

Image Splicing Detection, Localization and Attribution via JPEG Primary Quantization Matrix Estimation and Clustering

Yakun Niu¹, Benedetta Tondi², *Member, IEEE*, Yao Zhao¹, *Senior Member, IEEE*,
Rongrong Ni¹, *Member, IEEE*, and Mauro Barni², *Fellow, IEEE*

Abstract—Detection of inconsistencies of double JPEG artifacts across different image regions is often used to detect local image manipulations, like image splicing, and to localize them. In this paper, we move one step further, proposing an end-to-end system that, in addition to detecting and localizing spliced regions, can also distinguish regions coming from different donor images. We assume that both the spliced regions and the background image have undergone a double JPEG compression, and use a local estimate of the primary quantization matrix to distinguish between spliced regions taken from different sources. To do so, we cluster the image blocks according to the estimated primary quantization matrix and refine the result by means of morphological reconstruction. The proposed method can work in a wide variety of settings including aligned and non-aligned double JPEG compression, and regardless of whether the second compression is stronger or weaker than the first one. We validated the proposed approach by means of extensive experiments showing its superior performance with respect to baseline methods working in similar conditions.

Index Terms—Image forensics, double JPEG compression, image forgery localization, deep learning based image forensics, primary quantization matrix, spectral clustering, normalized mutual information (NMI).

I. INTRODUCTION

DETECTION of double JPEG (DJPEG) compression plays a major role in image forensics since double compression reveals important information about the past history of an image [1], [2]. This is the case of image splicing detection

Manuscript received February 2, 2021; revised June 16, 2021; accepted November 1, 2021. Date of publication November 19, 2021; date of current version December 3, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant U1936212 and Grant 62120106009, in part by the PREMIER Project under Contract PRIN 2017 2017Z595XS-001, in part by the Italian Ministry of University and Research, and in part by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under Agreement FA8750-20-2-1004. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Rémi Cograne. (*Corresponding author: Yao Zhao.*)

Yakun Niu, Yao Zhao, and Rongrong Ni are with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China, and also with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China (e-mail: niuyakun@bjtu.edu.cn; yzhao@bjtu.edu.cn; rni@bjtu.edu.cn).

Benedetta Tondi and Mauro Barni are with the Department of Information Engineering and Mathematics, University of Siena, 53100 Siena, Italy (e-mail: barni@dii.unisi.it; benedettatondi@gmail.com).

Digital Object Identifier 10.1109/TIFS.2021.3129654

and localization. When part of an image is spliced from a donor JPEG image into a target JPEG image to create a composite forgery (which is eventually recompressed so that the final image is double compressed), it is quite common that the compression setting used for the donor image is not equal to that used to compress the target image. As a consequence, the original and spliced regions of the forged image exhibit different (double) compression artifacts thus providing the basis for the detection and localization of the spliced region. Most of the methods proposed so far to detect image splicing based on double compression artifacts work under the following simple assumptions:

- 1) the tampered region (or regions) comes from a single donor image. Very few attempts have been made to identify forgeries containing multiple spliced areas coming from different donor images. Yet, given an image with several copy-pasted regions, it is possible, at least in principle, to identify the different origin of the tampered areas by recognizing that spliced regions coming from different donor images probably underwent a different double compression history;
- 2) the spliced region (or regions) is taken from a non-compressed image and spliced into a JPEG image. After recompression, the forged area has undergone only a single JPEG compression (SJPEG) while the background has been compressed twice [3]–[5]. In the following, we refer to this situation as DJPEG vs SJPEG detection. In practical scenarios, however, it is more likely that both the donor and the target images have been JPEG compressed, thus calling for the development of techniques capable to work in a DJPEG vs DJPEG setting.¹

In this paper, we propose a general approach to simultaneously perform image splicing detection, localization and attribution of regions coming from different donors. The proposed method, which is specifically thought to work in the DJPEG vs DJPEG scenario (but can also work in the SJPEG vs DJPEG case), relies on the estimation of the

¹The DJPEG vs DJPEG scenario is often addressed indirectly assuming that the second compression of the foreground is performed on a misaligned JPEG grid, while an aligned DJPEG is applied to the background (or viceversa). In this setting, many systems implicitly regard the background as SJPEG hence reducing this case to a SJPEG vs DJPEG scenario.

quantization matrix used in the first compression step of a JPEG image (in the following, we will refer to such a matrix as primary quantization matrix and we will indicate it by Q_1). Specifically, the proposed method works by providing a local blockwise estimate of the primary quantization matrix and then clustering the image blocks according to such an estimate. Splicing detection is achieved by recognizing the presence of more than one cluster, while the exact number of clusters identifies the number of donor images used to create the forgery. Eventually, by looking at blocks belonging to different clusters, we can localize the spliced areas and attribute them to different donor images. As an additional advantage, the proposed method also works when the second compression is stronger than the first one, e.g. when the quality factor used for the first compression (QF_1) is larger than that used for the second one (QF_2), that is when $QF_1 > QF_2$. Moreover, it can also cope with nonstandard quantization matrices.

For the estimation of the Q_1 matrix, we adopt the CNN-based estimator proposed in [6], due to its capability to provide a good estimation results also on small patches. To get the tampering map with the indication of the spliced regions, we associate to each 8×8 block of the image a vector with the estimated quantization steps, then we apply Spectral Clustering (SC) to such vectors [7]. In order to determine the number of clusters, we trained an ad-hoc Convolutional Neural Network (CNN) taking as input the estimated quantization steps. If only one cluster is found, the image is classified as a non-forged image and no further operation is carried out. In the presence of multiple clusters, we apply SC, then the largest cluster is associated to the image background, while the others are regarded as belonging to spliced regions, each cluster corresponding to a different donor image. The tampering map and the estimated number of spliced regions are finally refined by enforcing the spatial coherence and smoothness of the clusters through morphological reconstruction.

The main contributions of our work can be stated as follows:

- We introduce a new image forensics problem, hereafter referred to as donor image attribution, or simply attribution, whose goal is to decide if two or more spliced regions within a tampered image come from the same donor image and group them accordingly.
- We propose a new method to jointly localize tampered regions in JPEG images and distinguish between spliced regions coming from different donor images, under the assumption that the donor images have been compressed with different Q_1 matrices. We do so in the most common JPEG vs JPEG scenario. To the best of our knowledge, this is the first method performing localization and attribution of multiple spliced regions by relying on the analysis of compression traces. The system is based on the following ingredients: primary quantization matrix estimation, clustering and morphological reconstruction. We also designed and trained a CNN to identify the number of clusters present in the image by looking at the local estimate of the primary quantization matrix. Thanks to such a CNN, the proposed system is able to carry out splicing detection, localization and attribution

simultaneously, hence providing an end-to-end system for image splicing forensics.

- We carried out an extensive experimental campaign to evaluate the effectiveness of the proposed method in a wide variety of settings and tampering scenarios. In order to evaluate the performance of tampering localization in the case of spliced regions originating from multiple sources or donor images, we introduce a new metric based on the Normalized Mutual Information (NMI) metric, commonly used in pattern recognition applications to assess the performance of clustering.

The proposed approach is a very general one since it can be seamlessly applied in a wide variety of cases. First, it is one of the few approaches explicitly thought to work in a JPEG vs JPEG setting, secondly it works also when $QF_1 > QF_2$. Eventually, it maintains good performance regardless of whether the first and second compression grids are aligned or not. Moreover, the method can be applied also when non-standard quantization matrices are used and hence the quality factor QF is not defined.

The rest of this paper is organized as follows. After a brief review of related methods (Section II), in Section III, we present the general tampering localization setup considered in this paper and introduce the main notations. The proposed method for image tampering localization is described in Section IV. Section V describes the methodology we followed for the experimental analysis, whose results are reported in Section VI. We conclude the paper with some final remarks in Section VII.

II. PRIOR ART

Several techniques for image tampering localization have been proposed in the forensic literature, relying on different manipulation traces, e.g. inconsistencies of resampling artifacts [8], [9], or presence of different sensor noise patterns [10], [11]. More often, inconsistencies of JPEG compression artifacts are exploited. Due to the wide diffusion of the JPEG compression standard, in fact, image editing software often re-save the edited images in JPEG format, hence making it possible to detect and localize tampering based on the analysis of the traces of double JPEG compression and their inconsistencies across the tampered image. In the following, we briefly review the relevant literature about JPEG detection for image tampering localization.

It is well known that double JPEG compression leaves peculiar artifacts in the DCT domain, in particular, in the histograms of block-DCT coefficients [12]. Accordingly, many tampering detection algorithms rely on the statistical analysis of DCT coefficients [2], [13]. Some example of methods for detecting double compression artifacts in the non-aligned JPEG scenario relying on handcrafted features computed in the pixel or the DCT domain are described in [5], [14]–[16]. Early approaches were designed to work on the whole image, to detect if the analysed image has undergone a global single or double JPEG compression. Such methods are not applicable in a tampering detection scenario, where only part of the image

has been manipulated, due to the difficulty of estimating the required statistics on small blocks. To cope with this problem, a bunch of other methods have been developed for DJPEG localization [3], [17], [18]. In general, these methods have low spatial resolution and their performance drop significantly when regions smaller than 256×256 are considered. More recently, a new class of CNN-based methods have been proposed. They are able to improve the spatial resolution of DJPEG localization and then can be conveniently used for tampering localization, (see, for instance, [19], [20] - for both aligned and not-aligned DJPEG detection, and [4], [21] - for the aligned DJPEG case). All the above methods focus on the SJPEG vs DJPEG scenario, that is, they work under the assumption that the tampered areas are single compressed while the background is double compressed. In contrast, very few work has been done to specifically address the more challenging DJPEG vs DJPEG scenario considered in this paper. In principle, methods capable to estimate the primary quantization matrix of DJPEG images, e.g. [1] and [22], could be applied to this scenario. However, the methods in [1], [22] work on the full image, and hence are not suitable for localization. In [23], a technique is proposed to detect whether part of an image was formerly compressed with a JPEG quality lower than that used for the rest of the image ($QF_1 < QF_2$), by means of exhaustive recompression with every quality factors.

The works that are more closely related to this paper are [24] and [25]. Both these methods estimate the Q_1 matrix on a local basis and output a map with the probability that a DCT block has been double-compressed. The method in [24] works under the assumption that the histograms of the unquantized DCT coefficients are locally uniform in the non-tampered region. Moreover, accurate detection can be achieved when the spatial resolution is larger than 256×256 pixel. Two approaches are proposed in [24] for the cases of aligned and non-aligned DJPEG. The method in [25] is designed for the case of aligned DJPEG compression. Both methods work better when $QF_2 > QF_1$, while performance are significantly worse in the opposite case.

Being able to distinguish spliced regions coming from different donor images, our method can also be used for image phylogeny, where the identification of the donor images is a required step to identify the relationships between the spliced image and its parent images [26]–[28] and use it to reconstruct the history of semantically similar images. From this perspective, the goal of the system described in this paper is not very different from that of image phylogeny applications, the main difference being that in the image phylogeny scenario the donor images are assumed to be available to the analyst.

Finally, we point that several methods based on CNNs have been developed in the last few years addressing other tampering detection and localization problems, e.g. the problem of copy-move tampering [29]–[31]. Moreover, CNN-based approaches addressing the problem of forgery detection and localization by looking for general traces of manipulation have also been proposed, see for instance [32]–[34].



Fig. 1. Image tampering setup considered in this paper.

III. PROBLEM STATEMENT

Let Q denote the 8×8 matrix with the quantization steps of the DCT coefficients, namely, the quantization matrix, used for JPEG compression. The image tampering scenario considered in this paper, i.e. the DJPEG vs DJPEG scenario, is illustrated in Fig. 1. The spliced regions (referred to as foreground regions), possibly coming from different donor images, and the background, are *doubly* JPEG compressed. Different quantization matrices have been used for the former compression (Q_1). In the figure, Q_1 denotes the quantization matrix of the background, Q'_1 and Q''_1 the quantization matrices of the spliced regions. The tampered image is finally JPEG compressed with another quantization matrix (Q_2). The resulting image is then a double compressed JPEG image. The second compression can be either aligned or non-aligned to the first one, depending on the position of the 8×8 JPEG compression grid. A misalignment occurs in the background, for instance, when the image is cropped between the former and the second compression stage. With regard to the spliced area(s), when a region of a JPEG image is copy-pasted into another JPEG image, it is very likely that the alignment between the compression grids is not preserved and the final JPEG compression will not be aligned with the grid of the spliced area(s). In the following, we denote the aligned DJPEG scenario, i.e., when no misalignment occurs between the two compressions, with the acronym A-DJPEG, and the non-aligned DJPEG scenario with NA-DJPEG.

In the scenario described above, spliced regions coming from different donor images can be distinguished by relying on the inconsistencies between the primary quantization matrices. Let k be the total number of donor images. Accordingly, for a pristine image, $k = 1$. For a tampered image, k corresponds to the number of donor images plus the background and the total number of donor images used to create the forgery being then $k - 1$. In this setting, the system we have developed aims at solving three different problems: tampering detection, localization and attribution. The detection part outputs a binary decision on the presence or absence of tampering based on the estimated k . When tampering is detected ($k > 1$), a tampering localization map is returned by the system. The tampering map is a coloured map with different colours assigned to the background pixels and to the pixels of spliced regions coming from different donors. Source attribution is performed based on the colours of the spliced regions.

In the following, we introduce the main notations used throughout the paper. We denote by \mathbf{q}_1 the 64-dim vector with the elements of the 8×8 Q_1 matrix, taken in zig-zag order [35]. The quantization steps corresponding to the medium-high frequencies are more difficult to estimate, since they are quantized more heavily. However, they are usually less

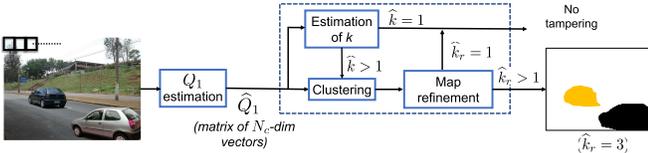


Fig. 2. Architecture of the proposed method for splicing detection, localization and attribution of JPEG images based on Q_1 matrix inconsistencies.

important since they tend to be similar for most quantization matrices. For this reason, as in most part of related literature, the estimation is restricted to the first N_c elements of \mathbf{q}_1 . Hereafter, we will use the symbol \mathbf{q}_1 to indicate only the first N_c quantization steps. With a slight abuse of notation, we denote with $\hat{Q}_1(\cdot, \cdot, \cdot)$ the tensor with the estimated primary quantization steps computed on 8×8 blocks. Specifically, for a given (i, j) , $\hat{Q}_1(i, j, v)$ corresponds to the estimation of the v -th element of \mathbf{q}_1 for the 8×8 block of pixels in the position indicated by (i, j) .

IV. PROPOSED METHOD

The overall architecture of the proposed method is illustrated in Fig. 2. Given a possibly tampered JPEG image, a block-wise estimation of the primary quantization matrix is first obtained (first block in Fig. 2), yielding \hat{Q}_1 , then splicing detection, localization and attribution is performed by clustering the image blocks according to the result of the estimation followed by a map refinement step (dashed block in Fig. 2).²

The number of clusters is first estimated from the \hat{Q}_1 tensor via a CNN model, then a clustering algorithm is applied to \hat{Q}_1 to obtain the tampering map. Specifically, the N_c -dim vectors with the estimated quantization steps of the image blocks are regarded as points in the clustering space. As a result of clustering, a label is associated to each block of the image. In this way, the clustering labels in the tampering map indicate the different donor images used to build the tampered image. A map refinement step is finally performed on the clustering map to improve the quality of the map based on spatial information.

In the above scheme, \hat{k} denotes the estimated value of k . Tampering detection is carried out on the basis of the estimated value of k . In particular, if $\hat{k} = 1$, the image is declared to be pristine and the process ends. If $\hat{k} > 1$, the clustering algorithm is applied to perform splicing localization and attribution, followed by map refinement. After map refinement, the number of clusters in the map might change. We let \hat{k}_r be the number of clusters after the refinement step. Then, if $\hat{k}_r = 1$, the image is declared to be pristine, while if $\hat{k}_r > 1$, the image is judged to be tampered. By looking at the labels of the different clusters, we can localize the spliced areas and attribute them to different donor images.

A detailed description of each block of Fig. 2 is provided in the following subsections.

²In principle, a similar pipeline can be used in conjunction with other local features extracted from the image.

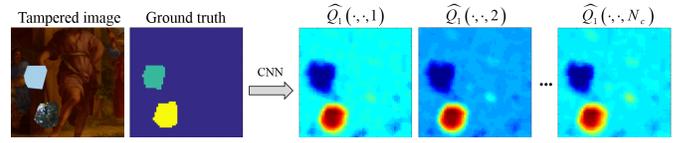


Fig. 3. Example of \hat{Q}_1 tensor for various quantization steps. The location of spliced regions can be easily spot from all the bands of the tensor.

A. Patch-Based Estimation of the Primary Quantization Matrix

The primary quantization matrix is estimated on a local window basis, by means of a CNN estimator. In particular, we chose the CNN-based approach described in [6], due to its ability to work regardless of the alignment/misalignment of the first and second compression grids and to the good performance obtained even when $QF_2 < QF_1$.

The network adopted in [6] for primary quantization matrix estimation has an input size of 64×64 , and N_c final output nodes, each providing the estimated value of the quantization step of a DCT coefficient. Given an input patch \mathbf{x} , the CNN is trained to minimize the difference between the predicted values $\mathbf{f}(\mathbf{x})$ and the true vector $\mathbf{q}_1(\mathbf{x})$. Rounding is performed independently on each element of the output vector to get the final prediction, that is $\hat{\mathbf{q}}_1(\mathbf{x}) = \text{round}(\mathbf{f}(\mathbf{x}))$.

Since the CNN is applied patch-wise to the input image, the estimation step returns a tensor with the N_c -dim vectors with the primary quantization steps of each 64×64 block. Specifically, given an input image of size $R \times C \times 3$, the CNN estimator is run on 64×64 overlapping patches (each shifted by 8 pixels with respect to the previous one). The estimated vector, then, is assigned to the central 8×8 block³ of the patch. At the end, a tensor \hat{Q}_1 with the estimated quantization steps is obtained. More precisely, by setting to 8 the stride s used to slide the estimation window over the image, and by assuming, for simplicity, that R and C are multiple of 8, the estimated tensor \hat{Q}_1 has size $R' \times C' \times N_c$, where $R' = R/8 - 7$, $C' = C/8 - 7$. With reference to the notation introduced in the previous section, $i \in \{1, 2, \dots, R'\}$ and $j \in \{1, 2, \dots, C'\}$. As we said, we set $N_c = 15$. Note that, for simplicity, we are not considering blocks close to the border of the image.⁴ In the following, we use the compact notation $\hat{Q}_{1,t}$ to denote the t -th estimated vector, that is $\hat{Q}_{1,t} = \hat{\mathbf{q}}_1(\mathbf{x}_t)$, where \mathbf{x}_t is the t -th 64×64 patch fed into the CNN estimator in left-to-right, top-to-bottom, scanning.

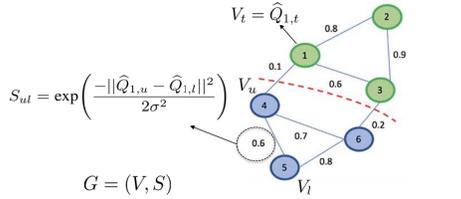
Fig. 3 shows an example of a tensor \hat{Q}_1 estimated by the CNN. We can observe that the various components of the tensor corresponding to different quantization steps provide useful information regarding the position and the provenance of the spliced areas.

B. Localization and Attribution of Spliced Regions

To localize and attribute multiple spliced regions to different donor images based on the estimated quantization matrix,

³Given that the patch contains an even number of blocks, rigorously speaking a central block does not exist. In the following we denote the block in the fourth block-row and fourth block-column as the central block.

⁴If needed, we can incorporate such blocks into the analysis by mirroring the border blocks.


 Fig. 4. Similarity graph associated to the \hat{Q}_1 tensor.

we adopted the Spectral Clustering (SC) algorithm [7]. SC has been used in the last years in related image forensics fields, including, for instance, camera identification [36] and mobile phone clustering [37]. Recently, SC has also been considered for splicing detection [38], to improve the performance of a deep-learning-based forensic approach for general forgery localization.

Based on some preliminary tests we carried out, SC provides better results compared to other clustering methods, like expectation maximization [39], hierarchical clustering [40], fuzzy clustering [41] and, in particular, K -means clustering.

SC exploits graph theory to map points (in our case, the N_c -dim vectors $\hat{Q}_{1,t} = \hat{\mathbf{q}}_1(\mathbf{x}_t)$) to a low dimensional space [7]. The problem of clustering is reformulated by using a similarity graph. More specifically, an undirected similarity graph $G = (V, S)$, where V denotes the set of vertexes or nodes and S the set of edges, is associated to the \hat{Q}_1 tensor as shown in Fig. 4. Each element of the tensor, i.e., each $\hat{Q}_{1,t}$ vector, represents a node of the graph.

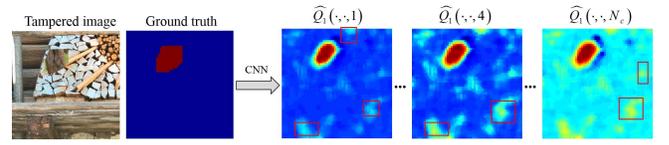
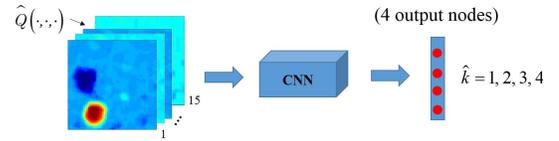
The number of nodes $N = |V|$ of the graph corresponds to the number of 8×8 blocks in the image. Then, $S \in \mathbb{R}^N \times \mathbb{R}^N$. The edge weights S_{ij} represent the similarity of the nodes i and j , and in our case they are defined as: $S_{ij} = \exp(-\|\hat{Q}_{1,i} - \hat{Q}_{1,j}\|^2 / 2\sigma^2)$. The choice of the scale parameter σ^2 is not obvious, so we determined it experimentally.

The goal of SC is to find a partition of the graph such that the edges within a group (cluster) have high weights, i.e., the points within the same cluster are similar to each other, and the edges between different groups (clusters) have very low weights, i.e., the points belonging to different clusters are different from each other. To do so, the SC algorithm computes the Laplacian matrix associated to G , then it applies the K -means algorithm to the eigenvalues matrix.

C. CNN-Based Estimation of the Number of Clusters

In principle, the spectral clustering algorithm would provide a way to estimate the number of clusters k by relying on graph theory, that is, through the analysis of the eigenvalues $(\lambda_i)_{i=1}^N$ of the Laplacian matrix associated to the graph (see [7]). Specifically, the eigenvalues are listed in descending order and the index i corresponding to the maximum gap between two consecutive eigenvalues $\lambda_{i+1} - \lambda_i$ is selected as \hat{k} , that is $\hat{k} = \arg \max_i (\lambda_{i+1} - \lambda_i)$. Based on our experiments, estimating k in this way provides poor results.

By carefully analyzing the results of the preliminary experiments we run to estimate k by means of SC, we concluded that the bad performance we got are due to the fact that SC does not exploit any spatial information. This prevents


 Fig. 5. Scattered clusters of \hat{Q}_1 in the spatial domain.

 Fig. 6. Input and output of the CNN used for estimating k from the \hat{Q}_1 tensor.

to estimate k correctly even in cases that appear easy to solve by visual inspection. Fig. 5 shows an example in which there are several scattered areas with yellow color in the estimated tensor \hat{Q}_1 . Such scattered areas will be mistakenly regarded as one cluster by SC, even if their spatial incoherence makes it very unlikely that they correspond to a spliced region. As a result, the estimated k by SC is 3 while the true one is 2. By exploiting the spatial information, such spatially scattered areas can be identified as a noisy cluster and assigned to the background.

To do so, we trained a CNN to estimate the number of clusters k directly from the tensor \hat{Q}_1 . The CNN has a number of output nodes equal to 4 (Fig. 6), corresponding to a maximum of 4 clusters.⁵ We chose an architecture commonly and successfully used for pattern recognition applications, namely the VGG-16 [42] network. The VGG-16 network has a number of 16 layers in total and 3 Fully Connected (FC) layers. The input layer is modified to accept input images with size $256 \times 256 \times 15$ (given an input with generic size $R' \times C' \times 15$, each of the 15-dim quantization maps are resized to normalize the input to the size required by the network). The CNN has been trained on \hat{Q}_1 tensors computed from pristine images, for $k = 1$, and tampered images, for $k = 2, 3, 4$. The dataset creation process for the tampered images and the setup considered for network training and validation is described in Section V-B.1, while the details of the training process are provided in Section V-D.

D. Tampering Map Refinement by Means of Morphological Reconstruction

A visual analysis of the tampering maps obtained after the application of the SC algorithm often reveals the presence of spurious isolated regions that do not correspond to any spliced region. By referring to Fig. 7 as an example, we observe two types of spurious regions: small and usually scattered regions belonging to the same cluster of one big region, and ring-shaped regions along the boundary of (usually big) spliced

⁵This choice was made based on the resolution capability of the estimation of the Q_1 matrix. From our experiments, when there are 5 or more distinct donor images with different compression factors QF_1 s, using a lower number of clusters yields better localization results. Specifically, the best performance are obtained using $\hat{k} = 4$ (see Section VI-A). In these cases, in fact, the difference between the QF_1 values is often small (5 or less), then the estimated coefficients are very close to each other and the clustering algorithm is not capable to correctly separate the clusters.

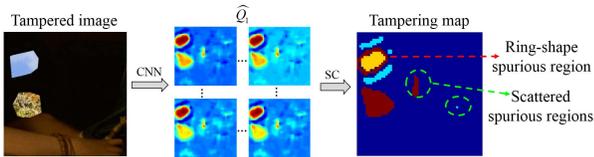


Fig. 7. Ring-shape and scattered spurious regions in the preliminary tampering map obtained after clustering.

areas. The first type of spurious regions are due to the scattered presence of image blocks for which the estimated quantization steps are similar to those of truly spliced regions. The presence of such regions is due to the noisiness of the Q_1 estimation step and to the lack of spatial information during the clustering phase. Ring-shaped spurious regions are due to the window-based approach used for Q_1 estimation. In the proximity of the boundary of spliced regions, in fact, the estimation window includes blocks from both the spliced and background regions, thus producing a somewhat mixed estimated vector. For large spliced regions, the number of blocks with the mixed estimate is large enough to represent a separated ring-shaped cluster (the mixed estimated quantization steps may also resemble the values of another truly spliced region, as the brown ring-shaped region in Fig. 7). We also observed that spurious regions are more frequent when $\hat{k} > k$, that is when the SC algorithm is run with a \hat{k} larger than the correct one.

Of course, the presence of spurious clusters reduces the performance of our algorithm in terms of localization (and attribution) accuracy. For this reason, we introduced a map refinement step, aiming at improving the quality of the tampering map based on spatial information. We did so by resorting to morphological reconstruction (MR [43]). In particular, the map refinement procedure consists of the following sequence of morphological operations:

- 1) for every cluster we consider the region formed by the pixels belonging to the cluster;
- 2) we apply a predefined number of *erosion* iterations with a small structuring element;
- 3) we apply a *conditional dilation* procedure starting from the regions (referred to as marks or *seeds* according to the terminology of morphological reconstruction theory [43]) obtained at the end of the erosion phase in 2.

The conditional dilation procedure works as follows: i) first, each seed is expanded by means of a conditional dilation, where the dilation is applied only to the pixels belonging to the cluster the region corresponds to. This procedure is carried out in parallel on the seeds of all clusters; ii) the regions obtained at the end of the previous step, are further dilated conditioned to all the pixels (if any) that do not belong to the background cluster and that have not been assigned yet; iii) if, at a given iteration, regions belonging to different clusters are expanded on the same pixels, disputed pixels are assigned randomly to one of the clusters.

The main goal of MR is to reassign pixels belonging to ring-shaped clusters. Fig. 8 shows the results of the process after each of the steps describe above. In the erosion step, the

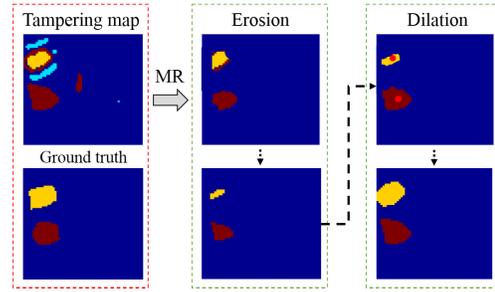


Fig. 8. The sequence of morphological operations in MR.

ring-shaped cluster and the isolated cluster are removed while the interior of the spliced regions are kept (see the erosion map). After the conditional dilation procedure, the pixels of the ring-shaped region are reassigned to the corresponding inner cluster (see the dilation map). Note that isolated (usually small) regions are completely eroded during the erosion step and are not reassigned during the conditional dilation. The choice of the number of erosion iterations and the size of the structuring element is a crucial one, all the more that the optimum setting depends on the size of the spliced areas. For a given structuring element, a too large number of iterations may cause the removal of spliced regions, while if the number of iterations is too small, the risk is that small isolated clusters are not removed. Given the difficulties of determining the best setting on a theoretical basis, we tuned the system by means of experimental analysis.

After the application of the MR procedure, the number of clusters might change. In particular, the final number \hat{k}_r of clusters may be *lower* than \hat{k} , because isolated clusters have been removed or ring-shaped clusters have been reassigned to the internal clusters. This happens especially when $k \geq 3$ (see Section VI-A). If after MR the number of remaining clusters is equal to 1, the image is considered to be a pristine one. Experiments carried out on several tampered images reveal that MR can indeed help to remove noisy clusters and the undesired rings around compact regions from the maps. The benefits obtained with the map refinement procedure are illustrated in the examples reported in Fig. 9. The number of erosion iterations considered in those examples is 2 and the size of the (disk-shaped) structuring element is 1. In the examples reported in the figure, at the end of the MR procedure, the ring-shaped clusters are reassigned to the internal clusters and then $\hat{k}_r < \hat{k}$. We notice that the estimation of k can be worse after map refinement, however, especially when k is large, a better clustering result is obtained when the value of k is underestimated. This is the case with the last example in Fig. 9, where we see that using 3 clusters instead of 4 permits to remove the two rings around one of the spliced regions. This, apparently counterintuitive, behavior is due to the fact that, especially when k is large, the value of the quantization coefficients may be similar for different donor images, and hence it might not be easy to correctly identify the clusters using the true k . In such cases, a better map is obtained by assigning the regions originating from the donor images with similar quantization matrices to the same cluster.

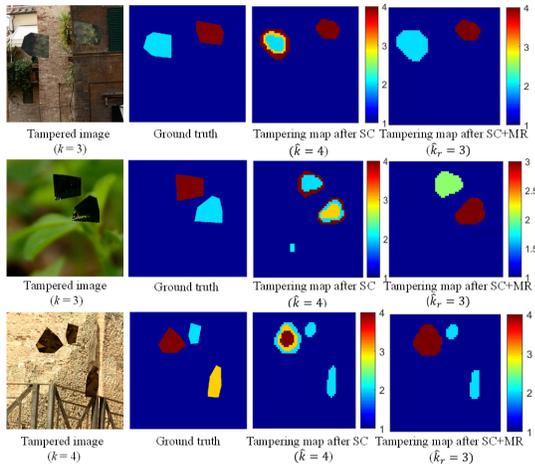


Fig. 9. Examples of tampering maps before and after the application of morphological reconstruction.

V. EXPERIMENTAL METHODOLOGY

In this section, we describe the methodology we followed to run the experiments whereby we validated the effectiveness of the proposed method. We first present the evaluation metrics used to measure the performance of the proposed tool to detect and localize the tampered areas. Then we introduce a metric explicitly thought to measure the effectiveness with regard to the attribution task. Afterwards, we pass to the description of the procedure that we followed to generate the tampered contents in the DJPEG scenario, and introduce the datasets used for: i) training and testing the CNN model for the estimation of k ; ii) assessing the detection, localization and attribution performance of the system. Finally, we describe the two closest related state-of-the-art methods and describe how they are applied for a fair comparison with the results obtained by our system.

For simplicity and without loss of generality, in most of our experiments, we considered standard quantization matrices, so we refer to them by means of the Quality Factor (QF). Specifically, we denote with QF_2 the QF used for the second compression and with QF_1 that of the primary compression.

A. Evaluation Metrics

In this section, we introduce the metrics used to measure the performance of our algorithm.

1) *Tampering Detection*: Tampering detection is a binary classification problem. For tampered images, we have a correct detection when $\hat{k}_r > 1$, while for pristine images, the detection is correct when $\hat{k}_r = 1$, wrong in all the other cases.

By adopting a common terminology in detection theory, in the following we assume that tampered images ($k > 1$) belong to the positive class and pristine images ($k = 1$) to the negative one. A Neyman Pearson setup is considered for the decision. Accordingly, we fixed the maximum admissible False Positive Rate ($FPR = Pr(\hat{k}_r > 1 | k = 1)$), i.e., the percentage of pristine images wrongly detected as tampered, and we evaluate the True Positive Rate ($TPR = Pr(\hat{k}_r > 1 | k > 1)$), namely, the percentage of correctly detected tampered images. The overall accuracy is given by the fraction

of correct decisions on both tampered and pristine images over the total number of tested images.

2) *Tampering Localization and Attribution*: With regard to the metrics for assessing the localization and attribution performance, we observe that we should not only evaluate the capability of the system to localize the tampered areas, but also the capability to identify the regions spliced from different donor images as belonging to different clusters.

Let us first focus on the former task. Tampering localization can be regarded as a binary classification problem applied at the pixel level. Pixels belong to one of two classes, the background (the negative class) or the foreground (tampered or positive class). To measure the localization performance, we consider the Matthews Correlation Coefficient (MCC) [44], defined as:

$$MCC = \frac{TP \times TN + FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (1)$$

where TP is the number of true positives pixels, i.e., the pixels correctly classified as tampered, TN the number of true negative pixels, i.e., the pixels correctly classified as non-tampered, FP the number of false positives and FN the number of false negatives. If any of the sums in brackets at the denominator is zero, the denominator is arbitrarily set to one. MCC is particularly helpful in the case of unbalanced classes, as it is almost always the case for tampering localization.⁶

The identification of spliced regions coming from different donor images is a new goal addressed in this paper, so no established metric exists to measure the performance with respect to this task. To fill this gap, we introduce a new metric called Normalized Mutual Information (NMI) [45].

To start with, we observe that the output of the clustering algorithm is a map assigning to each pixel a label, ranging from 1 to \hat{k}_r , indicating the cluster the pixel belongs to. The ground-truth map indicates for every pixel the true cluster, i.e., the corresponding donor image for foreground pixels or the background cluster. Note that, the exact label assigned to each cluster is irrelevant, as long as pixels coming from different donor images are assigned to different clusters. So it is not necessary that the labels of the clustering map are identical to those of the ground truth. As an additional difficulty, we observe that the number of clusters contained in the map output by the tampering localization and attribution system does not need to be equal to the number of clusters in the ground-truth map.

In the following, we will refer to the labels assigned to the regions of the ground-truth map as pixel classes. Let y denote the class label ($y = 1, \dots, k$) and c denote the cluster label ($c = 1, \dots, \hat{k}_r$) in the output clustering map. The NMI index is defined as:

$$NMI(y, c) = \frac{2I(y; c)}{H(y) + H(c)}, \quad (2)$$

where $H(y)$ and $H(c)$ denote, respectively, the empirical Entropy of y and c , and $I(y; c)$ the empirical Mutual Information between y and c . More precisely, let

⁶In the case of highly unbalanced classes, the overall accuracy is not a good indicator of the performance given that errors on the minority class have virtually no impact.

$p_y(i) = p(y = i) = \{\text{\#pixels in class } i\} / \text{total no. of pixels}$,
 $p_c(j) = p(c = j) = \{\text{\#pixels in cluster } j\} / \text{total no. of pixels}$, and $p_{y|c}(i|j) = p_{y|c}(y = i|c = j)$ where

$$p_{y|c}(i|j) = \frac{\{\text{\#pixels of class } i \text{ assigned to cluster } j\}}{\{\text{\#pixels in cluster } j\}}, \quad (3)$$

where the total number of pixels is $R' \cdot C'$. Then, $H(y) = \sum_{i=1}^k p_y(i) \log p_y(i)$, and

$$I(y; c) = \sum_{i=1}^k \sum_{j=1}^{\hat{k}} p_{y|c}(i|j) p_c(j) \log \left(\frac{p_{y|c}(i|j)}{p_y(i)} \right). \quad (4)$$

In case of perfect clustering, it is easy to see that $H(y) = H(c)$, while $I(y; c) = H(y)$, then $\text{NMI} = 1$. Note that being a normalized quantity, the NMI allows comparing cases with a different number of clusters.

B. Datasets

To build the datasets for our experiments, we started from the 8156 camera-native uncompressed large size images in the RAISE8K dataset [46]. We divided these images in two sets: 7000 images to be used for training (and validation) and 1156 images for the tests. On the average, about 5 non-overlapping patches are extracted from each RAISE image, for a total number of 41000 patches, 35000 of which (coming from the set of 7000 original images), denoted by S_{tr} , were used to produce the pristine and tampered images for training the models and the remaining 5780, denoted by S_{ts} , to produce the pristine and tampered images for the tests.

We considered two types of DJPEG pristine and tampered images, named Type I and Type II, described below, respectively for the case of Aligned DJPEG (A-DJPEG) and the case of Non-Aligned DJPEG (NA-DJPEG).⁷

- For **Type I** images: the pristine images are A-DJPEG images. This means that the first 8×8 DCT compression grid is aligned with the grid of the second compression. With regard to the tampered images, the first JPEG compression grid of the background (or, equivalently, of the source image) is aligned with the grid of the second compression, while the grid for the first compression of the foreground is misaligned with that of the second compression. Specifically, a random misalignment (r, c) , $0 \leq r, c \leq 7$, $(r, c) \neq (0, 0)$ is considered for the grid of the former compression of the foreground.
- For **Type II** images: we assume that the images are first JPEG compressed using a DCT grid shifted by a quantity (r, c) , randomly chosen in $0 \leq r, c \leq 7$, $(r, c) \neq (0, 0)$, with respect to the upper left corner, while for the second compression no grid misalignment is considered. Then, the pristine images are NA-DJPEG. For the tampered images, the JPEG grid of the background is non-aligned with the grid of the second compression. The same holds

⁷To be precise, we refer to the Type I (Type II) set as aligned (not-aligned) set by implicitly referring to the compression of the pristine images in the set, and, similarly, to the compression of the background of the tampered regions, while the compression of the foreground is always misaligned.

for the foreground regions. Note that the misalignments of foreground and background are generally different.

The datasets we used for our experiments are available online, together with a report detailing the exact procedure we have followed to build the pristine and tampered images for the various k and combination of QF_s .⁸ Below we provide a description of the datasets considered for the various tests. The size of the images in all the datasets is 512×512 .

1) *Dataset for Q_1 Matrix Estimation (training and testing)*: To build the datasets for training and testing the CNN for the estimation of Q_1 , we followed exactly [6]. Training and testing were carried out on 64×64 patches, obtained from the set of 7000 and 1156 images of RAISE. For DJPEG, a random grid shift (r, c) is applied between the two compressions, $0 \leq r, c \leq 7$, then, as in [6], the A-DJPEG case occurs with probability $1/64$.

2) *Dataset for k Estimation (training and testing)*: The dataset used to train and test the CNN for the estimation of the number of clusters consists of:

- a set \mathcal{D}_{tr} of 18000 images for each k (for a total of 72000 images) used for training. The set is obtained from 18000 (randomly chosen) images in S_{tr} ;
- a set \mathcal{D}_{ts} of 4000 images for $k = 1$ and 4000 images for $k > 1$, in equal proportions for $k = 2, 3$ and 4. The set is obtained from 4000 images in S_{ts} .

Let $V = \{60, 65, 70, 75, 80, 85, 95, 98\}$. To get the pristine images, QF_1 is randomly chosen in V and $QF_2 = 90$. For the tampered images, when $k = 2$, QF_1 is randomly chosen in $\{75, 85, 95, 98\}$, and $QF_{1,2} \in V$, $QF_{1,2} \neq QF_1$. When $k = 3$, $QF_1 \in \{75, 85, 95, 98\}$, and $QF_{1,2}, QF_{1,3} \in V$, $QF_{1,2} \neq QF_{1,3} \neq QF_1$. Finally, for $k = 4$, $QF_1 \in \{75, 85, 95, 98\}$, and $QF_{1,2} \neq QF_{1,3} \neq QF_{1,4} \neq QF_1 \in V$. The height h and width w of the bounding-box of the tampered regions are randomly selected in $\{64, 96, 128, 156\}$. Misalignment is applied to the background with 0.5 probability, then the dataset consists of both Type I and Type II images in similar proportions.

3) *Dataset for Detection, Localization and Attribution Tests*: Detection performance are measured over the same dataset \mathcal{D}_{ts} considered to test the CNN for k estimation, where we have 4000 images representative of the negative class (pristine), and 4000 for the positive class (tampered). The threshold achieving the desired FPR is set on these 4000 pristine images. To better assess the localization performance, and to ease the comparison with state-of-the-art methods (see the next section), we additionally built two separate Type I and Type II datasets, named \mathcal{D}_I and \mathcal{D}_{II} , whose images are generated from S_{ts} under specific setting. Specifically, in both \mathcal{D}_I and \mathcal{D}_{II} , we considered 100 images for every combination of k and $\{QF_{1,i}\}_{i=2}^k$, for the tampering sizes $h \times w = 96 \times 96$ and 128×128 .

A summary of the datasets used in our experiments is reported in Table I.

4) *Additional Datasets*: Although most of the experiments were carried out using the datasets described so far, to test

⁸The document is made available at: https://drive.google.com/drive/folders/1ck-Xm1G3dxgGN717B_JVdKMPaBaPCZ3Ap, along with the datasets.

TABLE I
DATASETS OF IMAGES CONSIDERED IN OUR EXPERIMENTS

Name	\mathcal{D}_{tr}	\mathcal{D}_{ts}	\mathcal{D}_I	\mathcal{D}_{II}
Purpose	Training	Test	Test	Test
Original dataset	S_{tr}	S_{ts}	S_{ts}	S_{ts}
No. images	18000 per k (72000 total)	4000 per $k = 1$ 4000 per $k > 1$	100 per each (k , $\{QF_{1,i}\}, h \times w$)	100 per each (k , $\{QF_{1,i}\}, h \times w$)
DJPEG	Type I and II (50% each)	Type I and II (50% each)	Type I	Type II
Setting	$\{QF_{1,i}\}, h \times w$ randomly chosen	$\{QF_{1,i}\}, h \times w$ randomly chosen	$h \times w =$ $\{96 \times 96,$ $128 \times 128\}$	$h \times w =$ $\{96 \times 96$ $128 \times 128\}$

the generalization performance of our method in different and more challenging scenarios, the following datasets have also been considered:

- The datasets used in [24], named $\mathcal{D}_{[24]}$, whose images have resolution 1024×1024 . These images are obtained from a personal database of uncompressed images. $QF_1 \in \{50 : 100\}$ with step 5. The DJPEG vs SJPEG scenario is considered in this dataset for the tampering. The tampering region corresponds to the central portion (1/16 of the image). Two datasets are provided, one for the aligned case ($\mathcal{D}_{[24],A}$) - the foreground is A-DJPEG-, and one for the non-aligned case ($\mathcal{D}_{[24],NA}$) - the foreground is NA-DJPEG. In all cases $k = 2$, in particular, we considered the images with $QF_2 = 90$ for comparison.
- The dataset used [25], named $\mathcal{D}_{[25]}$, whose images have been obtained starting from the uncompressed images in the UCID dataset [47], with size 512×384 . The primary quality factor QF_1 was chosen in the set $\{55, 65, 75, 85, 95\}$. The DJPEG vs SJPEG scenario is considered for the tampering. The tampering region is 10% of the image, whose location is randomly chosen. In all cases $k = 2$. For our comparison, we considered the images with $QF_2 = 90$.
- A dataset built as detailed above but starting from uncompressed images of Dresden dataset [48], named \mathcal{D}_{Dr} . For the tampered area we let $h \times v = 96 \times 96, 128 \times 128,$ and 156×156 . We built Type I and Type II images with $k = 2, 3, 4$. For each setting, we considered 100 images for every combination of QF_1 .
- A dataset built from S_{ts} in which the first compression was carried out by using Photoshop, with PS qualities in the range [7:12] (the second compression is the same as before). The more general non-aligned scenario is considered. The tampering sizes are set to $h \times v = 96 \times 96, 128 \times 128,$ and 156×156 with $k = 2$. 100 images are considered for each PS quality in each setting. The results are reported in Table XIX.

Examples of tampered images from $\mathcal{D}_{[24]}$ and $\mathcal{D}_{[25]}$ are provided in Fig. 10. These images are built by cutting a central portion of the image, compressing it and pasting it back in the same position, thus not leaving any visual boundary artifacts.

C. Baseline Methods for Comparison

The baselines we compared our method with are the methods described in [24] and [25], since, as we said in Section II, these are the methods most closely related to



Fig. 10. Examples of tampered images from $\mathcal{D}_{[24]}$ (left) and $\mathcal{D}_{[25]}$ (right).

the system proposed in this paper. In [24], statistical models are used to build a map reporting the likelihood that image blocks have been double compressed. The cases of A-DJPEG and NA-DJPEG are treated separately. In [25], the authors propose a strategy to estimate the posterior probability that an image block has been tampered with, by minimizing a properly defined energy function. The approach works only for the case of A-DJPEG compression. Similarly to our system, both these methods are designed for tampering localization in double compressed images and can be applied to a scenario wherein both the background and spliced regions are double compressed but exhibit different compression artifacts (i.e., they are compressed with a different quality factor), since they are based on the estimation of the DCT quantization steps used for the first compression. Both methods perform well when the former compression quality is *lower* than the second, while they provide poor performance in the reverse case.

To turn the tampering localization map provided by the baseline methods into a tampering detection output, we followed the approach used in [25] and trained an SVM classifier with 3 features. The first feature is given by the perimeter-area ratio of the localized tampered region. The second one is the percentage of pixels detected as tampered. These two features are used to characterize scattered and small regions, that are usually linked to false alarms. The third feature is a measure of the periodicity consistency between the DCT coefficient histograms of the localized region and the entire image. We refer to [25] for more details. To train the SVM, we considered 3000 pristine ($k = 1$) and 3000 tampered images (for $k = 2, 3, 4$, where each k is represented in equal proportions), obtained from the images in S_{tr} as detailed in the previous section, with the difference that only Type I images are considered for [25], while Type I images and Type II images are considered for the methods in [24], respectively for the A-DJPEG and NA-DJPEG cases.⁹ Given the trained model, the operating point is determined from the ROC curve by fixing the desired FPR and deriving the SVM decision threshold accordingly. About 2000 pristine images obtained from the images in S_{ts} (different from those used to build the datasets for the tests) are considered to set the threshold, for the Type I setting and Type II setting respectively.

We stress that the comparison with [24] and [25] is possible only when the considered setting satisfies the operative conditions such methods have been built for. In fact, an advantage of our method is that it is much more general than [24] and [25], so that it can work in a wider variety of situations. In addition, our method is able to distinguish between spliced regions coming from different donor images, which is something

⁹The same proportion of pristine and tampered images is considered in these cases (the SVM is trained for the binary tampering detection task).

TABLE II
OPERATIVE CONDITIONS OF OUR SYSTEM
AND THE METHODS IN [24] AND [25]

	DJPEG grid	Background setting	Non-std Q_1 matrix	Purpose
[24]-AI	Aligned	$QF_1 < QF_2$, $QF_1 > QF_2$ (poor)	Yes	Localization
[24]-NAI	Non-Aligned	$QF_1 < QF_2$, $QF_1 > QF_2$ (poor)	Yes	Localization
[25]	Aligned	$QF_1 < QF_2$, $QF_1 > QF_2$ (poor)	Yes	Localization
Our	Both	$QF_1 \leq QF_2$	Yes	Localization Attribution

that neither [24] nor [25] can do. Such aspects must be taken into account for a fair judgement of the improvement allowed by the method proposed in this paper. The operative conditions, and expected performance of the various methods under different settings, are summarized in Table II.

For sake of completeness, we also compared our method with an anomaly-based detector that performs forgery localization by looking for general traces of manipulation, hence not focusing on JPEG compression artifacts. In particular, we considered a recently proposed deep learning method, named MantraNet [33], based on anomaly detection, that performs joint image-level detection and pixel-level localization of forgeries, regarded as local image anomalies.

D. Parameter Setting

Below, we report the setting of the parameters considered to implement the various steps of our system. To train the CNN for Q_1 estimation, we followed exactly [6]. Then, the network is trained on 64×64 patches (4×10^5 for each QF_1) obtained from the set of 7000 RAISE images used for training as detailed in [6]. The network is trained for 60 epochs with batch size 32. The solver is the Adam optimizer. The learning rate is 10^{-5} . The \hat{Q}_1 map is obtained as described in Section IV-A. With regard to the CNN used to estimate k , as we said, we considered the VGG-16 network [42]. We started from the pre-trained solution for ImageNet classification, and re-trained the network on the dataset described in Section V-B.1 for 50 epochs, with batch size 16 and applying data augmentation. The Adam optimizer with learning rate 10^{-5} was considered as solver. Regarding the spectral clustering algorithm, the scale parameter σ is set to 0.6 if the estimated k is 2, while we use 0.15 if $k = 3$ or 4. Finally, for the morphological reconstruction, we considered 2 erosion iterations with a disk-shaped structuring element of radius 1. We found experimentally that this setting is a good choice for the range of tampering sizes that we are considering.

VI. EXPERIMENTAL RESULTS

The results of our experiments are reported and discussed in this section for a wide variety of settings.

A. Accuracy of k Estimation

The estimation of k is a crucial step since it directly affects the performance of tampering detection, localization and attribution. In particular, if $k = 1$ and the estimated $\hat{k}_r > 1$,

TABLE III
CONFUSION MATRIX OF THE CNN-BASED k ESTIMATION (LEFT) AND
OF THE FINAL \hat{k}_r (RIGHT), COMPUTED OVER THE \mathcal{D}_{ts} SET

$k \setminus \hat{k}$	1	2	3	4	$k \setminus \hat{k}_r$	1	2	3	4
1	0.95	0.05	0.00	0.00	1	0.95	0.05	0.00	0.00
2	0.09	0.82	0.08	0.01	2	0.09	0.84	0.06	0.01
3	0.01	0.21	0.67	0.11	3	0.01	0.28	0.64	0.07
4	0.00	0.04	0.24	0.72	4	0.00	0.09	0.36	0.55

TABLE IV
THE DETECTION ACCURACY OF THE PROPOSED METHOD

$k \setminus \hat{k}_r$	$\hat{k}_r = 1$	$\hat{k}_r > 1$
$k = 1$	0.95	0.05
$k > 1$	0.03	0.97

TABLE V
COMPARISON OF THE DETECTION PERFORMANCE OF OUR METHOD
WITH [24] AND [25]. THE TABLE REPORTS THE TPR WHEN FPR=0.05
OVER \mathcal{D}_{ts} , \mathcal{D}_{ts} (TYPE I ONLY) AND \mathcal{D}_{ts} (TYPE II ONLY)

	\mathcal{D}_{ts}	\mathcal{D}_{ts} (Type I)			\mathcal{D}_{ts} (Type II)		
		All	$QF_1 < QF_2$	$QF_1 > QF_2$	All	$QF_1 < QF_2$	$QF_1 > QF_2$
[24]-AI	-	0.31	0.50	0.14	-	-	-
[24]-NAI	-	-	-	-	0.18	0.25	0.11
[25]	-	0.39	0.61	0.17	-	-	-
Our	0.97	0.97	0.97	0.96	0.96	0.97	0.95

TABLE VI
AVERAGE LOCALIZATION PERFORMANCE OF THE METHODS (MCC),
AVERAGED ON THE SET OF TP IMAGES FOR EACH METHOD

	\mathcal{D}_{ts}	\mathcal{D}_{ts} (Type I)			\mathcal{D}_{ts} (Type II)		
		All	$QF_1 < QF_2$	$QF_1 > QF_2$	All	$QF_1 < QF_2$	$QF_1 > QF_2$
[24]-AI	-	0.63	0.77	0.09	-	-	-
[24]-NAI	-	-	-	-	0.47	0.58	0.20
[25]	-	0.64	0.81	0.02	-	-	-
Our	0.64	0.65	0.61	0.69	0.63	0.59	0.67

pristine images are mistakenly identified as tampered images, and vice versa.

The results of the k estimation step are reported in the confusion matrix shown in Table III (left). The average accuracy of the estimation is 0.79. Somewhat expectedly, the estimation works better for smaller k , since the minimum difference between the QF_1 values considered tends to be smaller in the presence of multiple spliced regions.

Table III (right) reports the number of clusters \hat{k}_r after spectral clustering and map refinement. The average accuracy of the estimation now is 0.75. Although \hat{k}_r is a worse estimate, better localization results can be obtained with \hat{k}_r . In fact, as we mentioned at the end of Section IV-D, in some cases, better clustering results can be obtained by underestimating k . This is especially the case for large k , when using the true value may result in wrong clusters, e.g. ring-shaped and noisy clusters - see the examples in Fig. 9. This is confirmed by the results of our ablation study (end of Section VI-D). For the same reason, we also found that there is no need to consider more than 4 clusters: in fact, when $k > 4$, the best localization results are obtained using $\hat{k} = 4$, that is, assigning the regions originating from donor images with similar quantization matrices to the same cluster. With $k = 5$, for instance, we obtain a gain of 0.007 in MCC and 0.004 in NMI by using 4 clusters instead of 5. The gain is more significant for larger k .

TABLE VII
LOCALIZATION PERFORMANCE (MCC) FOR $k = 2$, FOR THE CASES OF TAMPERING SIZE 96×96 (LEFT) AND 128×128 (RIGHT). PERFORMANCE ARE MEASURED ON \mathcal{D}_{II} . THE NUMBER OF TP IMAGES IS REPORTED IN BRACKETS

$QF_{1,2}$		65	75	85	95	98	$QF_{1,2}$		65	75	85	95	98
QF_1							QF_1						
75	[24]-NAI	0.659 (10)	—	0.643(9)	0.650(13)	0.667(12)	75	[24]-NAI	0.775 (25)	—	0.744(30)	0.540(26)	0.741(31)
	Our	0.428(92)	—	0.701 (99)	0.794 (99)	0.799 (100)		Our	0.542(97)	—	0.771 (100)	0.833 (100)	0.845 (100)
85	[24]-NAI	0.102(3)	0.091(14)	—	0.114(14)	0.123(10)	85	[24]-NAI	0.179(10)	0.201(9)	—	0.230(9)	0.286(8)
	Our	0.738 (98)	0.558 (98)	—	0.646 (94)	0.708 (97)		Our	0.786 (100)	0.701 (98)	—	0.694 (99)	0.757 (100)
95	[24]-NAI	0.086(12)	0.103(15)	0.108(17)	—	0.153(13)	95	[24]-NAI	0.186(16)	0.183(12)	0.199(9)	—	0.240 (8)
	Our	0.805 (98)	0.769 (98)	0.665 (95)	—	0.159 (19)		Our	0.826 (99)	0.847 (100)	0.703 (98)	—	0.136(27)
98	[24]-NAI	0.090(17)	0.151(18)	0.140(8)	0.102(13)	—	98	[24]-NAI	0.223(14)	0.131(5)	0.103(7)	0.175(8)	—
	Our	0.804 (96)	0.777 (98)	0.698 (94)	0.421 (41)	—		Our	0.835 (100)	0.850 (100)	0.778 (100)	0.546 (100)	—

TABLE VIII
LOCALIZATION PERFORMANCE (MCC) FOR $k = 3$, 128×128 , $QF_1 = 85$ (LEFT), $QF_1 = 95$ (RIGHT). PERFORMANCE ARE MEASURED ON \mathcal{D}_{II} . THE NUMBER OF TP IMAGES IS REPORTED IN BRACKETS

$QF_{1,3}$		65	75	95	98	$QF_{1,3}$		65	75	85	98
$QF_{1,2}$						$QF_{1,2}$					
65	[24]-NAI	—	0.198(11)	0.356(7)	0.232(12)	65	[24]-NAI	—	0.168(15)	0.178(15)	0.260(23)
	Our	—	0.728 (100)	0.634 (100)	0.684 (100)		Our	—	0.814 (100)	0.679 (100)	0.565 (100)
75	[24]-NAI	0.127(10)	—	0.254(12)	0.243(9)	75	[24]-NAI	0.292(9)	—	0.148(11)	0.255(22)
	Our	0.717 (100)	—	0.614 (99)	0.636 (100)		Our	0.817 (100)	—	0.720 (100)	0.560 (98)
95	[24]-NAI	0.221(12)	0.192(18)	—	0.356(11)	85	[24]-NAI	0.198(18)	0.336(17)	—	0.137(11)
	Our	0.667 (100)	0.621 (100)	—	0.701 (99)		Our	0.688 (100)	0.694 (99)	—	0.463 (98)
98	[24]-NAI	0.376(8)	0.176(16)	0.329(8)	—	98	[24]-NAI	0.145(11)	0.148(14)	0.274(13)	—
	Our	0.705 (100)	0.626 (100)	0.694 (99)	—		Our	0.595 (100)	0.584 (100)	0.498 (100)	—

TABLE IX
LOCALIZATION PERFORMANCE (MCC) FOR $k = 4$, 128×128 , $QF_1 = 85$ (LEFT), $QF_1 = 95$ (RIGHT). $QF_{1,2}$ IS SET TO 60. PERFORMANCE ARE MEASURED ON \mathcal{D}_{II} . THE NUMBER OF TP IMAGES IS REPORTED IN BRACKETS

$QF_{1,4}$		65	75	95	98	$QF_{1,4}$		65	75	85	98
$QF_{1,3}$						$QF_{1,3}$					
65	[24]-NAI	—	0.322(14)	0.240(14)	0.320(21)	65	[24]-NAI	—	0.198(20)	0.228(19)	0.297(13)
	Our	—	0.741 (100)	0.707 (100)	0.710 (100)		Our	—	0.807 (100)	0.738 (100)	0.646 (100)
75	[24]-NAI	0.381(14)	—	0.309(16)	0.410(12)	75	[24]-NAI	0.211(18)	—	0.371(15)	0.277(13)
	Our	0.709 (100)	—	0.664 (100)	0.659 (100)		Our	0.797 (100)	—	0.734 (100)	0.625 (100)
95	[24]-NAI	0.334(11)	0.293(17)	—	0.335(17)	85	[24]-NAI	0.280(16)	0.290(15)	—	0.233(25)
	Our	0.689 (100)	0.656 (100)	—	0.724 (100)		Our	0.730 (100)	0.719 (100)	—	0.477 (100)
98	[24]-NAI	0.288(17)	0.235(13)	0.269(11)	—	98	[24]-NAI	0.239(17)	0.140(17)	0.266(19)	—
	Our	0.757 (100)	0.692 (100)	0.698 (100)	—		Our	0.643 (100)	0.632 (100)	0.494 (100)	—

TABLE X
LOCALIZATION PERFORMANCE (MCC) FOR $k = 2$, 96×96 (LEFT) AND 128×128 (RIGHT). PERFORMANCE ARE MEASURED ON \mathcal{D}_I . THE NUMBER OF TP IMAGES IS REPORTED IN BRACKETS

$QF_{1,2}$		65	75	85	95	98	$QF_{1,2}$		65	75	85	95	98
QF_1							QF_1						
75	[24]-AI	0.878(10)	—	0.862(9)	0.851(13)	0.888 (12)	75	[24]-AI	0.864(58)	—	0.872(74)	0.856(66)	0.857(65)
	[25]	0.891 (60)	—	0.885 (54)	0.875 (70)	0.879(68)		[25]	0.893 (58)	—	0.891 (71)	0.892 (71)	0.893 (73)
	Our	0.476(98)	—	0.744(100)	0.796(100)	0.814(99)		Our	0.581(99)	—	0.808(100)	0.846(100)	0.853(100)
85	[24]-AI	0.417(11)	0.608(13)	—	0.463(19)	0.590(22)	85	[24]-AI	0.618(41)	0.606(34)	—	0.621(34)	0.617(42)
	[25]	0.725 (40)	0.848 (50)	—	0.706(47)	0.781 (43)		[25]	0.732(57)	0.791 (52)	—	0.823 (53)	0.797 (64)
	Our	0.710(100)	0.523(93)	—	0.738 (98)	0.734(100)		Our	0.810 (100)	0.650(100)	—	0.821(98)	0.797(99)
95	[24]-AI	0.044(6)	0.054(14)	0.069(21)	—	0.081(15)	95	[24]-AI	0.049(15)	0.075(11)	0.096(9)	—	0.059(16)
	[25]	0.002(24)	0.051(14)	0.056(11)	—	0.000(16)		[25]	0.000(100)	0.011(18)	0.020(11)	—	0.039(21)
	Our	0.792 (99)	0.775 (100)	0.717 (98)	—	0.210 (34)		Our	0.850 (100)	0.833 (100)	0.702 (99)	—	0.137 (36)
98	[24]-AI	0.039(15)	0.023(6)	0.163(12)	0.027(13)	—	98	[24]-AI	0.099(12)	0.043(5)	0.188(15)	0.101(10)	—
	[25]	0.000(18)	0.000(10)	0.000(14)	0.006(21)	—		[25]	0.051(15)	0.012(13)	0.007(20)	0.006(19)	—
	Our	0.804 (100)	0.784 (99)	0.680 (94)	0.377 (34)	—		Our	0.857 (100)	0.840 (100)	0.772 (99)	0.425 (49)	—

B. Detection Performance ($k = 1$ Vs $k > 1$)

With regard to the detection performance ($k = 1$ vs $k > 1$) of our system, it can be derived from Table III (right) by

measuring the capability of distinguishing pristine ($k = 1$) from tampered images ($k > 1$). The detection accuracies we got for $k = 1$, $k > 1$ and over the \mathcal{D}_{ts} set (we remind that

TABLE XI
LOCALIZATION PERFORMANCE (MCC) FOR $k = 3$, 128×128 , $QF_1 = 85$ (LEFT), $QF_1 = 95$ (RIGHT). PERFORMANCE ARE MEASURED ON \mathcal{D}_I . THE NUMBER OF TP IMAGES IS REPORTED IN BRACKETS

$\frac{QF_{1,3}}{QF_{1,2}}$		65	75	95	98	$\frac{QF_{1,3}}{QF_{1,2}}$		65	75	85	98
65	[24]-AI	—	0.641(43)	0.695(45)	0.722(51)	65	[24]-AI	—	0.133(15)	0.028(14)	0.116(19)
	[25]	—	0.782 (50)	0.768 (68)	0.829 (57)		[25]	—	0.000(9)	0.034(17)	0.063(19)
	Our	—	0.685(100)	0.671(100)	0.686(100)		Our	—	0.799 (100)	0.726 (100)	0.564 (100)
75	[24]-AI	0.725(45)	—	0.771(48)	0.676(51)	75	[24]-AI	0.136(12)	—	0.089(19)	0.056(22)
	[25]	0.781 (57)	—	0.842 (60)	0.732 (64)		[25]	0.000(12)	—	0.044(16)	0.000(14)
	Our	0.671(100)	—	0.654(99)	0.636(100)		Our	0.802 (100)	—	0.758 (100)	0.550 (98)
95	[24]-AI	0.714(47)	0.602(46)	—	0.698(41)	85	[24]-AI	0.092(15)	0.128(21)	—	0.097(11)
	[25]	0.816 (58)	0.798 (58)	—	0.838 (65)		[25]	0.059(11)	0.030(25)	—	0.078(14)
	Our	0.683(100)	0.618(100)	—	0.768(100)		Our	0.697 (99)	0.749 (100)	—	0.539 (97)
98	[24]-AI	0.706(42)	0.623(43)	0.687(49)	—	98	[24]-AI	0.090(12)	0.110(12)	0.083(15)	—
	[25]	0.771 (60)	0.740 (59)	0.791(54)	—		[25]	0.000(12)	0.035(18)	0.026(18)	—
	Our	0.656(100)	0.616(99)	0.798 (94)	—		Our	0.593 (100)	0.578 (100)	0.545 (100)	—

TABLE XII
LOCALIZATION PERFORMANCE (MCC) FOR $k = 4$, 128×128 , $QF_1 = 85$ (LEFT) AND $QF_1 = 95$ (RIGHT), $QF_{1,2}$ IS SET TO 60. PERFORMANCE ARE MEASURED ON \mathcal{D}_I . THE NUMBER OF TP IMAGES IS REPORTED IN BRACKETS

$\frac{QF_{1,4}}{QF_{1,3}}$		65	75	95	98	$\frac{QF_{1,4}}{QF_{1,3}}$		65	75	85	98
65	[24]-AI	—	0.679(40)	0.619(41)	0.592(40)	65	[24]-AI	—	0.135(22)	0.170(16)	0.221(14)
	[25]	—	0.802 (53)	0.762 (54)	0.767 (51)		[25]	—	0.000(8)	0.152(8)	0.033(12)
	Our	—	0.695(100)	0.707(100)	0.714(100)		Our	—	0.803 (100)	0.755 (100)	0.625 (100)
75	[24]-AI	0.646(50)	—	0.639(44)	0.648(47)	75	[24]-AI	0.119(12)	—	0.054(17)	0.118(8)
	[25]	0.792 (58)	—	0.780 (50)	0.787 (55)		[25]	0.091(13)	—	0.081(8)	0.000(15)
	Our	0.709(100)	—	0.664(100)	0.679(100)		Our	0.821 (100)	—	0.770 (100)	0.621 (100)
95	[24]-AI	0.682(39)	0.672(39)	—	0.710(39)	85	[24]-AI	0.113(11)	0.065(21)	—	0.142(20)
	[25]	0.766 (52)	0.802 (58)	—	0.793 (51)		[25]	0.090(9)	0.088(17)	—	0.054(9)
	Our	0.702(100)	0.647(100)	—	0.720(100)		Our	0.774 (100)	0.752 (100)	—	0.483 (100)
98	[24]-AI	0.673(38)	0.684(43)	0.687(41)	—	98	[24]-AI	0.105(22)	0.113(8)	0.069(16)	—
	[25]	0.769 (55)	0.796 (44)	0.814 (57)	—		[25]	0.084(11)	0.054(13)	0.058(14)	—
	Our	0.724(100)	0.693(100)	0.725(100)	—		Our	0.637 (100)	0.668 (100)	0.534 (100)	—

this set comprises both Type I and Type II images in equal percentage) are 0.95, 0.97 (see Table IV) and 0.96 respectively; specifically, we got $TPR = 0.97$ at $FPR = 0.05$. Table V (last row) reports the accuracy results split for different settings. Specifically, we got $TPR = 0.97$ for Type I and 0.96 for Type II tamperings.

The comparison with [24] and [25], where detection is performed via SVM classification as described in Section V-C is reported in Table V, where TPR values corresponding to $FPR = 0.05$ are reported for both methods in Type I and Type II scenarios. For the baselines, the table reports only values obtained under the operative conditions the methods were thought for. As it can be seen, the proposed method greatly outperforms the baselines in the various settings. The poor results of [24] and [25] are in line with those reported in the reference papers, and are mainly due to the weak performance achieved by these methods in the scenario $QF_1 > QF_2$ (while the performance when $QF_1 < QF_2$ are good, in their operative scenarios).

C. Localization Performance ($k > 1$)

The results we obtained for tampering localization, averaged on the TP images only, that is the images correctly detected

TABLE XIII
AVERAGE ATTRIBUTION PERFORMANCE (NMI)
OF THE PROPOSED METHOD

Test set	\mathcal{D}_{ts}	\mathcal{D}_{ts} (Type I)	\mathcal{D}_{ts} (Type II)
NMI	0.475	0.481	0.468

as tampered, are reported in Table VI. Not surprisingly, the proposed method works better than method [24] in the Type II scenario, since the CNN-based Q_1 estimator is designed to work particularly for the NA-DJPEG case (the aligned case is assumed to occur with probability 1/64). The performances in the Type I scenario are also good and slightly better than those achieved by the best performing method [25] on the average.

The capability to work both in the case of A-DJPEG and NA-DJPEG is a noticeable property of our method, since the information about the alignment of the compression grids is not in general available.

Tables VII through IX show the localization results of the proposed method for the NA-DJPEG case in the various settings, for various combinations of the quality factors of the background and foreground regions. Specifically, Table VII reports the results for $k = 2$, for two different tampering sizes, namely 96×96 and 128×128 . Table VIII and Table IX

TABLE XIV

ATTRIBUTION PERFORMANCE (NMI) OF THE PROPOSED METHOD FOR $k = 3$ ((A) AND (C)) AND $k = 4$ ((B) AND (D)), TAMPERING SIZE 128×128 , $QF_1 = 85$. (A) AND (B) REFER TO THE NON-ALIGNED CASE (PERFORMANCE MEASURED ON \mathcal{D}_{II}), (C) AND (D) REFER TO THE ALIGNED CASE (PERFORMANCE MEASURED ON \mathcal{D}_I). FOR $k = 4$, $QF_{1,2}$ IS SET TO 60

(a)					(b)					(c)					(d)				
$QF_{1,3}$ \ $QF_{1,2}$	65	75	95	98	$QF_{1,4}$ \ $QF_{1,3}$	65	75	95	98	$QF_{1,3}$ \ $QF_{1,2}$	65	75	95	98	$QF_{1,4}$ \ $QF_{1,3}$	65	75	95	98
65	—	0.559	0.497	0.532	65	—	0.555	0.535	0.536	65	—	0.518	0.524	0.531	65	—	0.525	0.535	0.548
75	0.547	—	0.463	0.482	75	0.529	—	0.515	0.503	75	0.507	—	0.496	0.682	75	0.529	—	0.515	0.523
95	0.516	0.467	—	0.488	95	0.527	0.502	—	0.521	95	0.536	0.467	—	0.553	95	0.532	0.498	—	0.514
98	0.508	0.475	0.476	—	98	0.573	0.532	0.503	—	98	0.501	0.468	0.553	—	98	0.548	0.495	0.523	—

TABLE XV

RESULTS OF THE ABLATION STUDY (MCC/NMI)

	Type I	Type II
$\hat{k} + K$ -means	0.567/0.409	0.520/0.374
SC	0.636/0.443	0.610/0.423
$\hat{k} + SC$	0.643/0.470	0.622/0.456
$\hat{k} + SC + MR$	0.652/0.481	0.633/0.468

TABLE XVI

AVERAGE LOCALIZATION PERFORMANCE (MCC) ON THE DATASETS IN [24] AND [25]. TPR IS REPORTED WITHIN BRACKETS

Method	$\mathcal{D}_{[24],A}$	$\mathcal{D}_{[24],NA}$	$\mathcal{D}_{[25]}$
[24]-AI	0.657(0.46)	—	0.664(0.46)
[24]-NAI	—	0.518(0.53)	—
[25]	0.781(0.92)	—	0.701(0.91)
Our	0.904(0.82)	0.906(0.80)	0.712(0.89)

report the results for $k = 3$ and $k = 4$, for two different values of the quality factor of the background QF_1 of the background, i.e., $QF_1 = 85$ and 95 , when the tampering size is equal to 128×128 . Similarly, Tables X through XII show the localization results for the A-DJPG case, for $k = 2, 3$ and 4 respectively. The sets \mathcal{D}_I and \mathcal{D}_{II} are considered for these tests, respectively in the NA-DJPG and A-DJPG case.

Expectedly, better results are achieved when the tampering size is large (see Tables VII and VIII), for $k = 2$. The average difference in the MCC between the tampering size of 96×96 and 128×128 for the other values of k is similar, ranging from 0.030 to 0.090 depending on the setting. We can observe that our method greatly outperforms [24] in all the settings for the NA-DJPG case. Regarding the performance in the A-DJPG scenario, the method in [25] always outperforms [24] (the A-DJPG method) when $k > 2$ and $QF_1 < 90$ (QF_2), while when $QF_1 > QF_2$ both methods cannot correctly localize tampering. Compared to our method, the performance of [25] are superior in all the cases when the background QF_1 is smaller than 90, with a gain in the MCC which is about 0.1.¹⁰ The performance loss in these cases is the price to pay for a general method, that can work in all the settings of $QF_{1,i}$ and QF_2 , and focuses on the more probable NA-DJPG scenario (and then is not specifically designed for the A-DJPG scenario).

D. Attribution Performance ($k > 1$)

As opposed to state-of-the-art methods, the system introduced in this paper allows to distinguish spliced regions

¹⁰These values are not directly comparable since they are averaged on a different image sets, which in the case of the proposed method is much larger.

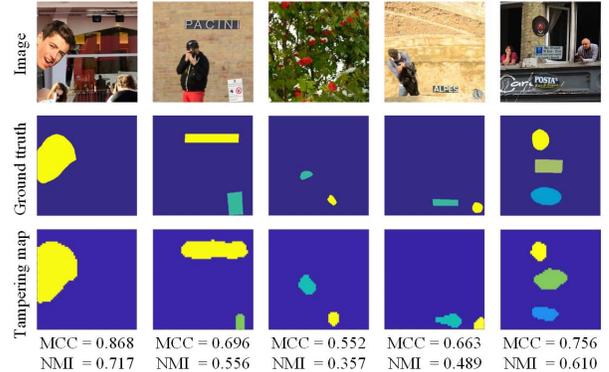


Fig. 11. Examples of the results provided by our system on some tampered images, for the case of A-DJPG, with $QF_1 < QF_2$. The ground truth and the output maps produced by our method are reported, along with the MCC and NMI values.

coming from different donor images. Fig. 11 shows some examples of the output maps obtained on some tampered images with different k . The corresponding NMI value measuring the goodness of the clustering (see Section V-A) is also reported. We can see that, even if the NMI indexes can theoretically reach 1, satisfactory clustering results are already obtained with much lower NMI values.

The average NMI values obtained over the test set \mathcal{D}_{IS} and the subsets of Type I and Type II images in \mathcal{D}_{IS} are reported in Table XIII. The NMI values of the state-of-the-art methods are not reported since they do not distinguish spliced regions coming from different donor images. The average NMI in the Type I and II cases are very similar. Notice that a value of the NMI around 0.5 is satisfactory, since the localization and clustering results are already good with an NMI around 0.6, while the NMI is close to zero when either the localization or the clustering result is poor, then the average NMI is not expected to be very high. The clustering performance for some settings of $k > 2$, with background JPEG quality $QF_1 = 85$ (measured on 100 TP images each) are reported in Table XIV. Similar NMI values are obtained for the various combinations of quality factors of the foreground regions.

E. Ablation Study

We also carried out an ablation study to measure the impact of the most important parameters of the system on localization and attribution accuracy. The results we got on the dataset \mathcal{D}_{IS} are reported in Table XV. The performance of K -means clustering are also reported.

The last line corresponds to the proposed pipeline, including the step of refinement of the map (the final number of clusters

TABLE XVII
AVERAGE LOCALIZATION PERFORMANCE (MCC) ON \mathcal{D}_{Dr} FOR VARIOUS TAMPERING SIZES, AVERAGED ON THE SET OF TP IMAGES. THE TPR IS REPORTED WITHIN BRACKETS

	96 × 96			128 × 128			156 × 156		
	All	Type I	Type II	All	Type I	Type II	All	Type I	Type II
[24]-A1	—	0.532(0.46)	—	—	0.549(0.61)	—	—	0.570(0.63)	—
[24]-NA1	—	—	0.504(0.30)	—	—	0.507(0.26)	—	—	0.511(0.29)
[25]	—	0.544(0.50)	—	—	0.573(0.39)	—	—	0.589(0.61)	—
Our	0.568(0.97)	0.579(0.97)	0.557(0.97)	0.641(0.98)	0.645(0.97)	0.636(0.98)	0.683(0.98)	0.693(0.98)	0.673(0.97)

TABLE XVIII
AVERAGE ATTRIBUTION PERFORMANCE (NMI) OF THE PROPOSED METHOD ON \mathcal{D}_{Dr}

Test set	\mathcal{D}_{Dr}	\mathcal{D}_{Dr} (Type I)	\mathcal{D}_{Dr} (Type II)
NMI	0.481	0.486	0.476

TABLE XIX
LOCALIZATION PERFORMANCE (MCC) WHEN THE FIRST COMPRESSION IS DONE WITH PHOTOSHOP (PS). TPR IS REPORTED AMONG BRACKETS

	96 × 96	128 × 128	156 × 156
[24]-NA1	0.285(0.35)	0.334(0.36)	0.362(0.37)
Our	0.534(0.66)	0.590(0.71)	0.620(0.73)

TABLE XX
LOCALIZATION PERFORMANCE (MCC) FOR THE CASE 128 × 128, WHEN THE FIRST COMPRESSION IS CARRIED OUT WITH THE PS SOFTWARE

QF_1 \ $QF_{1,2}$	7	8	9	10	11	12
7	[24]-NA1 — Our —	0.690(46) 0.721(94)	0.686(39) 0.760(97)	0.723(50) 0.784(99)	0.698(54) 0.803(97)	0.784(51) 0.802(99)
8	[24]-NA1 0.375(52) Our 0.484(97)	— —	0.364(52) 0.385(72)	0.305(52) 0.489(83)	0.356(50) 0.524(87)	0.365(55) 0.566(85)
9	[24]-NA1 0.195(43) Our 0.654(85)	0.150(37) 0.447(33)	— —	0.158(48) 0.208(21)	0.168(42) 0.235(21)	0.159(41) 0.198(24)
10	[24]-NA1 0.177(29) Our 0.755(100)	0.157(28) 0.616(96)	0.196(24) 0.354(71)	— —	0.180(28) 0.115(32)	0.168(26) 0.072(31)
11	[24]-NA1 0.205(24) Our 0.786(100)	0.186(26) 0.683(98)	0.181(26) 0.532(78)	0.194(29) 0.320(44)	— —	0.198(21) 0.157(34)
12	[24]-NA1 0.166(27) Our 0.802(99)	0.229(25) 0.708(99)	0.173(23) 0.595(81)	0.183(25) 0.357(53)	0.172(25) 0.211(31)	— —

is \hat{k}_r). We see that the best results can be obtained in this case. Although the improvement in the NMI is not big, it is a relevant one, given the poor sensitivity of this metric.

F. Results Under Mismatched Testing Conditions

To test the generalization capability of our system, with particular reference to the CNN-based components, we run some experiments under different and more challenging testing conditions. The results of our tests are reported below.

1) *Mismatched Datasets*: The results we got on the datasets used in [24] and [25] are shown in Table XVI. By looking at this table, we see that the localization results of our method on the datasets $\mathcal{D}_{[24]}$ (A and NA) and $\mathcal{D}_{[25]}$ are very good, better than those achieved on \mathcal{D}_{ts} . The tamperings in these datasets are in fact performed under less challenging conditions compared to our home-made dataset, only in the case $k = 2$, with a rather large tampering area and considering smaller QF_1 values, facilitating the detection and localization tasks (in \mathcal{D}_{ts} , QF_1 is almost always above 75 and never goes below 60). It is also worth observing that our method is not

designed for the DJPEG vs SJPEG scenario considered in these datasets.¹¹

The average localization results on the \mathcal{D}_{Dr} datasets are reported in Table XVII, for various tampering sizes. These results are similar to those achieved on \mathcal{D}_{ts} , confirming the good performance of our method even in the presence of dataset mismatch. The localization performance for the various combinations of the quality factors of the background and foreground regions reveal that, similarly to what happen in the experiments with the matched dataset, the advantage of our method with respect to the state-of-the-art is stronger for large quality factors of the background. The results are not reported for sake of brevity.

Attribution performance can also be measured on this dataset. The average NMI values obtained on \mathcal{D}_{Dr} are reported in Table XVIII and are similar to those achieved on \mathcal{D}_{ts} .

2) *Non-Standard Quantization Matrices*: To assess the behavior of the system with non-standard quantization matrices, we run some tests on images for which the first compression was carried out by using Photoshop (PS). The results are reported in Table XIX.

We see that our method can generalize to non-standard matrices and still outperform the methods in [24] and [25]. Table XX shows the results for the various combinations of PS qualities, for case 128 × 128. We see that the cases where the background has lower quality than the foreground, and they are close to each other, are the most difficult cases for our method. The detection performance are also poor in these cases. Overall, we notice that these are particularly relevant results since these test corresponds to a case of strong mismatch between training and test data used for the CNN-based components and for the tuning of the parameters. In principle, the results could be improved by training, or fine-tuning, the models (and in particular, the CNN for Q_1 estimation) considering also non-standard quantization steps for the former compression.

G. Comparison With Anomaly-Based Forgery Detection

In this section, we discuss the performance achieved by the anomaly-based detector in [33] (MantraNet), performing forgery localization by looking for general manipulation traces, and the comparison with our method.

The average localization performance achieved by MantraNet on the dataset \mathcal{D}_{ts} and on the mismatched datasets

¹¹Notably, for single compressed regions, our estimator returns a quantization matrix of a very high compression quality ($QF > 98$). Therefore, when the method is applied for tampering localization in this scenario, the manipulation can still be exposed based on the inconsistencies between the estimated quantization steps of the background and the foreground.

TABLE XXI

AVERAGE LOCALIZATION PERFORMANCE OF THE METHODS (MCC) ON THE VARIOUS DATASETS. TPR IS REPORTED AMONG BRACKETS

	\mathcal{D}_{ts}			\mathcal{D}_{Dr}			$\mathcal{D}_{[24],A}$	$\mathcal{D}_{[24],NA}$	$\mathcal{D}_{[25]}$	\mathcal{D}'_{ts}		
	All	Type I	Type II	All	Type I	Type II				All	Type I	Type II
[33]	0.651 (0.96)	0.672	0.631	0.617(0.97)	0.637	0.595	0.411(0.37)	0.119(0.33)	0.328(0.72)	0.427(0.30)	0.474	0.341
Our	0.643(0.97)	0.652	0.633	0.631 (0.98)	0.639	0.622	0.904 (0.82)	0.906 (0.80)	0.712 (0.89)	0.624 (0.95)	0.631	0.617



Fig. 12. Examples of tampering maps from MantraNet [33].

are reported in Table XXI. We observe that our method and MantraNet can achieve comparable performance on the dataset \mathcal{D}_{ts} , with the MantraNet method having a small advantage on Type I images (A-DJPEG case).

Similar results are achieved on the \mathcal{D}_{Dr} dataset, with our method being slightly superior.

On the contrary, on the $\mathcal{D}_{[24]}$ and $\mathcal{D}_{[25]}$ dataset, the performance of MantraNet are significantly worse than those of our method. Our explanation for this result is that MantraNet benefits from the way the tampered images are generated in our datasets. In fact, the procedure of automatic generation of the tampered images in \mathcal{D}_{ts} dataset, and also in \mathcal{D}_{Dr} , leaves evident visual boundary artifacts clues in the images, with the shapes and the edges of the spliced regions being clearly visible, see the examples in Fig. 9. This is an obvious asset for a method like MantraNet that reveals the manipulation by looking at any kind of anomalies. By inspecting the localization maps we indeed see that MantraNet detects the border artifacts at the boundary of the spliced regions, see Fig. 12. On the datasets in [24] and [25] (column 8-10 in Table XXI), where such visual artifacts are not present, the performance of MantraNet are much worse, and also the detection performance are poor. To further confirm this explanation, we also tested the various methods on a dataset \mathcal{D}'_{ts} whose images are obtained as for \mathcal{D}_{ts} with the only difference that, to avoid the boundary artifacts, for a given source image, the same images compressed with different quality factors are considered as donor images (similarly to what is done in [24] and [25]). The results we achieved (Table XXI, last three columns) confirms our expectation. We can conclude that when the compression artifacts are the only traces of tampering, an anomaly-based method looking for general features like MantraNet might fail to detect and localize the tampering, thus confirming the advantage of resorting to dedicated solutions.

Last but not least, we stress that MantraNet can only localize the forgery, without distinguishing spliced regions from different donor images, that is, without performing attribution, which is a central goal of our method.

VII. CONCLUSION

We have proposed an end-to-end system capable to detect and localize image splicing operations by additionally distinguishing spliced regions derived from different donor images (a task referred to as spliced regions attribution). The basic assumption behind the proposed system is that the spliced and background regions have been double JPEG-compressed, but

the quantization matrices used for the first compression are different for the background and spliced regions stemming from different sources. Estimating the primary quantization matrix and clustering image blocks according to the result of the estimation provides the basic mechanism underlying the detection, localization and attribution of the spliced regions. We instantiated the proposed system by adopting state-of-the-art solutions for each step the system consists of, including estimation of the primary quantization matrix, estimation of the number of clusters, clustering and spatial refinement of the tampering map. The good performance of the resulting system have been proven by means of extensive experiments and compared with those of two baseline methods operating in similar conditions. The main strength of the proposed system is that it can be applied to a wide variety of situation, concerning the alignment of the double compression steps and the quantization steps used for the first and the second compression. Moreover, to the best of our knowledge, this is the first method that can distinguish multiple tampered regions taken from different donor images based on the analysis of JPEG traces.

It goes without saying that any improvement of each of the steps the system consists of will result in a consequent improvement of the overall accuracy of the system. From this point of view, improving the accuracy of the estimation of the primary quantization matrix would play a crucial role, since the entire system relies on the accuracy of such a step. Clustering is also an area where improvements are possible, both with regard to the estimation of the number of clusters and the subsequent clustering process. In particular, finding better ways to fuse spatial information with the information provided by the estimated quantisation matrix could significantly improve the accuracy of the system.

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 DARPA and AFRL or the U.S. Government.

REFERENCES

- [1] T. Pevný and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 247–258, Jun. 2008.
- [2] B. Li, Y. Q. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *Proc. IEEE 10th Workshop Multimedia Signal Process. (MMSP)*, Oct. 2008, pp. 730–735.
- [3] I. Amerini, R. Becarelli, R. Caldelli, and A. D. Mastio, "Splicing forgeries localization through the use of first digit features," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2014, pp. 143–148.

- [4] C. Deng, Z. Li, X. Gao, and D. Tao, "Deep multi-scale discriminative networks for double JPEG compression forensics," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–20, Feb. 2019.
- [5] Y.-L. Chen and C.-T. Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 396–406, Jun. 2011.
- [6] N. Yakun, B. Tondi, Y. Zhao, and M. Barni, "Primary quantization matrix estimation of double compressed JPEG images via CNN," *IEEE Signal Process. Lett.*, vol. 27, pp. 191–195, Dec. 2019.
- [7] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
- [8] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 529–538, Sep. 2008.
- [9] J. Bunk *et al.*, "Detection and localization of image forgeries using resampling features and deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1881–1889.
- [10] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "Guided filtering for PRNU-based localization of small-size image forgeries," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 6231–6235.
- [11] P. Korus and J. Huang, "Multi-scale analysis strategies in PRNU-based tampering localization," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 809–824, Apr. 2017.
- [12] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proc. 6th Int. Workshop Inf. Hiding*, 2004, pp. 128–147.
- [13] C. Pasquini, G. Boato, and F. Perez-Gonzalez, "Multiple JPEG compression detection by means of benford-Fourier coefficients," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2014, pp. 113–118.
- [14] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 2, Apr. 2007, pp. 217–220.
- [15] Z. Qu, W. Luo, and J. Huang, "A convolutive mixing model for shifted double JPEG compression with application to passive image authentication," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 1661–1664.
- [16] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 842–848, Apr. 2012.
- [17] Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognit.*, vol. 42, no. 11, pp. 2492–2501, 2009.
- [18] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 2444–2447.
- [19] Q. Wang and R. Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP J. Inf. Secur.*, vol. 2016, no. 1, pp. 1–12, Dec. 2016.
- [20] M. Barni *et al.*, "Aligned and non-aligned double JPEG detection using convolutional neural networks," *J. Vis. Commun. Image Represent.*, vol. 49, pp. 153–163, Nov. 2017.
- [21] M.-J. Kwon, I.-J. Yu, S.-H. Nam, and H.-K. Lee, "CAT-net: Compression artifact tracing network for detection and localization of image splicing," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 375–384.
- [22] J. Lukáš and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Proc. Digital Forensic Res. Workshop*, 2003, pp. 1–17.
- [23] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 154–160, Mar. 2009.
- [24] 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.
- [25] W. Wang, J. Dong, and T. Tan, "Exploring DCT coefficient quantization effects for local tampering detection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1653–1666, Oct. 2014.
- [26] Z. Dias, A. Rocha, and S. Goldenstein, "First steps toward image phylogeny," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2010, pp. 1–6.
- [27] Z. Dias, A. Rocha, and S. Goldenstein, "Image phylogeny by minimal spanning trees," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 774–788, Apr. 2012.
- [28] A. A. de Oliveira *et al.*, "Multiple parenting phylogeny relationships in digital images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 2, pp. 328–343, Feb. 2016.
- [29] Y. Wu, W. Abd-Almageed, and P. Natarajan, "Busternet: Detecting copy-move image forgery with source/target localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 168–184.
- [30] M. Barni, Q.-T. Phan, and B. Tondi, "Copy move source-target disambiguation through multi-branch CNNs," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1825–1840, 2021.
- [31] A. Islam, C. Long, A. Basharat, and A. Hoogs, "DOA-GAN: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4676–4685.
- [32] 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.
- [33] 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.
- [34] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection," *IEEE Access*, vol. 8, pp. 133488–133502, 2020.
- [35] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. New York, NY, USA: Springer, 1992.
- [36] I. Amerini, R. Caldelli, P. Crescenzi, A. D. Mastio, and A. Marino, "Blind image clustering based on the Normalized Cuts criterion for camera identification," *Signal Process. Image Commun.*, vol. 29, no. 8, pp. 831–843, 2014.
- [37] Y. Li, X. Zhang, X. Li, Y. Zhang, J. Yang, and Q. He, "Mobile phone clustering from speech recordings using deep representation and spectral clustering," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 965–977, Apr. 2018.
- [38] O. Mayer and M. C. Stamm, "Exposing fake images with forensic similarity graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 5, pp. 1049–1064, Aug. 2020.
- [39] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*, vol. 84. New York, NY, USA: Marcel Dekker, 1988.
- [40] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.
- [41] Y. El-Sonbaty and M. A. Ismail, "Fuzzy clustering for symbolic data," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 2, pp. 195–204, May 1998.
- [42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [43] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [44] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. et Biophys. Acta (BBA)-Protein Struct.*, vol. 405, no. 2, pp. 442–451, 1975.
- [45] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Reading, MA, USA: Addison-Wesley, 2005.
- [46] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proc. 6th ACM Multimedia Syst. Conf.*, Mar. 2015, pp. 219–224, doi: [10.1145/2713168.2713194](https://doi.org/10.1145/2713168.2713194).
- [47] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," *Proc. SPIE*, vol. 5307, pp. 472–480, Dec. 2003.
- [48] T. Gloe and R. Böhme, "The 'Dresden image database' for benchmarking digital image forensics," in *Proc. ACM Symp. Appl. Comput. (SAC)*, New York, NY, USA, 2010, pp. 1584–1590, doi: [10.1145/1774088.1774427](https://doi.org/10.1145/1774088.1774427).