

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Morphological Filter Detector for Image Forensics Applications

GIULIA BOATO¹ (Member, IEEE), DUC-TIEN DANG-NGUYEN² (Member, IEEE) and FRANCESCO G.B. DE NATALE¹ (Senior Member, IEEE)

¹University of Trento, Italy (e-mail: giulia.boato@unitn.it; francesco.denatale@unitn.it)

²University of Bergen, Norway (e-mail: ductien.dangnguyen@uib.no)

Corresponding author: Giulia Boato (e-mail: giulia.boato@unitn.it).

ABSTRACT Mathematical morphology provides a large set of powerful non-linear image operators, widely used for feature extraction, noise removal or image enhancement. Although morphological filters might be used to remove artifacts produced by image manipulations, both on binary and graylevel documents, little effort has been spent towards their forensic identification. In this paper we propose a non-trivial extension of a deterministic approach originally detecting erosion and dilation of binary images. The proposed approach operates on grayscale images and is robust to image compression and other typical attacks. When the image is attacked the method loses its deterministic nature and uses a properly trained SVM classifier, using the original detector as a feature extractor. Extensive tests demonstrate that the proposed method guarantees very high accuracy in filtering detection, providing 100% accuracy in discriminating the presence and the type of morphological filter in raw images of three different datasets. The achieved accuracy is also good after JPEG compression, equal or above 76.8% on all datasets for quality factors above 80. The proposed approach is also able to determine the adopted structuring element for moderate compression factors. Finally, it is robust against noise addition and it can distinguish morphological filter from other filters.

INDEX TERMS Digital Image Forensics, Media Authentication, Morphological Filter Detection

I. INTRODUCTION

In the last decade, researchers and practitioners in multimedia forensics have been developing a substantial body of knowledge and techniques targeted to the authentication of multimedia objects and their processing history recovery [1]–[5]. A recent trend tries to define universal detectors able to reveal manipulations independently from the type of processing applied, which could leverage media authentication in applications like journalism or social media analysis [6]. On the other hand, many methods have been proposed to detect different types of forgeries, which is very relevant for diverse applications. In particular, first it is crucial in digital investigations, given that images, audio tracks and video sequences now play a crucial role where they often represent digital evidences to the court [7]. Secondly in multimedia data phylogeny, which aims at recovering and tracing back the life-cycle of an image or a video [8]–[11].

This broad class of specific manipulation detectors includes the identification of pasted regions [12]–[16], resizing [17], [18], re-compression [19], image enhancing [20], in-

consistencies in the geometry and illumination of the image due to possible manipulations [21]–[23], and various types of non-linear filtering (especially median) [24]–[35].

In the context of non-linear filtering detection very little attention was given to morphological filters [36] often used in image processing for artifacts removal and image enhancement [37] [38]. The detection of this kind of filtering is of interest in the context of both image phylogeny and specific tampering identification in legal scenarios, but could be very useful also to detect possible counter forensic attacks based on morphology, where such filters, very powerful in the removal of local noise, could be exploited at the end of the image manipulation process to cover other types of traces.

In this paper we present a non trivial extension of a recent work [39] which introduced a deterministic detector of erosion and dilation in binary images. The proposed extension works on grayscale images by detecting morphological filters application in an accurate way both in uncompressed and compressed images. The method also allows for erosion versus dilation discrimination and in many cases also for the

adopted structural element identification. Robustness against JPEG compression, noise addition and confusion with other types of filters is also tested on various datasets.

The rest of the paper is organized as follows: Section II provides the theoretical background for the problem formulation; Section III describes the proposed methodology for morphological filtering detection in uncompressed and compressed images; Section IV describes the experimental setup, datasets and scenarios adopted for testing and validation; Section V details the experimental analysis and obtained results; finally, Section VI reports some concluding remarks.

II. THEORETICAL FORMULATION

In this section we introduce the mathematical formulation of the problem, and we derive the proposed methodology for morphological filter detection.

A. MATHEMATICAL FORMULATION AND PROPERTIES

Mathematical morphology defines a set of nonlinear filters commonly employed in digital image processing to modify the local structural content of images. All the morphological filters are derived from the various combinations of two basic operators, *erosion* and *dilation*, and a kernel mask (or shortly, a kernel) called *structuring element*, characterized by a shape, a size, and a reference point. The shape and the size of the kernel are responsible for the behavior of the operator on the image, while the reference point just defines the shift of the filtered image with respect to the original. The invention of such mathematical tools dates back to 1964 [36], and was meant to the filtering of binary images for mineral studies. Later studies [40] led to the generalization of the theory to the case of grayscale images.

According to this theory, given a grayscale image $f(x, y)$ and a binary structuring element B , the two fundamental morphological operators, erosion and dilation, are respectively defined as:

$$f \ominus B = \min_B(f(x, y) \cap B_{xy}) \quad (1)$$

$$f \oplus B = \max_B(f(x, y) \cap B_{xy}) \quad (2)$$

where B_{xy} represents the structuring element (kernel) B with the reference point centered at the coordinates x, y of the image plane, while the intersection operations returns the subset of the image pixels overlapped with the 1s of B . In this respect, the basic grayscale operators are particular cases of rank-order filters, and behave very similarly to min-max (see examples in Figure 1) and median operators, except for the shape of the mask.

As in binary morphology, the composition of erosion and dilation allows defining more complex filters, among which the most common are the *open* and *close* operators, respectively defined as follows:

$$(f \circ B) = (f \ominus B) \oplus B \quad (3)$$

$$(f \bullet B) = (f \oplus B) \ominus B \quad (4)$$

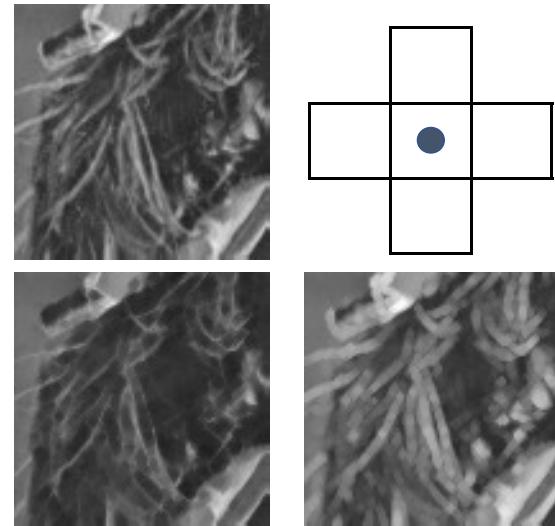


FIGURE 1: Example of grayscale erosion (bottom left) and dilation (bottom right) of an image detail (top left), using a cross-shaped structuring element (top right). The resulting patches show the effect of min and max operations: erosion produces a darker version of the original image eliminating small cross-shaped details, while dilation produces the opposite effect.

Also the mathematical properties of morphological grayscale operators match the ones of the corresponding binary operators. Consequently, the theoretical background of [39] remains valid also for in the grayscale domain. In particular, the following properties are exploited in the construction of the proposed detector:

- (i) Translation invariance: the position of the reference point only affects the translation of the filtered image
- (ii) Dilation commutativity: $A \oplus B = B \oplus A$
- (iii) Associativity: a cascade of erosions (dilations) is equal to the erosion (dilation) with a mask generated by dilating each other the original masks (*A, B and C* are **binary structuring elements**)

$$A \ominus B \ominus C = A \ominus (B \oplus C) \quad (5)$$

$$A \oplus B \oplus C = A \oplus (B \oplus C) \quad (6)$$

- (iv) Open and close idempotence: iterating open and close with the same structural element does not produce additional changes in the image

$$A \circ B \circ B = A \circ B \quad (7)$$

$$A \bullet B \bullet B = A \bullet B \quad (8)$$

Additionally, it is easy to see that the two theorems introduced in [39] remain valid, since their demonstrations do not depend on Eq. 1 and Eq. 2, but only on the properties (i)-(iv), which hold also for grayscale images.

Theorem 1: Let $I' = I \ominus K$, then $I' \bullet K = I'$. Respectively, if $I' = I \oplus K$, then $I' \circ K = I'$.

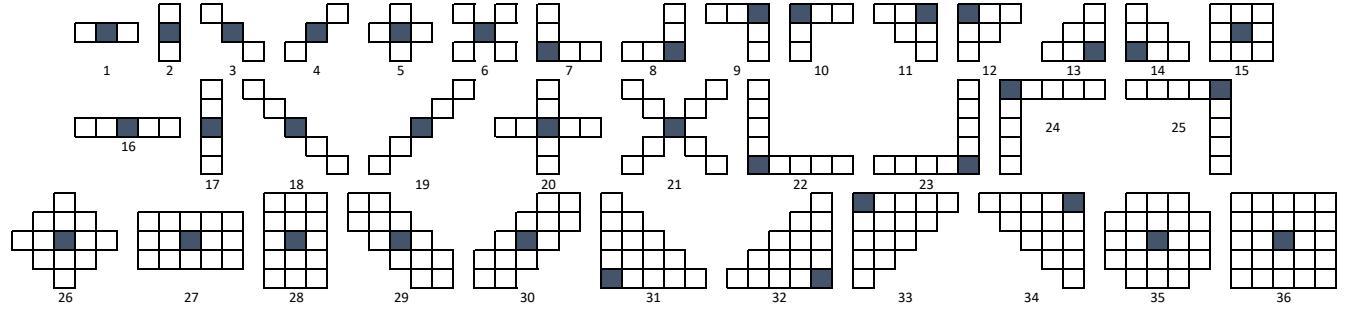


FIGURE 2: Set of structuring elements (kernels) used for simulations. In dark, structural element's reference point.

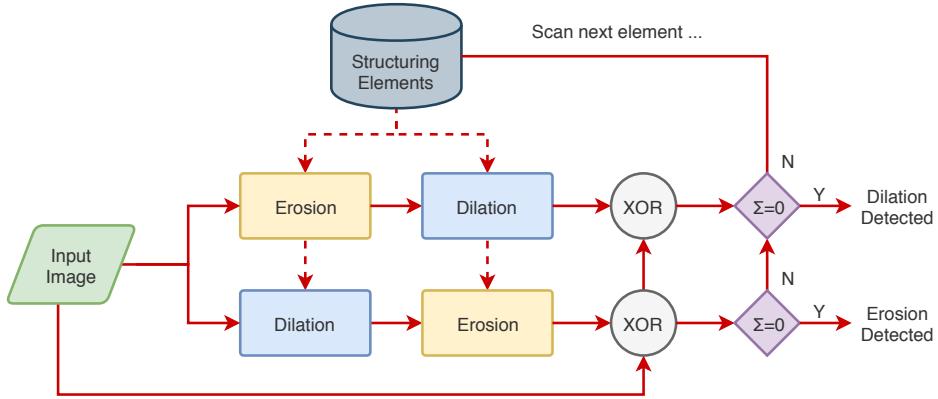


FIGURE 3: Proposed detection scheme for raw grayscale images. After opening/closing with a kernel K , if the image does not change, this means that it was previously dilated/eroded with that kernel (details in Section II.A). By applying the procedure for a set of possible masks, the detector is thus able to reveal a perfect match between the input eroded/dilated image and the corresponding opened/closed version.

Theorem 2: Let $I' = I \ominus K$, then $\forall M$ such that $\exists E | M \oplus E = K$, we have that $I' \bullet M = I'$. Respectively, if $I' = I \oplus K$, then $I' \circ M = I'$.

Theorem 1 can be equivalently formulated in terms of series of erosion and dilation operators, according to the definition of open and close operators. Theorem 2 extends the equality of Theorem 1 to any kernel mask M that can produce K by dilation with an appropriate kernel E .

As an immediate consequence of the above theorems, an image dilated (eroded) with a given kernel K , will remain unchanged after applying an open (close) operator with the same element. This provides a simple test to detect a filtered image: apply an open (close) operator with a kernel K , if the image does not change, this means that it was previously dilated (eroded) with that kernel, otherwise it was not. The detection consists then in subsequently checking with the above procedure a set of possible masks. In [39], a set of common kernels were proposed, characterized by some level of symmetry (see Figure 2).

III. PROPOSED APPROACH

In this section we introduce the proposed detector, distinguishing two cases: the detection of morphological filtering on raw images, and the detection in the presence of post-

processing (e.g. compression, noise addition, filtering). We will see that the former is a trivial extension of the binary case but has very limited applicability, while the latter requires further attention.

A. FILTER DETECTION ON RAW IMAGES

According to the theory stated in Section II, in the absence of post-processing grayscale morphological operators can be easily and deterministically detected by applying the schema proposed in [39] and reported in Figure 3.

However, it is worth mentioning that this scenario is rarely verified in grayscale images, which are typically stored in compressed format after filtering. The compression (as well as most other image processing operations) modifies the image, thus hindering the applicability of Theorem 1 and thus the applicability of the deterministic method depicted in Figure 3. Indeed, the detector will never report a perfect match between the input eroded (dilated) image and the corresponding opened (closed) version.

In the next section, we propose an extension of the above approach that is able to deal with such scenario. The new algorithm will exploit the traces left by the morphological filters even after post-processing, and will rely on a statistical analysis, thus loosing its deterministic nature. We will see

however that it is sufficiently robust to many attacks, and in particular to the JPEG compression.

B. FILTER DETECTION ON POST-PROCESSED IMAGES

As mentioned in the previous section, any further processing on the filtered image will modify the pixel values, possibly re-introducing structures that were eliminated by the morphological operator. A further morphological filtering will then produce a non-null effect on the image, which will be revealed by the detector. It is therefore necessary to verify if some traces of the original filter survive the post-processing.

Recalling that the grayscale morphology operates as a local min-max filter, we can expect that it will produce larger variations in the presence of high-contrast structures that match the structural element geometry. On the contrary, unless the post-processing is meant to be extremely visible, it will introduce small gray level variations on the image. Consequently, although we will not have a null difference image at the detector output, we can expect that the local differences will be much smaller for a filtered image than for an original one. An experimental evidence of this fact is provided in Figure 4. Here we plot the log-scale histograms of the absolute differences before and after the application of an open operator to a JPEG compressed unfiltered image, and to the relevant JPEG compressed dilated image (the two right most columns). Both open and dilation operators have been applied using the kernel mask 35 in Figure 2, and the JPEG quality factor has been set to $QF = 95$. It is possible to observe that the histogram referred to the dilated image decreases steeply, with significant bins only for low values of the difference, while the histogram referred to the unfiltered image shows a long tail with significant values also above 50. This means on one side that the compression does not affect the high-contrast structures present in the original image (which are then removed by the following open operator), and on the other side that it does not re-introduce in the filtered image any high-contrast structure sensitive to the filter itself. Similar results on the eroded version and the close operator are shown on the two left most columns.

On this basis, we propose to modify the detection scheme as shown in Figure 5. The core of the procedure still involves the application of grayscale opening/closing. In this case, however, we take into consideration the statistical properties of the differences between the input and output images, to see how such residual is distributed. Therefore, we calculate the histogram of the difference image (processing block HIST in Figure 5) and feed it into a statistical classifier to perform the decision. As far as the classifier is concerned, we adopted a properly trained classifier.

Finally, we should notice that not all the areas of an image are equally affected by a morphological filter. In particular, the filter has a negligible effect on flat areas, thus possibly jeopardizing the results of the detector. To avoid this problem we decided to limit the analysis to the image regions that contain significant textures or edges. To this purpose, we calculate the block-wise normalized local variance α , i.e.,

for each block, we normalise all the pixel values to $[0, 1]$ and then compute the standard variance, and we restrict the computation of the histogram to the blocks with $\alpha > \alpha_{th}$, where α_{th} has been empirically set to 0.15. This task is performed by the first processing block in Figure 5.

IV. EXPERIMENTAL SETUP

In order to assess the performance of the proposed detector, we tested it in various scenarios and we evaluated it in terms of accuracy. In this section we describe such scenarios and the relevant experiments. Furthermore, we introduce the datasets used for the testing, and we provide additional details the relevant training procedure.

A. DATASETS

Three publicly available datasets have been used in the experiments:

- The Uncompressed Colour Image Database (UCID) [41], built with the purpose of providing a standard set for performance assessments in image retrieval and compression. The dataset consists of 1,338 uncompressed color images, with fixed sizes of 512×384 or 384×512 pixels in uncompressed TIFF format.
- The Dresden Image Database (DRESDEN) [42], originally created for evaluation of forensic techniques related to camera-based information. From their public web-interface, we selected the complete set of RAW images (1,189 uncompressed images, all with fixed size of 3008×2000 pixels).
- The Raw Images Dataset (RAISE) [43], consisting of 8156 raw images with resolutions ranging from $3,008 \times 2,000$ to $4,928 \times 3,264$ pixels. Authors provide also smaller subsets, among which we selected the one containing 1,000 images (RAISE-1k).

The three datasets were selected to diversify the range of resolutions in the experimental tests. In order to evaluate our proposed schema, all the images were converted to grayscale with a depth of 8 bits.

B. TESTING SCENARIOS

The proposed detector has been tested in various practical scenarios:

- The first set of tests were performed in order to understand the impact of the parameters as well as to select a proper classifier. Based on the setups found on this set of tests, we kept the best setups (in terms of performance) for the next tests.
- The second set of tests refers to the detection of morphological filtering. In this case, we want to establish the accuracy of the detector in discriminating filtered vs. pristine (unfiltered) images, in classifying the type of operator (erosion vs. dilation), and finally in determining the exact structuring element used for filtering, both in uncompressed and JPEG compressed images with various quality factors (from $QF = 100$ indicating

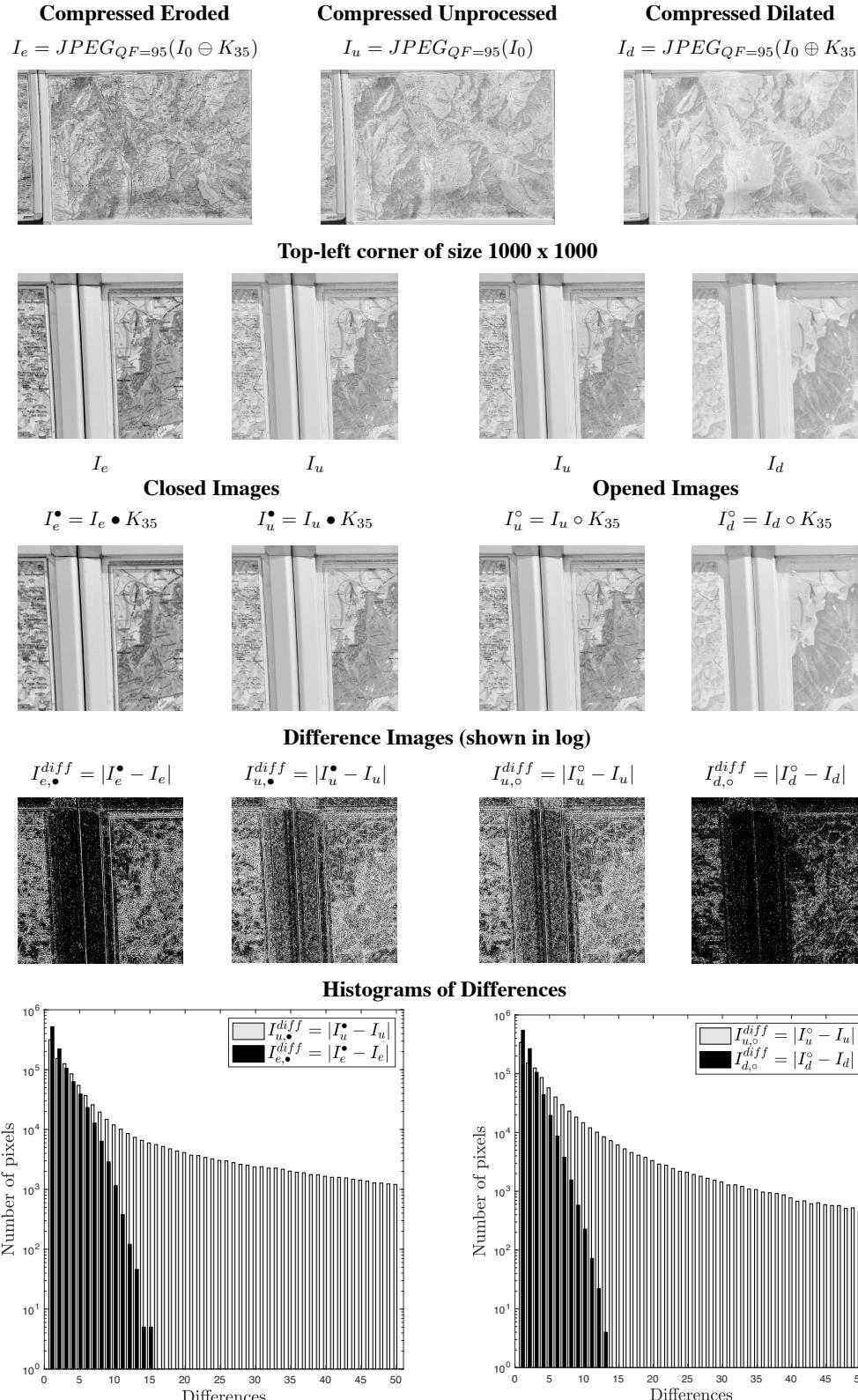


FIGURE 4: Histograms of differences before and after application of the close and open operators to an unprocessed image (grayscale version of ‘r3ba1827ft.TIF’ from RAISE-1k) and its eroded and dilated versions, respectively. Both images have been JPEG compressed with $QF = 95$. Kernel mask 35 and the analysed windows size of 1000×1000 are used. The chart shows the first 50 bins in logarithmic scale.

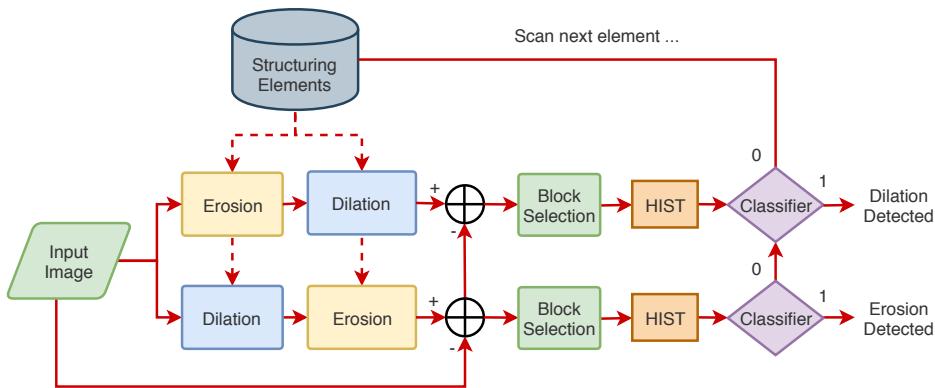


FIGURE 5: Proposed detection scheme for attacked grayscale images. After opening/closing, we select image regions that contain significant textures or edges and there we calculate the histogram of the differences between the input and output images. Such histograms are fed into a statistical classifier to perform the decision (details in Section III.B).

highest quality images, down to $QF = 70$ which still corresponds to compressed images of high quality).

- The third set of tests concerns robustness against noise. The test image is contaminated with random noise and then compressed, and we evaluate the accuracy of the detector in determining the presence and type of filtering and the structuring element used.
- Finally, we want to determine the capability of our approach in distinguishing between morphological filters and other filters that produce similar results. In particular, we considered Gaussian lowpass and median filtering. In both cases, the filtered images are compressed and passed to the detector to reveal possible false alarms.

It is worth noting that, due to the properties analysed in Section II, in the presence of a cascade of different basic operators (erosion and dilation), the detector will reveal the last operator applied. Accordingly, when processing an opened or a closed image, the detector will reveal the last dilation or erosion, respectively. Furthermore, in the presence of a cascade of the same basic operator, the detector will reveal a single erosion or dilation with the composed structuring element. Therefore, in the experimental section we will just consider erosion and dilation detection, even if the image may have been potentially processed with more complex combinations of filters.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, all experimental tests carried out are described. All tests are performed using Python 3.6 and libSVM on a standard machine (Macbook Pro 2016 2,3GHz 4-kernel Intel Core i5, 8GB ram).

A. THE SENSIBILITY AND THE CHOICE OF THE CLASSIFIERS

In order to understand the impact of the parameters as well as to select a proper classifier, we ran some experiments on a subset of images. After that, we empirically determined

the values for these parameters. Table 1 summarizes these parameters, their meaning and how their values were decided.

The classifier receives in input the histogram of the differences between the input image and the relevant opened (closed) image and returns a binary decision. An important aspect is the training of the classifier. In fact, there is a clear dependency of the image statistics on the level of compression applied, which is reflected on the characteristics of the histogram. As an example, Figure 6 (a) shows the same situation of Figure 4 at a lower quality factor. It can be observed that the two distributions are still well separable, but the histogram related to the filtered image shows a longer tail, due to the larger artifacts introduced by the compression.

Accordingly, we decided to train a set of classifiers for varying JPEG quality factors, from 100 to 70. During the test phase, we tested with different classifiers: k-NN, decision trees, naive Bayes classifier, and SVM, and empirically decided to use SVM with the radial basis function (RBF) kernel.

We also note that given the appropriate quality factor, the histograms of differences show a peculiar behavior only if the detector parameters match the input filter in both the operator (erosion or dilation) and the structural element (kernel), whereas such behavior is never found in all the other combinations. It is important to notice that the application of the detector to an unfiltered image or to an image filtered with a different combination of operator and/or kernel produces similar results, i.e., only the matching detector responds to the filtered image. As an example, in Figure 6 (b) we show the histograms deriving from the application of an erosion detector to an eroded and a dilated image, both with the same kernel. It can be observed that the dilated image responds similarly to the unfiltered image in Figure 6 (a). Analogously, in Figure 6 (c), we compare the histograms deriving from the erosion detector applied to two eroded images, one with the same kernel and the other with a different one. Also in this case, the image filtered with a different kernel

TABLE 1: Parameters selection.

Parameter/ Classifier	Meaning	Range	Empirical Value/Selection
Image resolution	The size of the analyzed region of interest	384×512 to 4928×3264	Good results for resolution from 1000×1000 .
Block Selection Threshold α_{th} and Block Size s	Erosion and dilation have a negligible effect on flat areas, thus these two parameters allow the method to effectively work on non-flat blocks only.	$\alpha_{th} \geq 0, \alpha = 0$ means all pixels in the block has the same values. $s \in \{3 \times 3, 5 \times 5, \dots\}$	$\alpha_{th} = 0.15$ and $s = 3 \times 3$.
n	Number of bins of the analyzed histograms.	$0 \leq n \leq 256$	Even if for $n > 30$ we already achieve 90% accuracy, we set n at the highest value, i.e., considering all bins.
The classifiers		SVM, KNN, Linear Regression, and others	SVM with RBF kernel and grid search provides the best performance.

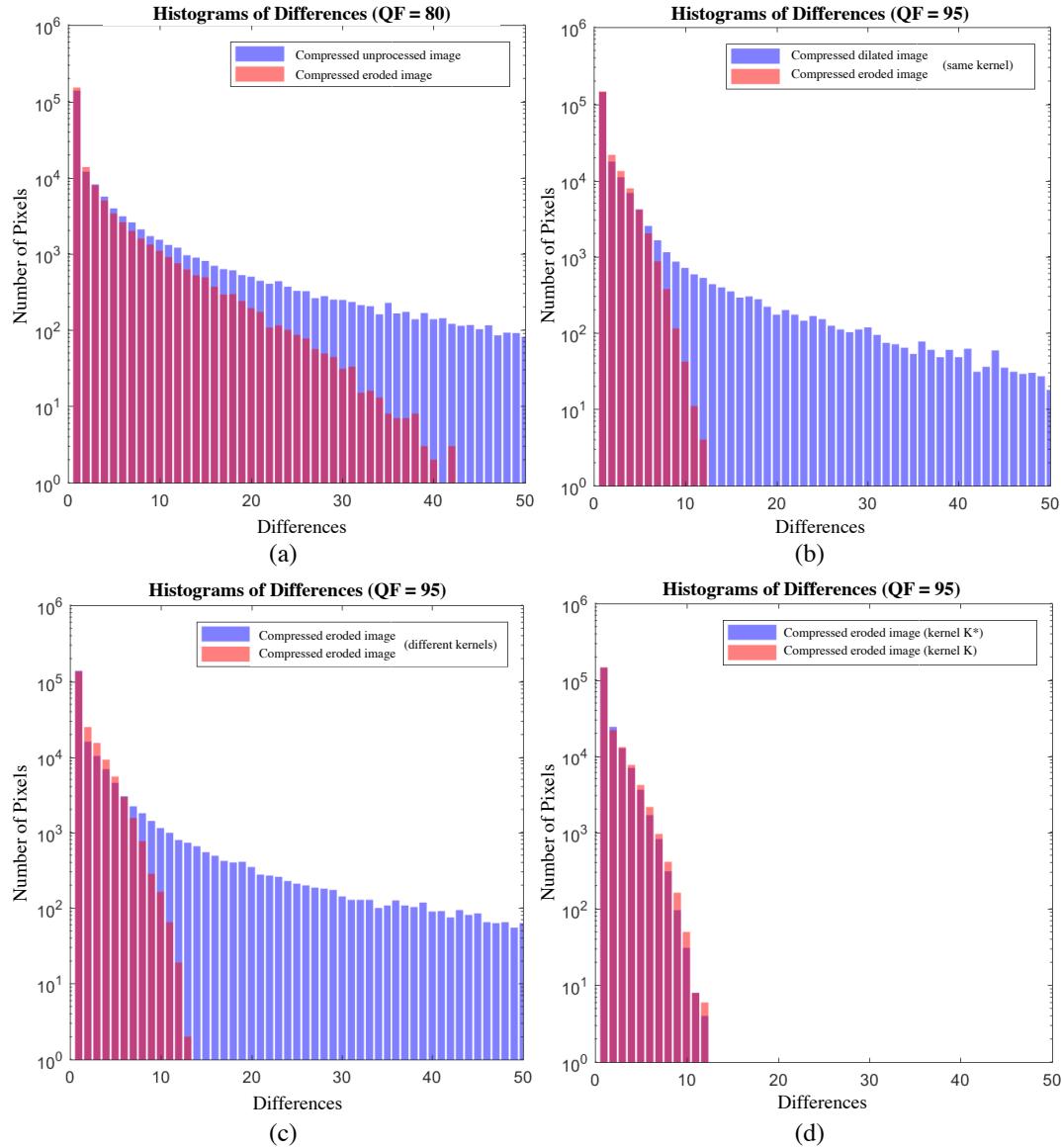


FIGURE 6: Analysis of different cases in erosion detection. (a) Histograms resulting from original image and its dilated version, after compression with $QF = 80$; (b) Histograms resulting from dilated and eroded images produced with the same kernel, after compression at $QF = 95$; (c) Histograms resulting from eroded images produced with unrelated kernels, after compression at $QF = 95$; and (d) Histograms resulting from eroded images produced with kernel K and its dilated version K^* , after compression at $QF = 95$

TABLE 2: Morphological filtering detection results on all three datasets using SVM with RBF kernel, 10-fold cross validation. The numbers are in percentage and each of them is the average performance (for distinguishing between pristine versus filtered images) over all 36 kernels. Detailed results are reported in Table 3.

Dataset	Operator	QF = 100	95	90	85	80	75	70
UCID	Erosion	100	90.86	82.62	79.63	77.45	67.41	64.42
	Dilation	100	90.16	82.86	79.95	76.80	68.77	64.94
Dresden	Erosion	100	97.56	96.94	95.20	88.73	79.85	78.34
	Dilation	100	97.93	96.82	96.11	88.74	81.08	75.08
Raise	Erosion	100	98.66	96.25	94.82	89.00	80.42	78.55
	Dilation	100	97.98	97.43	93.72	91.02	80.49	78.53

responds as unfiltered. Finally, in Figure 6 (d) we show the case of application of the erosion detector to two images filtered with kernels belonging to the same group (i.e., one can be obtained from the other by dilation). As expected from Theorem 2, the two histograms are almost overlapped, since both show the statistical properties of a filtered image.

For some other choices, for example the block size or α_{th} , all values are empirically selected and are reported in Table 1.

B. DISTINGUISHING BETWEEN PRISTINE VERSUS MORPHOLOGICAL FILTERED IMAGES

1) Experimental Results on Raw Images

We ran a test on uncompressed images in order to confirm the deterministic nature of the proposed approach on raw images. Since the detection strategy in this case is the very same proposed in [39], we followed their general approach for deriving results relative to raw grayscale images. Each image has been processed with erosion and dilation operators considering all the 36 kernels. All images, along with their unprocessed versions, were fed to both dilation and erosion detectors. For each image, all 36 kernels are tested, returning either the largest kernel with perfect match between input and output or no detection if all kernels fail. All three datasets result in 100% accuracy in discriminating the presence and the type of morphological filter.

2) Experimental Results on JPEG Compressed Images

In order to test the approach on JPEG compressed images, we considered a set of seven different quality factors with $QF \in \{100, 95, 90, 85, 80, 75, 70\}$, with 14 (7×2) binary classifiers, respectively, using a Gaussian kernel with grid-search for the parameters. We apply k-Fold validation with $k = 10$. Shown in Table 2 are the average results (over all kernels) of the proposed method on three datasets at the seven quality factors. The detailed results for all kernels and both morphological operators on the three datasets are reported in Table 3. According to the results, we can observe that the proposed method can provide high detection performance on low compression level (i.e., quality factor $QF \geq 80$) with accuracy equal or above 76.8% on all datasets. At stronger compression levels, the performance reduces significantly. The resolution also plays an important role as it can be grasped from the results. Indeed, UCID has lower performance with respect to Dresden and Raise. It is also interesting to notice that there is little difference between Dresden and

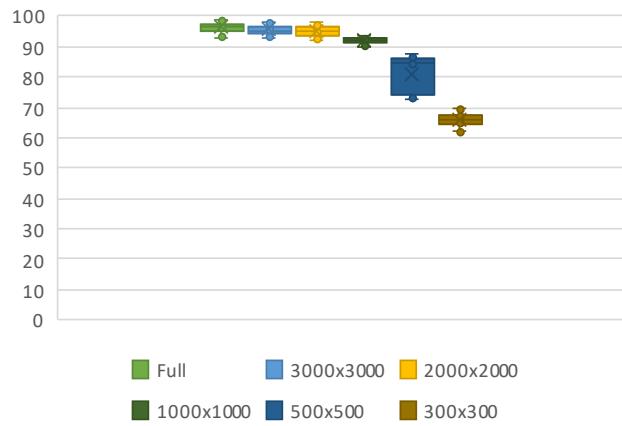


FIGURE 7: Precision (computed as percentage) on erosion detection under different resolutions. Dataset: Raise, QF = 90. All 36 kernels are used, results are shown in the error bars.

Raise. To have a deeper understanding of the impact of the image resolution, we ran another test on different image resolutions: the proposed method achieves good performance (over 90%) for resolution from about $1,000 \times 1,000$. Shown in Figure 7 are results over 36 kernels on erosion detection for RAISE-1k images.

C. DETECTOR ROBUSTNESS ANALYSIS

The second test in this set is to analyse the impact of different kernels. We applied all kernel masks and then tried to detect them. Shown in Figure 8 is the confusion matrix of the erosion detection (similar results were obtained on dilation detection). According to the results, the detector was able to recognize small size kernels (kernels from 1 to 14), or larger kernels in special shape (kernels 20 to 25), but poorly classified the others. This is understandable as a consequence of Theorem 2.

The third set of tests concerns analysis of robustness against noise addition. In the first test, morphological filtering is applied to each image and then this is contaminated with a noise. Five different attacks - pepper & salt noise, Gaussian filtering with window size 3×3 , median filtering with window size 3×3 , scaling at $1.1x$ and scaling at $0.9x$ - were applied and average results (on all kernels) for

TABLE 3: Detailed results on erosion and dilation detection on UCID, Dresden and RAISE-1k datasets. K indicates the kernel mask used (from 1 to 36). Average results are summarised in Table 2.

	QF	K = 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
UCID Erosion	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	95	90.42	90.89	90.34	89.61	90.15	90.56	91.2	92.2	92.69	92.89	93.09	93.48	94.47	93.59	94.4	88.41	88.49	88.78
	90	83.42	82.73	82.57	83.06	83.14	82.49	82.67	82.62	83.59	84.08	83.41	83.63	83.62	84.32	84.02	82.22	83.18	83.72
	85	79.06	79.7	80.69	80.74	80.91	81.42	81.03	80.17	80	80.65	79.88	80.87	80.73	80.25	79.3	80.45	81.17	81.99
	80	75.83	76.23	77.19	76.99	77.62	78.42	78.66	78.19	79.1	79.67	80.26	80.08	79.93	79.59	80.32	76.24	76.4	76.76
	75	64.25	65.2	66.06	65.31	65.09	65.55	66.22	65.24	65.9	65.78	65.59	64.86	64.35	64.49	65.07	69.13	69.07	69.08
	70	63.34	64.23	64.91	63.92	62.95	63.56	64.53	65.45	64.87	64.5	65.28	65.41	64.61	65.6	64.72	62.8	63.66	63.69
UCID Dilation	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	95	89.24	88.4	88.1	88.11	89.09	89.99	89.28	89.9	90.46	91.03	90.76	91.28	91.44	90.49	91.33	90.05	90.66	90.5
	90	83.18	82.29	81.66	82.34	82.67	82.85	82.35	83.04	83.42	82.55	83.13	83.21	83.85	83.33	84.22	82.22	82.24	82.21
	85	80.05	81.01	81.54	81.51	82.35	82.43	83.15	82.9	82.3	81.76	80.9	81.69	81.01	81.7	82.61	81.15	81.36	81.39
	80	75.29	76.16	75.77	75.33	74.89	75.42	75.81	75.06	75.99	75	75	75.84	76.02	76.56	75.65	77.25	77.96	78.31
	75	65.35	65.08	65.49	65.17	65.95	65.94	65.53	66.12	66.64	67.59	67.69	67.9	68.6	68.58	69.13	70.23	70.63	70.22
	70	63.34	63.93	64.77	65.4	66.36	67.35	67.89	68.67	67.95	68.91	68.3	67.33	66.94	66.35	66.05	62.8	62.47	63.29
Dresden Erosion	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	95	99.03	99.04	98.32	97.41	97.12	97.74	98.66	98.18	97.72	96.76	95.9	96.28	96.15	97	96.34	98.95	98.2	97.58
	90	98.85	97.91	97.83	97.41	96.92	97.28	97.74	97.22	96.58	96.45	95.9	95.93	96.15	96.73	96.34	97.93	97.8	96.91
	85	97.01	97.24	97.67	97.41	96.75	95.78	95.15	94.63	94.76	93.76	94.61	95.17	94.33	95.24	95.8	96.52	97.01	96.81
	80	87.43	88.17	89.16	88.31	88.29	87.49	87.41	87.22	87.66	87.88	88.45	88.11	87.41	86.53	86.72	89.32	89.85	90.15
	75	78.63	78.3	78.64	79.59	78.84	79.73	79.33	78.57	79.42	78.93	78.02	78.05	78.76	78.35	79.28	80.03	79.32	79.56
	70	76.19	76.7	76.22	76.2	76.3	76.66	77.39	77.18	76.73	77.3	78.02	77.9	78.35	78.02	79.02	82.23	81.79	82.78
Dresden Dilation	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	95	98.32	97.86	98.41	97.59	96.9	97.26	96.35	95.9	96.47	97.34	96.93	97.7	97.37	98.35	97.95	98.83	99.81	99.26
	90	98.32	97.86	98.16	97.59	96.9	97.26	96.35	95.9	96.18	97.14	96.93	96.52	96.36	95.99	95.26	97.58	98	98.06
	85	96.74	96.67	97.06	96.72	96.5	96.03	96.35	95.87	96.18	96.73	96.75	96.39	95.62	95.99	95.26	96.86	97.47	98.06
	80	87.78	88.29	89.27	88.79	88.14	88.85	88.18	88.43	87.66	87.65	87.2	87.23	87.53	86.79	87.78	90.24	89.36	88.43
	75	79.34	79.2	79.79	80.75	80.9	80.69	79.84	79.66	80.51	79.95	80.72	81.69	81.27	80.45	79.62	80.21	80.72	81.13
	70	73.81	74.7	74.9	73.98	73.64	74.33	74.21	74.23	74.3	73.4	72.66	72.22	72.15	72.65	72.27	75.01	74.82	74.74
RAISE-1k Erosion	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	95	98.78	98.97	99.03	98.65	99.08	98.16	98.17	97.99	97.97	97.96	97.27	97.42	97.82	96.87	95.97	98.55	98.8	99.74
	90	97.34	97.53	97.16	96.34	96.15	95.42	95.96	95.47	96.4	95.68	96.43	97.29	97.46	96.87	95.97	98.03	98.96	98.96
	85	94.23	94.54	95.24	94.41	94.82	94.05	93.78	94.61	95.26	94.33	93.35	94.21	94.98	95.76	95.97	96.94	97.65	96.69
	80	88.17	88.53	87.96	87.19	86.43	86.89	85.98	86.4	86.05	86.19	85.98	85.16	84.73	84.48	85.26	91.52	91.63	90.8
	75	78.26	78.57	77.6	78.56	78.8	77.9	77.18	77.76	78.21	78.35	78.16	77.67	77.95	77.11	82.99	83.14	82.75	
	70	77	77.63	76.94	76.42	76.53	76.45	76.78	77.18	76.65	77.42	78.25	78.16	78.02	77.67	77.95	81.96	81	
RAISE-1k Dilation	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	95	98.22	98.07	98.25	99.04	99.12	99.75	99.25	99.02	98.94	98.48	97.59	96.83	96.17	95.26	99.36	99.49	99.92	
	90	97.81	98.07	97.33	98.33	99.04	99.12	99.72	99.25	98.36	97.53	97.41	97.59	96.83	96.17	95.26	97.96	98.49	
	85	94.74	93.79	93.83	94.55	95.13	94.24	94.73	93.81	94.58	95.42	94.56	94.25	95.24	94.95	94.62	96.13	95.71	96.06
	80	88.46	88.97	89.76	90.07	90.22	91.18	91.09	91.07	91.61	91.52	90.57	89.65	90.63	91.07	90.53	92.52	93.43	92.55
	75	77.85	76.87	76.98	76.83	76.34	75.63	75.31	74.31	73.89	74.45	74.17	73.83	74.53	74.61	82.68	82.58	83.06	
	70	77.46	76.73	76.14	75.66	74.71	65.67	65.73	65.93	62.58	62.74	62.69	62.64	63.05	63.03	63.58	63.61	64.04	64.03
Dresden Erosion	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	95	97.84	98.17	98.73	98.09	98.17	98.03	97.07	98.03	97.22	97.95	97.84	98.19	97.97	97.16	96.67	96.15	96.68	95.76
	90	97.59	98.17	97.78	97.98	97.66	96.72	97.07	96.65	95.72	96.57	96.58	96.44	96.06	96.71	96.27	96.15	96.25	95.69
	85	96.51	97.21	96.96	96.56	95.69	95.69	95.62	93.89	94.15	93.67	94.08	93.95	93.5	93.16	92.9	92.03	91.64	
	80	90.05	89.45	89.25	88.97	88.55	89	88.3	88.9	89.56	89.71	90.29	90.16	89.64	88.65	89.32	89.67	89.83	
	75	79.4	80.18	79.99	79.43	79.61	78.99	81.44	81.44	82.37	82.33	81.64	81.26	81.41	80.83	81.14	81.31	81.89	
	70	83.08	83.64	83.46	84.22	84.65	84.18	84.58	73.26	73.62	74.6	75.4	75.12	75.53	76.29	76.76	76.76	76.76	
Dresden Dilation	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	95	99.05	99.96	99.53	99.12	98.32	97.5	98.05	97.84	98.45	98.65	98	97.64	97.06	97.75	99.33	97.44	97.44	96.55
	90	97.06	97.5	97.17	97.52	96.7	95.83	96.75	97.51	97.07	96.95	96.53	97.21	96.36	95.98	95.31	96.01	95.67	
	85	97.06	97.21	97.17	97.52	96.7	95.83	95.38	94.81	94.71	95.65	96.16	95.28	95.41	95.35	95.21	94.61	94.76	93.8
	80	89.2	88.78	88.51	88.83	88.92	88.64	89.69	90.24	90.14	90.91	90.4	89.08	89.21	88.83	88.82	89.09	89.14	
	75	81.46	81.36	81.68	81.56	81.32	81.12	80.21	82.31	81.63	81.76	81.92	82.78	82.69	82.3	81.63	82.55	81.79	
	70	75.02	75.73	76.2	75.7	75.99	76.95	77.03	74.12	74.83	75.04	75.79	75.83	76.52	77.19	77.79	78.31	77.45	77.42
RAISE-1k Erosion	100	100																	

TABLE 4: Results on erosion detection on Raise under different attacks, using SVM with RBF kernel, 10-fold cross validation.

QF	Pepper & Salt	Gaussian (3x3)	Median Filter (3x3)
100	97.92	81.21	87.94
95	95.72	81.14	86.44
90	95.19	81.01	84.27
85	91.79	79.16	80.67
80	86.99	78.76	80.57
75	83.04	75.06	76.42
70	81.23	70.24	71.49

dilation detection using SVM with RBF kernel on RAISE1k are correspondingly: 97.92%, 81.21%, 87.94%, 67.44%, and 67.81% (the results are summarized in Figure 9. According to the results, we can claim that the proposed method is robust against noise addition but not against processing involving interpolation (e.g., resizing).

We applied a further test to understand if the proposed method can still detect the morphological filter after two level of processing where the second one is a compression. Thus, each image after a morphological filter is first contaminated with a noise, and then is compressed under different quality factors. The results in Table 4 show that the proposed method can still detect the morphological filter, however, its performance is getting worst as the compression level increases.

D. MORPHOLOGICAL FILTERS VERSUS OTHER FILTERS

Finally, we want to determine the capability of our approach in distinguishing between morphological filters from other filters that produce similar results. In particular, we con-

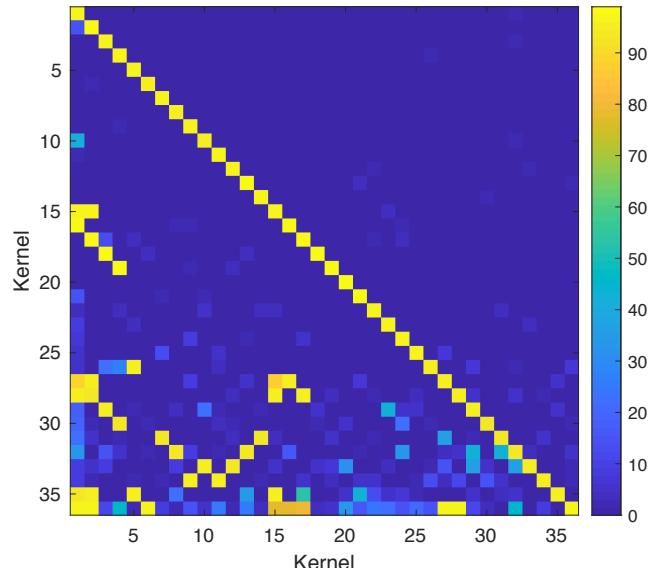


FIGURE 8: Confusion matrix on erosion detection for multiple kernels. Dataset: Raise, QF = 90, full resolution. Values are normalized and are in percentage.

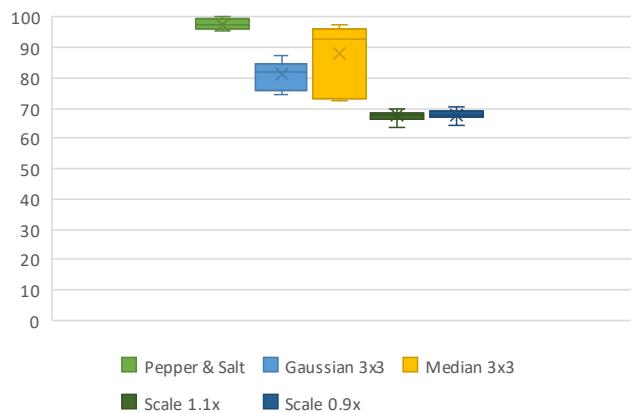


FIGURE 9: Precision (computed as percentage) on dilation detection under different attacks. Dataset: RAISE-1k, full resolution, QF = 100, using SVM with RBF kernel, 10-fold cross validation. All 36 kernels are computed and the results are shown in the error bars.

sidered three filters: pepper & salt, Gaussian lowpass and median filtering. In all cases, the filtered images are uncompressed and passed to the detector to reveal possible false alarms. In this experiment, we trained the erosion versus pristine classifier with 800 uncompressed images from RAISE in full resolution. The trained classifier is then applied to the rest of the 200 images, filtered with pepper & salt, Gaussian lowpass and median filters. The average number of false positives (over 200 images) for pepper & salt, Gaussian with window size 3×3 , Gaussian with window size 5×5 , median with window size 3×3 , and median with window size 5×5 are 3.51%, 5.72%, 5.28%, 14.11%, and 13.17%, respectively. The results are summarized in Figure 10. This confirms that the proposed method can distinguish between morphological filter and other filters. Only in the case of median filters the confusion increases, but this is very reasonable, since in the case of grayscale images the effect of morphological filters and median filters is very similar.

VI. CONCLUSIONS

In this work we propose an effective detection strategy to assess the use of morphological filtering in a grayscale context. We deal with uncompressed images proposing a deterministic approach, based on mathematical properties enjoyed by basic morphological operators. We additionally propose a modified pipeline to detect morphological processing in compressed images, by exploiting the difference histogram information as feature for classification. We present a testing phase in which both uncompressed and compressed scenarios are taken into consideration. Results show the effectiveness of our proposed approach in both cases. Moreover, the proposed approach is able to determine the adopted structuring element for moderate compression factors, and is robust against a number of attacks.

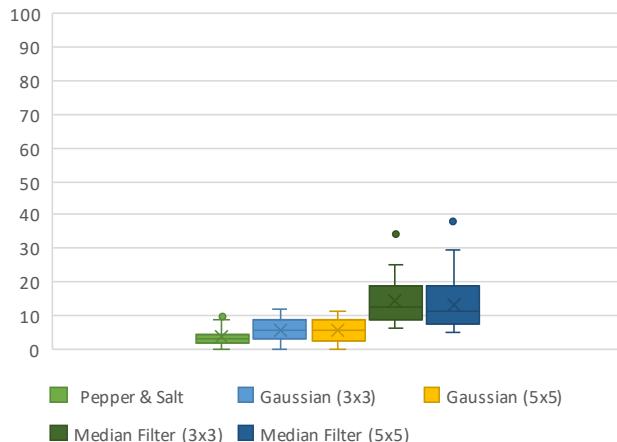


FIGURE 10: False Positive (computed as percentage) on 200 images from RAISE on erosion detection over all 36 filters. Images are in raw format, full resolution. The classifiers were trained on 800 images (eroded vs. non-filtered) and tested on the rest 200 images (filtered with pepper & salt, Gaussian and median).

REFERENCES

- [1] M. Stamm, M. Wu, and K. Liu, “Information forensics: an overview of the first decade,” *IEEE Access*, pp. 167–200, 2013.
- [2] H. Farid, *Photo forensics*, M. Press, Ed., 2016.
- [3] A. T. Ho, *Handbook of digital forensics of multimedia data and devices*. John Wiley & Sons, 2015.
- [4] A. Piva, “An overview on image forensics,” *ISRN Signal Processing*, 2013.
- [5] H. Sencar and N. Memon, Eds., *Digital Image Forensics - There is more to a picture than meets the eye*. Springer, 2013.
- [6] D. Cozzolino and L. Verdoliva, “Noiseprint: A cnn-based camera model fingerprint,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2020.
- [7] A. D. Rosa, A. Piva, M. Fontani, and M. Iuliani, “Investigating multimedia contents,” in *IEEE International Carnahan Conference on Security Technology (ICCST)*, 2014, pp. 1–6.
- [8] P. Bestagini, M. Tagliasacchi, and S. Tubaro, “Image phylogeny tree reconstruction based on region selection,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.
- [9] Z. Dias, A. Rocha, and S. Goldenstein, “Image phylogeny by minimal spanning trees,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 774–788, 2012.
- [10] A. de Oliveira and et al., “Multiple parenting phylogeny relationships in digital images,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 328–343, 2016.
- [11] F. de O. Costa and et al., “Image phylogeny forests reconstruction,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1533–1546, 2014.
- [12] E. Ardizzone, A. Bruno, and G. Mazzola, “Copy-move forgery detection by matching triangles of keypoints,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2084–2094, 2015.
- [13] D. Cozzolino, G. Poggi, and L. Verdoliva, “Efficient dense-field copy-move forgery detection,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2284–2297, 2015.
- [14] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra, “A SIFT-based forensic method for copy-move attack detection and transformation recovery,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099–1110, 2011.
- [15] D. Cozzolino, G. Poggi, and L. Verdoliva, “Splicebuster: a new blind image splicing detector,” in *IEEE Workshop on Informations Forensics and Security (WIFS)*, 2015, pp. 1–6.
- [16] X. Zhao, S. Wang, S. Li, and J. Li, “Passive image-splicing detection by a 2-D noncausal Markov model,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 185–199, 2015.
- [17] D. Vázquez-Padín, F. Pérez-González, and P. Comesáñ-Alfaro, “A random matrix approach to the forensic analysis of upscaled images,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2115–2130, 2017.
- [18] C. Pasquini and R. Böhme, “Information-theoretic bounds for the forensic detection of downsampled signals,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1928–1943, 2019.
- [19] C. Pasquini, G. Boato, and F. Pérez-González, “Statistical detection of JPEG traces in digital images in uncompressed formats,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2890–2905, 2017.
- [20] W. Shan, Y. Yi, R. Huang, and Y. Xie, “Robust contrast enhancement forensics based on convolutional neural networks,” *Signal Processing: Image Communication*, vol. 71, pp. 138–146, 2019.
- [21] V. Conotter, H. Farid, and G. Boato, “Detecting photo manipulation on signs and billboards,” in *IEEE International Conference on Image Processing (ICIP)*, 2010.
- [22] E. Kee, J. O’Brien, and H. Farid, “Exposing photo manipulation from shading and shadows,” *ACM Transactions on Graphics*, vol. 33, no. 165, pp. 1–21, 2014.
- [23] M. Iuliani, G. Fabbri, and A. Piva, “Image splicing detection based on general perspective constraints,” in *IEEE Workshop on Informations Forensics and Security (WIFS)*, 2015, pp. 1–6.
- [24] K. Bahrami, A. Kot, L. Li, and H. Li, “Blurred image splicing localization by exposing blur type inconsistency,” *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 5, pp. 999–1009, 2015.
- [25] G. Cao, Y. Zhao, R. Ni, and X. Li, “Contrast enhancement-based forensics in digital images,” *IEEE Trans. on Information Forensics and Security*, vol. 9, no. 3, pp. 515–525, 2014.
- [26] G. Cao, Y. Zhao, R. Ni, and A. Kot, “Unsharp masking sharpening detection via overshoot artifacts analysis,” *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 603–607, 2011.
- [27] M. Kirchner and J. Fridrich, “On detection of median filtering in digital images,” in *SPIE*, vol. 7541, (2010), pp. 101–112.
- [28] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian, “Forensic detection of median filtering in digital images,” in *IEEE Int. Conf. on Multimedia and Expo, ICME2010*, (2010), pp. 89–94.
- [29] H.-D. Yuan, “Blind forensics of median filtering in digital images,” *IEEE Trans. Information Forensics and Security*, vol. 6, no. 4, pp. 1335–1345, 2011.
- [30] X. Kang, M. C. Stamm, A. Peng, and K. J. R. Liu, “Robust median filtering forensics using an autoregressive model,” *IEEE Trans. Information Forensics and Security*, vol. 8, no. 9, pp. 1456–1468, 2013.
- [31] C. Chen, J. Ni, and J. Huang, “Blind detection of median filtering in digital images: A difference domain based approach,” *IEEE Trans. Image Processing*, vol. 22, no. 12, pp. 4699–4710, 2013.
- [32] Y. Zhang, S. Li, S. Wang, and Y. Q. Shi, “Revealing the traces of median filtering using high-order local ternary patterns,” *IEEE Signal Processing Letters*, vol. 21, no. 3, pp. 275–280, 2014.
- [33] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, “Median filtering forensics based on convolutional neural networks,” *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849–1853, 2015.
- [34] C. Pasquini, G. Boato, N. Anjalic, and F. D. Natale, “A deterministic approach to detect median filtering in 1d data,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1425–1437, 2016.
- [35] H. Gao, M. Hu, T. Gao, and R. Cheng, “Robust detection of median filtering based on combined features of difference image,” *Signal Processing: Image Communication*, vol. 72, pp. 126–133, 2019.
- [36] A. Haas, G. Matheron, and J. Serra, “Morphologie mathématique et granulométries en place,” in *Annales des mines*, vol. 11, no. 736–753, 1967, pp. 7–3.
- [37] N. M. Al-Shereefi, “Morphological filter is an active tool for edge detection in noisy image,” *British Journal of Science*, vol. 3, no. 2, pp. 148–156, 2012.
- [38] P. Maragos, “Morphological filtering for image enhancement and feature detection,” in *The Image and Video Processing Handbook*, 2004.
- [39] F. G. De Natale and G. Boato, “Detecting morphological filtering of binary images,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1207–1217, 2017.
- [40] Y. Nakagawa and A. Rosenfeld, “A note on the use of local min and max operations in digital picture processing,” *Maryland University College Park Computer Science Center, Tech. Rep.*, 1977.

- [41] G. Schaefer and M. Stich, "Ucid: An uncompressed color image database," in Storage and Retrieval Methods and Applications for Multimedia 2004, vol. 5307. International Society for Optics and Photonics, 2003.
- [42] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," Journal of Digital Forensic Practice, vol. 3, no. 2-4, pp. 150–159, 2010.
- [43] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "Raise: a raw images dataset for digital image forensics," in ACM Multimedia Systems Conference, 2015.



GIULIA BOATO is Associate Professor at the Department of Information Engineering and Computer Science (DISI) of the University of Trento (Italy). She is Associate Editor of the IEEE Transactions on Image Processing, the ELSEVIER Signal Processing: Image Communication. Her research interests are focused on image and signal processing, with particular attention to multimedia data security and digital image and video forensics, but also intelligent multidimensional data management and analysis. She is author of 120 papers in international conferences and journals, with H-index 22 (Google Scholar). She is elected member of the IEEE Multimedia Signal Processing Technical Committee (MMSP TC), the IEEE Information Forensics and Security Technical Committee (IFS TC) and the EURASIP Special Area Teams Biometrics, Data Forensics, and Security.



DUC-TIEN DANG-NGUYEN is currently an Associate Professor at the Department of Information Science and Media Studies (Infomedia), University of Bergen (Norway). His main area of expertise is on multimedia forensics, lifelogging, multimedia retrieval, and computer vision. He is author of 90 papers in international conferences and journals, with H-index 18 (Google Scholar, December 2019). He is member of IEEE Signal Processing, ACM, SIGIR, IAPR and INSTICC. He has organised and chaired over 20 workshops and special sessions, has been served as program committee for numerous international conferences and workshops, and has peer-reviewed for many top journals and conferences in the fields of lifelogging, multimedia forensics, and pattern recognition.



FRANCESCO G.B. DE NATALE was the Head of the Department of Information Engineering and Computer Science (DISI), University of Trento (Italy) from 2007 to 2010, where he currently leads the MMLab Research Laboratory. He is also a Professor of telecommunications with the University of Trento. His research interests include multimedia communications, where he published over 200 works in the major international journals and conferences. He is also a member of the Board of Governors of the CNIT Consortium. He has been an Associate Editor of the IEEE Transactions on Multimedia and the IEEE Transactions on Circuits and System for Video Technologies.

• • •