bondi, S. Lameri, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, "Tampering detection and localization through clustering of camera-based CNN features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1855–1864.
[36] D. Cozzolino and L. Verdoliva, "Camera-based image forgery localization using convolutional neural networks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 1372–1376.
[37] M. Huh, A. Liu, A. Owens, and A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 101–117.
[38] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Secur.*, vol. 2017, no. 1, pp. 1–16, Dec. 2017.
[39] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," *Proc. SPIE*, vol. 5307, pp. 472–480, Dec. 2003.
[40] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.
[41] T. Gloe and R. Böhme, "The 'Dresden image Database' for benchmarking digital image forensics," in *Proc. ACM Symp. Appl. Comput.*, 2010, pp. 1585–1591.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[43] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 4278–4284.
[44] Y. Wu, W. Abdalmageed, and P. Natarajan, "ManTra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9543–9552.
[45] T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. de Rezende Rocha, "Exposing digital image forgeries by illumination color classification," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 7, pp. 1182–1194, Jul. 2013.
[46] P. Korus and J. Huang, "Evaluation of random field models in multi-modal unsupervised tampering localization," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.
[47] H. Guan, M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus, "MFC datasets: Large-scale benchmark datasets for media forensic challenge evaluation," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2019, pp. 63–72.
[48] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

FRANCESCO MARRA (Member, IEEE) received the M.Sc. degree in computer science and the Ph.D. degree in information technology and electrical engineering from the University of Naples Federico II, Italy, in 2013 and 2018, respectively. He is currently a Postdoctoral Researcher position with the Image Processing Research Group at the University of Naples Federico II. His study and research interests include image processing, in particular, adversarial deep learning and forgery detection. He has been the Co-Chair of the 2nd International Workshop on Recent Advances in Digital Security: Biometrics and Forensics (BioFor'19). He is also serving as a Guest Editor for Elsevier *Computer Vision and Image Understanding*, and Elsevier *Pattern Recognition Letters* journals.

DIEGO GRAGNANIELLO (Member, IEEE) received the Laurea degree in telecommunications engineering and the Ph.D. degree in electronic and telecommunications engineering from the University of Naples Federico II, Italy, in 2011 and 2015, respectively.

He is currently a Postdoctoral Researcher with the Image Processing Research Group, University of Naples Federico II. His study and research interests include image processing, in particular, adversarial deep learning and forgery detection. He has been the Co-Chair of the 2nd International Workshop on Recent Advances in Digital Security: Biometrics and Forensics (BioFor'19). He is serving as Guest Editor of the special issues on Adversarial Deep Learning in Biometrics and Forensics for Elsevier *Computer Vision and Image Understanding*, and on Advances in Digital Security: Biometrics and Forensics for Elsevier *Pattern Recognition Letters*.

LUISA VERDOLIVA (Senior Member, IEEE) is currently an Associate Professor of telecommunications with the Department of Industrial Engineering, University Federico II of Naples, Italy. Her scientific interests are in the field of image processing, with main contributions in the area of multimedia forensics. She has been a General Co-Chair of the 2019 ACM Workshop on Information Hiding and Multimedia Security, a Technical Chair of the 2019 IEEE Workshop in Information Forensics and Security, and a Tutorial Chair of the 2016 IEEE Workshop in Information Forensics and Security. Since 2016, she has been an Area Chair of the IEEE International Conference on Image Processing. She is also a Vice-Chair of the IEEE Signal Processing Society's Information Forensics and Security Technical Committee. She is on the Editorial Board of the IEEE Transactions on Information Forensics and Security and IEEE Signal Processing Letters.

GIOVANNI POGGI (Member, IEEE) is currently a Full Professor of telecommunications with the University of Naples Federico II, Naples, Italy. His main research interests include digital multimedia forensics (forgery detection and localization and source identification), remote sensing image processing (restoration, segmentation, and classification of both optical and SAR images), and image biometrics. He is also an Associate Editor for MDPI *Remote Sensing*. He has been an Associate Editor of the IEEE Transactions on Image Processing and Elsevier *Signal Processing*.