Multimed Tools Appl (2017) 76:4801–4834
DOI 10.1007/s11042-016-3795-2

Large-scale evaluation of splicing localization algorithms for web images

Markos Zampoglou[1] · Symeon Papadopoulos[1] · Yiannis Kompatsiaris[1]

Received: 2 November 2015 / Revised: 27 June 2016 / Accepted: 19 July 2016 /
Published online: 1 September 2016

Abstract With the proliferation of smartphones and social media, journalistic practices are increasingly dependent on information and images contributed by local bystanders through Internet-based applications and platforms. Verifying the images produced by these sources is integral to forming accurate news reports, given that there is very little or no control over the type of user-contributed content, and hence, images found on the Web are always likely to be the result of image tampering. In particular, image splicing, i.e. the process of taking an area from one image and placing it in another is a typical such tampering practice, often used with the goal of misinforming or manipulating Internet users. Currently, the localization of splicing traces in images found on the Web is a challenging task. In this work, we present the first, to our knowledge, exhaustive evaluation of today's state-of-the-art algorithms for splicing localization, that is, algorithms attempting to detect which pixels in an image have been tampered with as the result of such a forgery. As our aim is the application of splicing localization on images found on the Web and social media environments, we evaluate a large number of algorithms aimed at this problem on datasets that match this use case, while also evaluating algorithm robustness in the face of image degradation due to JPEG recompressions. We then extend our evaluations to a large dataset we formed by collecting real-world forgeries that have circulated the Web during the past years. We review the performance of the implemented algorithms and attempt to draw broader conclusions with respect to the robustness of splicing localization algorithms for application in Web environments, their current weaknesses, and the future of the field. Finally, we openly share the framework and the corresponding algorithm implementations to allow for further evaluations and experimentation.

Keywords Image forensics · Image splicing · Forgery localization · Web multimedia verification

✉ Markos Zampoglou
markzampoglou@iti.gr

[1] Centre for Research and Technology Hellas, Information Technologies Institute, 6th km Harilaou - Thermi, 57001 Thessaloniki, Greece

1 Introduction

Since the dawn of photography, providing photographic evidence has been a powerful way to bolster or refute a claim, and alleviate ambiguity. When dealing with a story, be it a news report or an accusation in a court of law, an image can be used to strengthen or disprove a certain claim and, as a result, inform the public, and condemn or acquit an individual respectively. But all this only holds under the provision that what is depicted in the image can be trusted to be true. Modifying the content of an image after it has been captured is a practice that dates back to the first days of photography. Nowadays, the proliferation of digital image capturing devices and user-friendly image processing software have made changing the content of images an easy feat both for professionals and non-experts.

Courthouse investigations and news reporting are the two fields that are most commonly associated with the problem of image manipulation, its implications, and the need for reliable forensics analysis tools in order to verify the information contained in images. In both cases, there is a need –and ongoing research- to devise tools capable of scanning an image and –either with or without other background information- provide the user with a report with respect to the originality and authenticity of the image. In the general case, any piece of information can be of significant value. This can include the capture parameters (geolocation, camera model, environment lighting level), post-processing (compression quality, filters applied) or explicit tampering (e.g. locating modifications on an image region). However, detecting actual forgeries is obviously the aspect that attracts the most research attention.

Our work focuses specifically in the journalistic case, and especially in the scenario where journalists evaluate social media streams and reports posted on the Web. In this manner, investigators can extract real-time information from grassroots sources to form reports on important events without being present in the field. In that case, journalists coming across a photo depicting an interesting event need to be able to quickly verify the truthfulness of its content. This is important, since being able to accurately tell whether it is a forgery or not means all the difference between selling an interesting news story and harming the credibility of the news organization (and their professional reputation). Figure 1 presents two cases of user-provided news content from the recent past. One is an authentic image captured by a bystander and consecutively used by news agencies for reporting the event, while the other is a forgery that was disseminated via Twitter. While the latter was not officially reproduced as a confirmed news item, it was actually featured as an unconfirmed source in a prominent news agency's website. Both events depicted in Fig. 1 appear to be very unlikely, but it is the aim of a verification system to authoritatively distinguish truth from fiction.

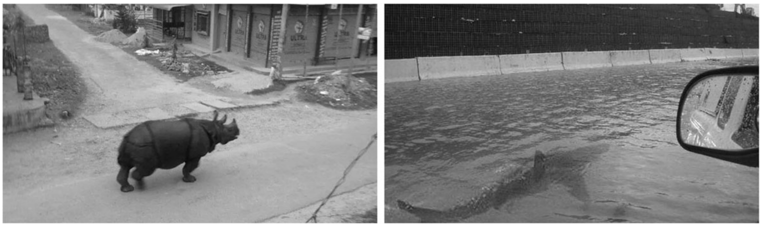

Fig. 1 News images contributed by non-journalists in the past and disseminated via social media. Left: an authentic photo of a rhinoceros chasing vehicles in the streets of Hetauda City, Nepal, March 2015. Right: a forged image of a shark supposedly swimming in the streets of Puerto Rico following Hurricane Irene, August 2011

The task of verifying images collected from the Web or social media puts a number of additional constraints on what an algorithm should be able to achieve, and what resources we can expect it to have access to. There are two main particularities of the specific application field: One is that, unlike courthouse forensics, we cannot expect to have access to the device used to capture the original image, or even know the model. The other is that, since we have acquired the image from the Web, there is a significant chance that the image has undergone some transformations. Such transformations can take the form of resaves, rescales, and/or filtering, both before and after the possible tampering.

In our previous work [54] we first identified the problems caused by these specific task characteristics, presented a dataset of real-world forgeries to use as an evaluation framework, and presented a limited evaluation of a small number of algorithms on this scenario. In this paper, we present a significantly extended version of our previous work, by taking a larger set of algorithms that, to our knowledge, covers the bulk of state-of-the-art tampering localization approaches, and evaluate them against all major datasets we are aware of. We further evaluate the performance of these algorithms in the face of image degradation caused by lossy recompression at various qualities and rescaling at various sizes, and discuss the main issues with today's evaluation methodologies and the interpretability of the obtained results. Guided by our findings, we discuss the real-world applicability of these algorithms, and consider further research directions.

2 Background and related work

One fundamental distinction often made in image forensics algorithms is that of *active* versus *passive image forgery detection*. Active forgery detection concerns algorithms where images have been preemptively processed and imperceptibly marked, so that alterations can subsequently be detected and localized. Passive detection, on the other hand, which is the focus of this article, deals with completely unknown images with no prerequisites. Passive image forgery detection has been an active research field for more than a decade now, and a variety of methods have been proposed in the past, while a number of different reference datasets have been used for algorithm evaluations. In recent years, a number of surveys of the field have attempted to organize the state-of-the-art and the major challenges [46] [6] [49]. In this section, we present the most prominent methods relevant to our application scenario, as well as the benchmark datasets that can be used for evaluation.

2.1 Tampering detection approaches

2.1.1 Types of tampering attacks

The first-level taxonomy of digital image forgery detection methods is based on the type of forgery they aim to detect. Four broad categories are *copy-move forgeries*, *splicing*, *inpainting*, and *broad-scope image operations*. Copy-moving means taking a part of an image and placing copies of it within the same image. The intention behind such a practice may either be to add false information (e.g. make a structure or crowd appear bigger) or to hide information (by covering it using other parts of the image). Image splicing refers to the practice of copying a part of one image into another to give the false impression that an additional element was present in a scene at the time that the photograph was captured. Inpainting means drawing over the image using an image processing software tool such as a "brush". On the one hand, the result of inpainting is often similar to that of splicing, in the sense that the tampered area of the image takes different properties from the rest of the image, in terms of noise, JPEG compression traces, etc., and the same family of algorithms can generally be used to detect both. On the other hand, since in-painted areas often exhibit significant self-similarity, their localization is also possible using methods resembling copy-move detection algorithms [8]. Finally, the third category includes a diverse set of operations, such as filtering, cropping, rescaling or histogram adjustments, whose common characteristic is that they are very often used without malicious intent. Many images published on the Web, even from non-professional photographers, commonly undergo some form of post-processing, while publishing platforms also often automatically apply such transformations to images uploaded to them. Thus, while in a judicial investigation such operations may serve as indications of malicious tampering, with respect to everyday encounters in the Web they are generally considered as non-malicious. In this context, only copy-moving and splicing can unambiguously qualify as "forgeries".

Algorithms seeking to find copy-move attacks generally follow a common methodology [48] [15] [36] [2]: a) block or keypoint descriptors are extracted from the image, b) a matching step seeks the best similarities between them within the image, and c) a post-processing step attempts to organize and group the matched pairs into coherent candidate regions of copy-moving. In modern approaches, the process has to take into account the possibility of, filtering, rotation or other transformations besides simple translation. The search is conducted by attempting to group neighboring matches into clusters, and estimating the parameters of possible affine transforms that could best explain the matches found in a region. This is used to eliminate spurious matches and still be able to build models that are robust to transformations other than simple translation. A past large scale survey and evaluation of the state-of-the-art in the field can be found in [11]. The major research challenges in copy-move forgery detection are maintaining low computational complexity, maximizing localization precision, and maintaining robustness in the face of modifications.

In the work presented here, we do not consider copy-move detection, for two main reasons: a) the field of copy-move detection has achieved significantly more measurable progress, and recent surveys and comparative evaluations of the field already exist in literature [11] [45] [20], and b) copy-moving has essentially a narrower scope, since certain splicing detection algorithms should in fact be able to detect copy-move forgeries as well, due to the disturbance these attacks cause on local image structure. That is because, if the copy-moved region undergoes some form of resampling, such as scaling or rotation, then it will probably lose the common characteristics it shares with the rest of the image, such as JPEG compression traces or noise patterns, and will be detectable using splicing localization. Furthermore, even if no resampling occurs, it is possible that certain algorithms -such as the blocking artifact detection described below- could still achieve successful localization due to the disturbance caused by the move and resulting misalignment.

Algorithms detecting image splicing, on the other hand, are generally based on the assumption that the spliced area of the image will differ in some fundamental aspect from the rest of the image. A large family of algorithms for splicing detection [56] [42] [27] aim to simply deduce whether an image has been spliced (or, generally, locally altered) without being able to identify where. Such methods generally tend to use machine learning over visual features to train binary classifiers, and often achieve very high success rates. However, there are two major issues with such approaches: a) such algorithms run the risk of being dataset-specific, in the sense that training over one set does not necessarily allow for generalization into another, and likely, not in realistic application settings, and b) human users tend to be mistrustful of their results: indeed, an automated analysis claiming that there is a certain chance that an image has been tampered, without providing interpretable clues as to why this conclusion was drawn and offering no localization information, could only serve as an auxiliary tool and not as a substantial authoritative tool. In this work, we are instead focusing on forgery localization algorithms, attempting to find local discrepancies over some type of information that ought to be consistent across the whole image, and thus providing clues to the investigator as to where the forgery may have taken place. Many different types of traces have been used in the past, and can be categorized in three broad groups: *noise patterns*, *Color Filter Array interpolation patterns* and *JPEG-related traces*.

2.1.2 Noise patterns

Algorithms based on *noise patterns* are based on the assumption that the combination of capturing device, the capture parameters of each image, and any subsequent image post-processing or compression create unique noise patterns that differ between images. If we can isolate them, these patterns can potentially be used to separate the spliced area from the rest of the recipient image. One notable algorithm in this category is based on identifying noise patterns by wavelet-filtering the image under the assumption that the local variance of the high-frequency channel will differ significantly between the splice and the recipient image [41]. Another approach isolates local noise using the observation that different frequency sub-bands within an image tend to have constant and positive kurtosis values in their coefficients [40]. The former has the advantage of being based on a ubiquitous principle and of being able to generate output maps using a fairly simple transformation of the original image. On the other hand, it is fairly sensitive to variations in the image local frequency spectrum, a feature which is irrelevant to actual splicing, and is thus occasionally prone to generating artefacts (false positives). Also, the calculation of local noise variance has to take place over relatively large blocks, and thus it offers a relatively coarse localization of the tampered area. The latter method is much more refined in localizing forgeries; however, the localization accuracy is often dependent on a number of parameters, which could potentially make it difficult to find the optimal settings for automatic application. Recently, a novel, promising method called the SpliceBuster [16] was proposed for splicing localization based on the high-frequency image content. The algorithm is based on high-frequency information extracted from the image using a linear high-pass filter. This information is consecutively quantized and expressed using a co-occurrence descriptor. The assumption is that the local descriptions of image regions with different origins will exhibit different statistical properties. The algorithm can operate both in a semi-supervised scenario, and a fully automatic one. In the former, the user indicates an image region that is guaranteed to be untampered. This is used to calculate the natural statistical properties of the descriptor, and consecutively regions in the rest of the image are evaluated with respect to their conformance to the model. In the fully automatic case, where we are not sure which area of the image has not been tampered, the method uses an Expectation-Maximization algorithm to fit two different distributions (Tampered-Untampered) to the local descriptors. By locating the local descriptors that conform to the "Tampered" model, we can localize the regions where the splicing took place.

A large family of noise-related methods for splicing detection are based on sensor *Photo Response Non-Uniformity* (PRNU), that is the distinct noise patterns produced by the physical characteristics of each unique capturing device on all images taken by it [9] [34] [10]. In such methods, the basic idea is to first estimate a device's PRNU spatial map, and then evaluate whether the image under question conforms to it. If local deviations appear, the presence of a splice in the corresponding region is posited. A defining characteristic of all PRNU methods is the need to first estimate the PRNU pattern of the device used to capture the image. This can be achieved by capturing a large number of varied images using the same device, and using them to estimate the device's PRNU noise. This approach is reasonable in forensic scenarios where time and resources are available, such as judicial forensic analysis, but is unrealistic in the Internet-based scenarios we are investigating, where we are faced with isolated images captured using unidentified cameras. Thus, PRNU-based methods will not be further considered in this work.

2.1.3 CFA interpolation pattern disturbances

Most modern digital images are captured using a single sensor overlaid with a *Color Filter Array* (CFA), which produces one value per pixel, and the image is then transformed into three channels using interpolation. Thus, for each channel, a number of values originate from the environment, while the rest are interpolated from them. Splicing can disrupt the CFA interpolation patterns in multiple ways: a) splices are often filtered or rescaled, which disrupts the original pixel interrelations, b) not all cameras use the same CFA array or interpolation algorithm, so mixing two different images may cause discontinuities, and c) even simply misaligning the splice with the rest of the image disrupts the pattern –in the latter case, since typical CFA array patterns follow a 2 × 2 grid, there is a 75 % chance that any splice (or even copy-move) forgery will disrupt it.

Two recent promising CFA-based methods are [21, 23]. In the former, two different features are proposed: The first attempts to detect the CFA pattern used during image capture by subsampling the image using various possible selection patterns, re-interpolating it, and comparing it to the original. Having emulated the CFA interpolation process using the estimated parameters, one could then make use of local discrepancies between the interpolated and the observed values to detect local tampering. The second isolates image noise by using a filter de-noising process and then calculates a measure of the relationship between the noise variance of interpolated and natural pixels. If the two variances are too similar in a region, there is a high probability that pixel values have been disrupted by tampering.

More recently, in [23] the local variances of natural and interpolated pixel values are also compared, but following a formulation that allows estimating a tampering probability per block instead of simply indicating local discrepancies via an arbitrary value. Overall, using CFA interpolation discrepancies has yielded promising results in a number of cases in the past; however, such methods suffer from a major disadvantage: these traces are strongest at the moment the image is created, and tend to be particularly sensitive to the effects of JPEG compression [23]. Thus, CFA methods are at their best when operating on raw or losslessly compressed images, and are often inadequate for images circulating on the Web.

2.1.4 JPEG compression traces

The third category of splicing localization algorithms is based on exploiting the traces left by JPEG compression. The vast majority of such methods use features from one of two subgroups: *JPEG quantization artifacts* or *JPEG compression grid discontinuities*.

Quantization of an image's DCT coefficients is a major step in the JPEG compression pipeline, in which the quantization factor is a function of the chosen compression quality. It has been observed that consecutive JPEG compressions at different qualities lead to specific periodicities in the DCT coefficient distribution. A number of algorithms attempt to detect splicing under the assumption that the spliced region might have been smoothed, resampled or otherwise made to lose its first JPEG compression characteristics. Thus, in the final spliced image the corresponding blocks will appear to have undergone a single compression, while the rest of the image will have undergone two. Algorithms based on *Double Quantization* (DQ) attempt to model the periodic DCT patterns caused by the two compressions, and detect local regions that do not fall into this model. In [37], the DCT coefficient distribution is modelled for the entire image, and then each block is evaluated with respect to its conformance to the model. For each block, the probability that it has originated from a different distribution is evaluated. [5] is a direct extension of that principle, in which the model is made significantly more robust by taking into account the fact that the DCT coefficient distribution estimation may be tainted by the presence of both tampered and untampered blocks. Finally, a third approach [1] at detecting double quantization inconsistencies is based on the observation that the distribution of JPEG DCT coefficients changes with the number of recompressions. The DCT coefficient distribution of natural images follows Benford's law, i.e. their first digits follow a frequency distribution where small values are significantly more common than larger ones. With the DCT quantization step included at each JPEG recompression, this pattern is gradually destroyed. [1] proposes training a set of SVMs over the different first digit distributions produced by various single and double compression quality combinations. We can then split the image in sub-blocks, and estimate the probability of each having been single- or double-compressed. The presence of single-compressed blocks in an otherwise double-compressed image can be considered a strong indication of tampering.

In contrast to these DQ methods, when the 8 × 8 JPEG block grid is shifted between the two compressions (e.g. by cropping) an algorithm can attempt to model Non-Aligned (NA) JPEG compression [3] [4]. While Non-Aligned quantization detection methods attempt to capture a phenomenon that is not particularly likely to occur in the real world, it is still an eventuality that may not be captured by other techniques and thus these algorithms can play an important part in a forensic analysis process.

A simpler alternative [52] is to simply model the entire image DCT coefficient distribution for each channel using a measure of the degree of quantization, whose local inconsistencies can be indicative of splicing. While the algorithm can be effective in detecting discrepancies in the local JPEG quantization history of each block, it differs from other DCT-based methods in that its output is not probabilistic, which makes it relatively more difficult to interpret.

JPEG grid discontinuities, on the other hand, occur when the splice is placed in the recipient image in a way that misaligns with the 8 × 8 block grid used for compression. Even at very high compression qualities, the 8 × 8 pattern creates a grid structure in pixel values which is usually invisible to the eye but can be detected using appropriate filtering. When the 8 × 8 grid is absent from a region, or is misaligned with the rest of the image, this constitutes a strong hint towards the presence of a splice [39] [35]. In the former, a descriptor vector is extracted from the image, modelling the presence of such JPEG blocking artifacts, and then an SVM is trained for image-level inconsistency detection –but not localization. In the latter, a local feature corresponding to the local intensity of the blocking pattern is extracted. The feature's variations indicate local absence or misalignment of the grid, which is telltale of tampering, although the algorithm may also be misguided by variations in the actual image content.

With respect to exploiting JPEG compression, there exist two more approaches which have attracted attention from the forensics community, JPEG Ghosts [22] and Error Level Analysis (ELA) [33]. Both are based on the same practice, that of recompressing the suspect image as JPEG, and pixel-wise subtracting the recompressed version from the original, and then drawing conclusions from the residual map. The former is based on the principle that, if the splice originates from a JPEG image of quality Q1, and is placed in a JPEG image of a Q2, it will still be carrying certain characteristics of its first compression. Thus, if the resulting image is recompressed at quality Q1 and subtracted from itself, there will be significantly less residual over the area of the splice. By recompressing the image at multiple different qualities and subtracting each one from the original, we can check if some image region behaves differently from the rest of the image at some compression level. One major issue with JPEG Ghosts is that they produce a large number of output maps –one per recompression quality-, which makes it more likely that an investigator will be misguided by irrelevant artifacts in some of them. ELA, on the other hand is a particularity in the field, in the sense that it has attracted relatively little research attention [50] [44], yet is currently the only method explored here that has seen widespread application outside of the research community.[1] The basic principle of ELA is that, as an image undergoes consecutive JPEG compressions, even at high qualities, it begins to lose its high-frequency content. Thus, after a point, further recompressions cause no loss of such content. This means that, if a part of an image has undergone more resaves than the rest, when a recompressed version of the image is subtracted from it, the regions that have undergone fewer compressions will leave a stronger residual than the rest of the image.

2.1.5 Other splicing detection approaches

Besides the categories discussed above, a number of algorithms exist that take advantage of specific phenomena during image capturing to detect splicing. One such case is [53], where inconsistencies in lens-caused aberrations, and specifically a phenomenon called *Purple Fringing Aberration* (PFA) can be used to localize splicing in an image. Another is [30] where inconsistencies in motion blur are used to detect image regions of different origins. A third is [38], where inconsistencies in shadow parameters are treated as indications of splicing, and a fourth is [17] which localizes forgeries through discontinuities in illumination color. While experimentally very effective in particular cases, such approaches are effective only under very specific circumstances, for example [30] requires matte surfaces and outdoor environments, while [53] depends on the presence of motion blur. Such specialized phenomena can be used much more effectively for splicing detection when included in semi-automatic contexts. Semi-automatic methods require the user to assist the algorithm by providing additional information, such as the method in [31], where the user must provide shading constraints on the scene geometry in order to detect inconsistencies in shading. However, while such methods can occasionally provide good results, they are by nature difficult to evaluate on a large scale, as they require human intervention for each detection, and, as a result, we do not consider them in the comparative evaluation presented here.

Finally, one recent family of splicing detection algorithms attempts to take advantage of the fact that splices –especially in the Web-based news reporting scenario- are often composed from source

[1] Two Web platforms currently providing ELA analysis are FotoForensics (http://fotoforensics.com/), and Forensically (http://29a.ch/photo-forensics/#error-level-analysis).

images that can also be found on the Web. In that case, a similarity search could allow us to locate the sources of the splice. There is a significant body of work attempting to trace the lifecycle of an image in the realistic scenario, where an original image is uploaded on the Web and consecutively spliced, transformed, resaved and modified, and where the resulting images can be collected and analyzed. Approaches such as "image archaeology" [32] and "image phylogeny" [18] search within datasets in order to organize images in tree or forest structures that represent their history, and can be used to trace the earliest, least modified versions of the image. More recently, proposed approaches explicitly identify splices using partial similarity search [29] [43] or extending the phylogeny approach to cases of multiple parenting, which includes compositions from images that are both present in the dataset [19]. Such approaches have a very solid basis in the observation that splices where at least one of the parent images is available on the Web are common in practice. However, we can never exclude the possibility that no source image can be located, e.g. because they are no longer online, are not traceable by our search engine, or were never shared in the first place. A second consideration is our ability to crawl the Web for similar images. Currently, near-duplicate collection from the Web can only be achieved using third party services such as Google or TinEye reverse image search. The automatization of such an approach carries an economic burden which is not negligible. While these approaches constitute a very promising future research direction in the field, in this work we focus on splicing detection algorithms based solely on a single image, which in many cases provide the sole basis for journalistic investigation.

2.2 Datasets

While, as in many other fields, proposed algorithms are often evaluated on custom, one-off datasets, there also exist a number of public datasets of forged images which can serve as benchmarks for algorithm evaluation. Table 1 presents a list of today's major spliced image datasets and the characteristics of their content.

When considering the usefulness of experimental datasets in evaluating splicing detection algorithms, a number of characteristics are critical. The first and foremost is the presence of ground truth binary masks localizing the splice. Without the presence of such masks, we can only evaluate forgery detection algorithms, and not forgery localization algorithms, and thus the scope of the dataset is severely limited. Another issue is the file types contained in it. From this perspective, the CASIA v2 dataset, which is the largest realistic dataset currently available, suffers from a severe disadvantage: instead of ground truth masks, the dataset only provides information on which two source images were used to form each splice. As a result, reliable ground-truth masks for the entire dataset can only be acquired through a semi-automatic process, which, given the size of the dataset, would be extremely demanding. A second characteristic is the dataset's image format. The advantage of lossless formats such as TIFF and PNG is that they may allow for uncompressed files, thus maintaining the most sensitive traces necessary for noise-based and CFA-based methods. On the other hand, JPEG-based algorithms are not expected to work on such datasets, unless the images have a pre-history as JPEG and were consecutively decompressed and encoded in a lossless format. In that case many JPEG-based algorithms might still work. In reality, especially for the Web-based forensics case, despite the recent proliferation of PNG files, JPEG remains the norm: it is indicative that among the contents of the Common Crawl corpus,[2] 87 % of identifiable image suffixes correspond to JPEG (.jpg, .jpeg). In our own attempts at collecting forged images for

[2] http://commoncrawl.org/

Table 1 Benchmark spliced image datasets and their characteristics. With respect to ground truth masks, "Manual" means that coarse guidelines were provided, which required manual processing for precise localization. "Limited access" means that the masks are not accessible, and instead a submission system accepts binary mask estimates and returns an overall per-pixel F1 retrieval measure for the entire dataset

Name	Acronym	Format	Masks	Realistic	# fake/authentic
Columbia Monochrome [12]		BMP grayscale	Yes	No	933/912
Columbia Uncompressed [28]	COLUMB	TIFF	Yes	No	183/180
Fontani et al. Synthetic [25]	FON_SYN	JPEG	Yes	No	4800/4800
Fontani et al. Realistic [25]	FON_REAL	JPEG	Yes (Manual)	Yes	69/68
CASIA TIDE v2.0 [7]		JPEG, TIFF	Yes (Manual)	Yes	5123/7491
First IFS-TC Image Forensics Challenge[a], Training [47]	CHAL	PNG (with possible JPEG history)	Yes	Yes	442/1050
First IFS-TC Image Forensics Challenge, Phase 1 Testing [47]		PNG (with possible JPEG history)	No	Yes	5713 unlabeled
First IFS-TC Image Forensics Challenge, Phase 2 Testing [47]		PNG (with possible JPEG history)	Limited access	Yes	350/0
Carvalho et al. [17]	CARV	PNG (with possible JPEG history)	Yes	Yes	100/100
Wild Web dataset [54]		JPEG, PNG, GIF, BMP, TIFF	Yes	Yes	10,646/0

[a] http://ifc.recod.ic.unicamp.br/fc.website/index.py

the Web, roughly 95 % of the images encountered were in JPEG format -this means that algorithms exploiting JPEG features are more likely to prove useful in our target scenario than algorithms requiring uncompressed images. In Table 1, it can be seen that only the two datasets of [25] and the CASIA v2.0 dataset [7] contain JPEG images, in addition to our own Wild Web dataset. Finally, the last important characteristic in a forged image dataset is the quality of the splices. Some datasets contain forgeries that have been artificially created by automatically modifying part of an image, or inserting a part of one image in another. Out of the datasets listed in Table 1, the Synthetic dataset of Fontani et al. [25] and the two Columbia datasets [12, 28] fall in this category. The former contains images from which the central square has been extracted, modified (e.g. recompressed) and placed back in the image. This means that the splice does not follow any semantic pattern (e.g. object boundaries) in the image, but is entirely artificial. In the Columbia datasets, on the other hand, the splices are parts from one image pasted into a different one, but still do not correspond to any meaningful shape within the images. Instead, a random area is spliced each time, where the splicing boundaries are abrupt and have not undergone any realistic-looking post-processing. Figure 2 shows samples from the Synthetic dataset of [25] and the Columbia Uncompressed set, alongside the other three that we chose for our evaluations. Generally, artificial datasets may be easy to create or extend, and are well-suited to evaluating specific aspects of localization algorithms, but their characteristics may differ significantly from what we typically encounter in practice. On the other hand, Columbia Uncompressed contains TIFF uncompressed images, which allow us to study the degradation of performance of various algorithms as an image is consecutively resaved as JPEG. Also, the Synthetic dataset of [25] is well-organized in four distinct sub-groups, each

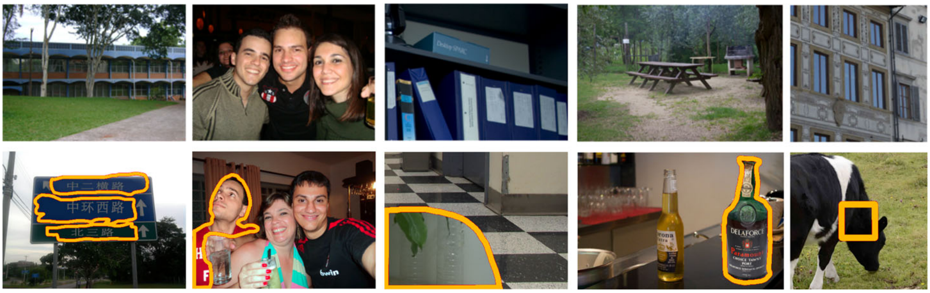

Fig. 2 Sample images from five benchmark datasets. The top row contains unspliced images, while the bottom row contains spliced images with the tampered area marked by us. From left to right, the datasets are: First Image Forensics Challenge [47], Carvalho et al. [17], Columbia Uncompressed [28], Fontani et al. Realistic [25], Fontani et al. Synthetic [25]

containing images that have been tampered with a different technique. The dataset contains splices detectable by different combinations of Non-Aligned JPEG quantization, Aligned JPEG quantization and JPEG Ghost in each subset. Both these datasets are of certain importance to our evaluations. On the other hand, we consider the Columbia Monochrome set to be of little relevance to our aims, as it consists of monochrome splices of very simple images, and do not include it in our evaluations.

The realistic datasets, on the other hand, have entirely different characteristics. The Realistic dataset of Fontani et al. [25] consists entirely of JPEG images and was aimed at evaluating a framework for detecting JPEG traces; it is thus expected to mainly exhibit these features. The datasets used for the First IFS-TC Image Forensics Challenge [47] constitute probably the largest realistic experimental set currently available, containing a large number of user-submitted forgeries. Due to the nature of the challenge, tampering masks are provided only for the training set, which is still considerably large. Besides the Wild Web dataset, this is the largest realistic image tampering localization dataset currently available, and it highlights all the current challenges of the field. While its images are saved as PNG, our preliminary analysis yielded some successes using JPEG-based algorithms, which means that at least some of the images were compressed as JPEG prior to being saved as PNG, and have retained certain traces from that step. However, the results of past evaluations are indicative of the general inadequacy of splicing detection algorithms to tackle this dataset. The currently top-ranking method [26] currently achieves a 0.45 F1-score in pixel-level localization, and does not actually use any fully automatic splicing localization algorithm. Instead, it is based on combining a dataset-specific PRNU estimation strategy, a copy-move localization algorithm, and a near-duplicate search to locate the sources from which the images were spliced. As a result, this method is not applicable in a real-world Web context, as a) the PRNU estimation only works if we have a constrained dataset, b) copy-move localization captures only a fraction of possible real-world forgeries, and c) developing an automatic near-duplicate search platform for the Web is currently extremely demanding in terms of resources, unless we purchase the services of a proprietary reverse search engine such as TinEye. Similarly, of the two most successful algorithms during the time of the challenge, the first [13] included a copy-move detection algorithm, a PRNU-based algorithm and one automatic splicing localization algorithm, while the second [51] only used a copy-move detection method. Thus it appears that the dataset remains particularly challenging for the splicing localization methods examined here.

Finally, the dataset of [17] consists of PNG images aimed to evaluate a method detecting illuminant color inconsistencies in human faces. However, preliminary analysis demonstrated that at least some images do contain detectable traces, including JPEG features. Thus, it was also deemed an interesting candidate dataset for evaluation, especially as it was not intentionally designed to highlight the performance of any of the traces we are investigating. This dataset is a subset of the IFS_TC Image Forensic Challenge, with images coming from both the training and test datasets. However, the dataset provides binary masks for a number of images that the Challenge dataset does not -namely, those belonging to "Test"-, and furthermore it was composed with an entirely different aim -face splicing localization- in mind. The fact that it only contains a specific type of forgeries may help highlight different aspects of the evaluated algorithms. Thus, we decided to include it in our evaluations as a distinct dataset.

Besides their differences, there is one more characteristic that is quite common for those image datasets: all the forgeries in the datasets were saved immediately following the splice, without undergoing further lossy recompressions, rescaling or other post-processing. This means that the forgery traces will be relatively intact and algorithms will have an easier time detecting them. There are two exceptions to this case: One is the First Image Forensics Challenge, where the images were gathered from user submissions and the precise image history is not disclosed. As a result, images in the respective datasets may have undergone some form of recompression or resampling following the forgery. This is a more realistic simulation of a forgery found on the Web, and may well be why this dataset is still particularly challenging for splicing localization algorithms.

2.3 Multimedia verification in the wild

As we examine the distinction between courthouse forensics and journalistic analysis of Web content, and the different use-case scenarios these entail, we see that they provide significantly different contexts for which algorithms should be evaluated. For example, PRNU-based methods are a very effective choice for the former case while being almost inapplicable in the latter. Likewise, the detection of any image alteration, even when it concerns a rescaling of the image, may be incriminatory enough in the former case. This is because the expectation in a legal investigation is that the submitted image is the camera-original, so even a seemingly harmless rescale may be masking other operations such as splicing or copy-moving. On the other hand in the case of Web sourced content, rescaling, resaving and even filtering of an image are rather common operations that are often applied automatically by publishing platforms and cannot be used to incriminate an image as a forgery.

In our preliminary work in this direction [54] we examined and reverse-engineered the operations performed by two popular social media platforms (Twitter, Facebook) to images uploaded to them. We observed that both platforms resave images as JPEG of medium-to-high quality, and also scale down images they consider too large, based on maximum dimension limits (2048 pixels) which, given today's camera phone resolutions, are rather restricting. While other services used by journalists to collect user-generated media, such as Bambuser,[3] do maintain image integrity, there exists a high probability that even innocuous images collected from social media or the Web have undergone such transformations.

[3] http://bambuser.com/

In our need to evaluate the state-of-the-art against the realistic Web-based forensic analysis task, we created an experimental dataset reflecting our specific needs: the "Wild Web dataset" [54] consists entirely of actual forgeries that have circulated the Web in the past years.

The dataset currently consists of 78 cases of confirmed forgeries. For each case, we used today's most popular reverse-image search engines (Google and TinEye) to collect as many near-duplicate instances as possible from the Web. We then applied bitwise comparison between the images to remove exact file duplicates, thus ending up with 14,478 images. For many cases, we found that multiple sub-cases exist, where it is not straightforward to identify which was the first forgery. In these cases, we separately kept each candidate sub-case (see Fig. 3 for an example), thus the dataset contains 86 such sub-cases. After manually filtering out versions that had been severely cropped, or obviously post-processed (e.g. by adding watermarks), we were left with 10,646 images, all containing confirmed forgeries -mostly splices, but including a few copy-move attacks. As the forging process was not known, and some cases contained multiple tampered areas/items, this meant that they could have taken place consecutively, and in this case it would be possible that only the last step of the process would be detectable. Thus, in creating the ground-truth binary masks for the set (with the help of the original, untampered sources, where available), in certain cases we created multiple ground truth masks, reflecting different possible steps of the operation –the total number of masks for all cases is 90. Of course, during evaluations, an algorithm output matching any of the possible masks of any sub-case should count as a success for the entire case.

As a result of the way it was formed, the Wild Web dataset is likely to contain certain versions of a forgery in which all traces have disappeared (e.g. rescaling, brightening and then resaving a tampered photo as an indexed GIF image will most certainly destroy all traces). However, due to the exhaustive nature of our search, it is also likely to include the first posting of the forgery, or at least several early versions of it. Thus, besides serving as a benchmark for evaluating detection algorithms, it also essentially presents us with a representative real-world sample from the Web concerning a large number of forgeries, i.e. how many variants exist, how heavy their degradation is, and in how many instances.

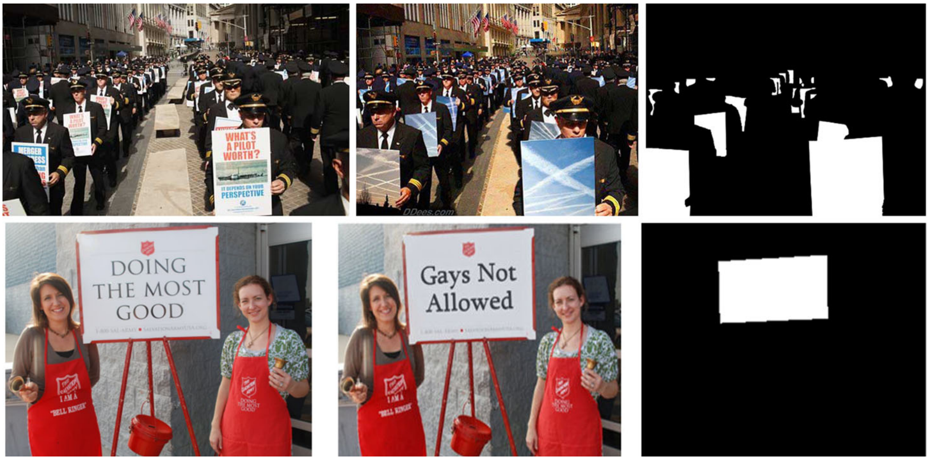

Fig. 3 Two forgeries from the Wild Web dataset. Left: original unspliced image. Middle: spliced image. Right: the binary mask we created from comparing the two

The major limitation of the dataset is the absence of a corresponding untampered ground-truth set, with which to evaluate against false positives. However, we came to the conclusion that it would not be possible to create an image dataset with similar characteristics, which would exclusively contain untampered images. The basic characteristic of the Wild Web dataset is that the images have been collected in bulk from the Web, and feature huge variability with respect to the number of resaves they have undergone, as well as to format changes, rescales, filtering and histogram adjustments, in addition to the actual splice. On the one hand, any dataset of verified untampered images that we could form from select images would in most probability not follow a similar distribution of such characteristics. On the other hand, if we instead attempted to randomly collect images from the Web, we would never be certain that the dataset does not contain splices, even in the form of watermarks that websites often attach over images published on them. Thus, the Wild Web dataset only contains forged images, and evaluation protocols ought to take this into account.

3 Evaluations

Our aim is to offer an exhaustive, comparative, cross-dataset evaluation of the state-of-the-art in image splicing localization, with an eye to Web image content. We have thus acquired or reproduced implementations of today's most popular and well-cited algorithms, and applied them, in a structured manner, to the dominant splicing detection datasets available today. We have also made the MATLAB source code of the algorithms and the evaluation framework publicly available on GitHub,[4] in order to allow researchers to replicate our results, incorporate novel algorithms in the existing framework, and foster further research under a common evaluation methodology. In this section we present the implemented algorithms, the datasets to which we applied them, the evaluation methodology that we adopted and the obtained experimental results.

3.1 Algorithms

In selecting algorithms for our evaluations, we made a number of choices for reasons of focus and resource usage. These choices were argued in depth in Section 2.1, and can be summarized in the following: a), we did not consider copy-move detection algorithms; b) we did not consider PRNU methods, since we currently consider it impossible to meaningfully apply them on images collected from the Web; c) we focused on forgery localization and not forgery detection. The 14 algorithms we used in our experiments are presented in Table 2.

These 14 algorithms represent, to the best of our knowledge, the state-of-the-art in image splicing localization. They cover the full extent of tampering traces that we are aware of (again, excluding PRNU noise) and have given promising results at the time of their publication. ELA is the only exception as it has not been academically published; it is, however, extremely popular among practitioners. In order to do each algorithm justice, we attempted to acquire the implementations from the authors. However, this was not always possible, thus DCT, ADQ1, ADQ3, ELA, GHO and NOI1 were implemented by us, BLK was re-implemented by us due to the increased computational complexity of the original implementation (but using it as a guide), while the rest were provided by the authors. Out of these, ADQ2, NADQ and CFA1 are publicly available,[5] while the rest were acquired through direct correspondence with the

[4] https://github.com/MKLab-ITI/image-forensics/tree/master/matlab_toolbox

[5] https://iapp.dinfo.unifi.it/index.php?page=source-code_en

Table 2 Splicing localization algorithms used in the evaluations

Acronym	Description	Ref.
DCT	A simple, fast detection method for inconsistencies in JPEG DCT coefficient histograms.	[52]
ADQ1	Aligned Double Quantization detection using the image DCT coefficient distribution. The authors further propose taking the local probability map produced, and using it to train a binary SVM to return a scalar probability on the image being tampered. We bypass this step and directly evaluate the local probability map.	[37]
ADQ2	Aligned Double Quantization detection claiming to improve upon the performance of ADQ1 by first estimating the Quantization table used by the previous compression. JPEG files only.	[5]
ADQ3	Aligned Double Quantization detection using SVMs trained on the distribution of DCT coefficients for various cases of single vs double quantization. JPEG files only.	[1]
NADQ	Non-aligned Double Quantization detection from the image DCT coefficients. JPEG files only.	[4]
BLK	Detection of disturbances of the JPEG 8 × 8 block grid in the spatial domain.	[35]
ELA	Error Level Analysis, aiming to detect parts of the image that have undergone fewer JPEG compressions than the rest of the image.	[33]
GHO	JPEG Ghosts, aiming to identify parts of the image, in which past recompressions were at a different quality than the rest of the image.	[22]
CFA1	Disturbances in the image CFA interpolation patterns, modeled as a mixture of Gaussian distributions. As the algorithm requires knowing the CFA filter pattern used by the camera, we took the CFA filter estimation algorithm of [21] and used it here as well.	[23]
CFA2	Estimation of the local error between the image and a re-interpolated version of it, following a simulation of the CFA filtering process.	[21]
CFA3	Isolation of image noise using de-noising, and comparison of the noise variance between interpolated and natural pixels.	[21]
NOI1	Modeling of the local image noise variance by wavelet filtering.	[41]
NOI2	Modeling of the local image noise variance utilizing the properties of the Kurtosis of frequency sub-band coefficients in natural images.	[40]
NOI3	While not strictly noise-based, this algorithm computes a local co-occurrence map of the quantized high-frequency component of the image. It then uses Expectation-Maximization to fit a two-class model on the distribution, assuming that the spliced area will have different characteristics from the rest of the image.	[16]

authors. All implementations, both our own and those provided, were in MATLAB. Furthermore, we did our best to adapt them to the needs of the evaluation to ensure the best possible performance. Thus a) although ADQ1 and DCT are strongest when drawing the DCT coefficients directly from the JPEG encoding, they can also operate on images that are no longer in JPEG format, by estimating the DCT coefficients from the decompressed image, b) CFA1 takes advantage of the CFA pattern estimation from [21] since the original implementation assumed that the standard Bayer CFA array was used in capturing image, and c) ADQ3 first estimates the JPEG quality from the file's quantization tables, in order to choose the appropriate SVM classifier for that quality.

In section 2.2, examination of existing evaluation datasets for image splicing localization made clear that only a subset of today's datasets are appropriate for our task. Hence, we only chose datasets of color images providing ground-truth binary masks. These include Columbia Uncompressed (*COLUMB*) [28], Fontani et al.: Synthetic (*FON_SYN*) [25], Fontani et al.: Realistic (*FON_REAL*) [25], Carvalho et al. (*CARV*) [17], First IFS-TC Image Forensics Challenge: Training (*CHAL*) [47], and the Wild Web dataset [54]. With respect to the *CHAL* dataset, as it contains many copy-move forgeries, it would be unfair to expect splicing localization algorithms to operate on those cases. Thus, a copy-move localization algorithm was run over the entire dataset [15], and the results were visually verified in order to identify such forgeries. Consecutively, 136 confirmed copy-move forgeries were removed from the dataset prior to our evaluations.

3.2 Evaluation methodology

In the process of our evaluations, all implemented algorithms were applied on the images of all datasets. Each algorithm produces an output map that can be used to localize tampered areas in the image. One typical evaluation methodology is based on comparing the values of the output map under the region defined as "tampered" by the ground-truth mask, with the values of the map in the rest of the image. One example is [25], where the medians inside and outside the mask are compared. If the absolute difference between the two values is above a certain threshold, the image is classified as tampered; otherwise it is classified as authentic. Another approach is the one in [22], where the Kolmogorov–Smirnov statistic is used to compare the value distribution in the two regions. In that case, the K-S statistic provides a metric of the similarity between the two regions' distributions, and can again be compared against a threshold to classify an image as tampered or authentic.

A characteristic of such evaluation approaches is that they do not inherently include a means of estimating false positives/true negatives. As each evaluation is bound to the existence of a ground truth binary mask, untampered (i.e. "Negative") images, for which the mask should consist entirely of zeroes, cannot be tested at all. A common solution to this is to form a test mask marking an arbitrary region in the image (such as a square in its center) and test all untampered images against that map. This method serves as a baseline against overestimating the performance of algorithms in a basic way: if the comparison threshold is too low, all tampered images would be found to be successful detections, since it is unlikely that e.g. the median within the mask would be perfectly equal to the median outside it. However, in the case of such a small threshold, any arbitrary region (including the square we defined) would also give a positive result, even for untampered regions. Thus, while not actually able to detect all possible false positives in an output map, this practice at least balances the possibility of random outputs passing as True Positives, by producing False Positives from similarly random outputs. Figure 4 shows three examples, their binary mask areas and the corresponding difference between means. In this case, the first and last examples give very confident conclusions (a true positive and a true negative respectively), while whether the middle example will be classified as a true positive or false negative will depend on the threshold value.

On the other hand, such an approach fails to guard us against actual false positives: it is possible that some images may contain certain spatial image features (e.g. oversaturation or high-frequency textures) that might misguide certain algorithms by returning a local positive in regions where there is none. Since the proposed false positive detection approach only looks for discrepancies in a specific, arbitrary region in the output, false positives popping up elsewhere will most likely not influence the result, and the evaluation will return a true negative.

An alternative evaluation methodology would be to binarize the output map and compare each pixel with its corresponding value in the ground-truth mask. This approach allows per-pixel estimation of success metrics, is far more precise in terms of evaluating localization quality, and inherently solves the issue of false positives. This

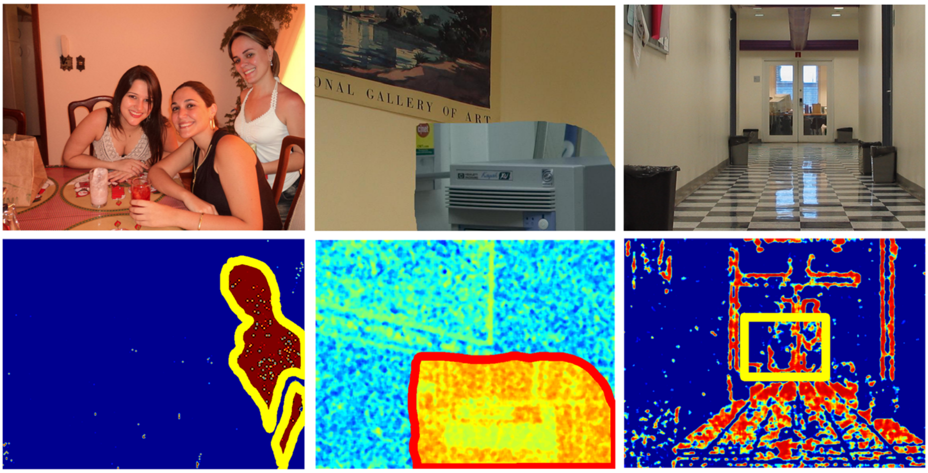

Fig. 4 An example of output map evaluations. Top: two spliced and one unspliced image. Bottom: the corresponding output value maps for using different algorithms (ADQ1, CFA1 and CFA1), with the borders of the binary mask used for the evaluations manually marked. For the forged images, this corresponds to the actual spliced area, while for the last column it corresponds to an arbitrary rectangle in the center of the image. In these examples all algorithms produce output in [0, 1] and the absolute difference of medians between the masked region and the rest of the image is, respectively: 1.0, 0.282 and 0.006

approach was followed in the Tampering Localization (Phase 2) of the First Image Forensics Challenge. However, one inherent disadvantage in this approach is that it requires the output map to be thresholded prior to any evaluation. Another disadvantage is that it cannot be easily applied to untampered images, as such images consist entirely of pixels labeled Negative, and it is difficult to devise metrics to simultaneously evaluate tampered and untampered images. This was not an issue for the Challenge, as Phase 2 only included tampered images, but would be problematic when applied to datasets consisting of both tampered and untampered images. In our evaluations, we used both evaluation practices, depending on the dataset: a) for the established benchmark sets containing both tampered and untampered images we used the first approach of comparing the value distributions inside and outside the mask, with artificial square maps for negative examples and b) for the Wild Web dataset which does not contain untampered images we used the second approach, of binarizing the outputs and evaluating whether the outputs match the ground truth masks.

3.3 Results

3.3.1 Datasets containing both tampered and untampered images

For existing reference datasets, we adopted the approach of assuming the existence of an artificial ground truth mask for each untampered image which, similar to [22, 25], corresponded to a block of size ¼ of each dimension, placed in the image center. As presented in Section 3.2, this is more appropriate for datasets that contain both tampered and untampered images, while the approach of using pixel-wise retrieval metrics is more fitting for datasets that only contain spliced images, such as the Wild Web dataset used in the next subsection. As a measure of dissimilarity between the output values distribution within the mask and outside it, the Kolmogorov-Smirnov statistic is calculated for each image:

$$k = max_u |C_1(u) - C_2(u)| \tag{1}$$

In Eq. (1), $C_1(u)$ and $C_2(u)$ are the cumulative probability distributions inside and outside the mask respectively. If the metric surpasses a threshold, then a positive is declared, i.e. a forgery is detected. By then shifting the threshold for each algorithm, and evaluating how many images return positives in the tampered and untampered subsets, we get the True Positive and True Negative value for each threshold, which we use to form the ROC curves for each dataset, presented in Fig. 5. Three of the algorithms of Table 2, namely ADQ2, ADQ3, NADQ, can only accept JPEG images as input, as they exploit certain JPEG data directly extracted from the compressed files. Specifically, ADQ2 and NADQ require the rounding error caused by decompression, while ADQ3 requires the JPEG quality under which the image was stored. In images that were compressed as JPEG in the past, but are now decompressed and stored in lossless formats, it is now impossible to retrieve these parameters with precision. As a result, these algorithms were only tested on datasets containing JPEG images (FON_REAL, FON_SYN). The rest of the algorithms exploiting JPEG-based traces (DCT, ADQ1, ELA, GHO and BLK), do not actually need the image to be currently stored in JPEG format, and only expect the image to have been stored as JPEG in its past. Thus, these JPEG algorithms were applied on all datasets.

In interpreting results, one should generally be wary of the possibility of overestimating algorithm performance. The most important source of such overestimations arises from the arbitrary shape of the binary mask used to evaluate True Negatives. In most cases, the real binary masks used for tampered images correspond to an actual physical object in the image. On the other hand, the artificial mask we use for untampered images corresponds to an arbitrary part of the image. As a result, there is generally a higher chance of random noise -corresponding to an image object- accidentally causing a True Positive than causing a True Negative. This effect is stronger at low threshold values, while for more strict values results are relatively more reliable.

The two non-realistic datasets, i.e. COLUMB and FON_SYN help highlight a number of aspects of the algorithms under evaluation. Perhaps the most striking result is the effectiveness of noise-based and CFA-based algorithms on the Columbia dataset. Indeed, all algorithms perform well even at a 0 % False Positive rate, with the CFA algorithms and NOI3 demonstrating superior performance that reaches almost 70 % detection rate at 95 % True Negatives. CFA1 and NOI3 are significantly more effective at extremely high TN rates, which implies higher robustness, but both CFA2 and CFA3 catch up as the threshold relaxes. The performance of JPEG-based algorithms, on the other hand is, as expected, significantly lower. However, we should observe here that none of the images in the dataset have ever undergone JPEG compression, and thus no JPEG traces should be found anywhere in the image. Therefore, theoretically JPEG algorithms should not be able to detect the splicing in any case and thus the performance of such algorithms should be zero, at least when True Negatives are high. Figure 6 provides one explanation for this phenomenon. In the figure, we can see the output of the algorithm we label as BLK, in an uncompressed image of the Columbia dataset. Due to the differences in content variance (and possibly high-frequency noise), the spliced region returns very different values, although we are certain there are no JPEG blocking

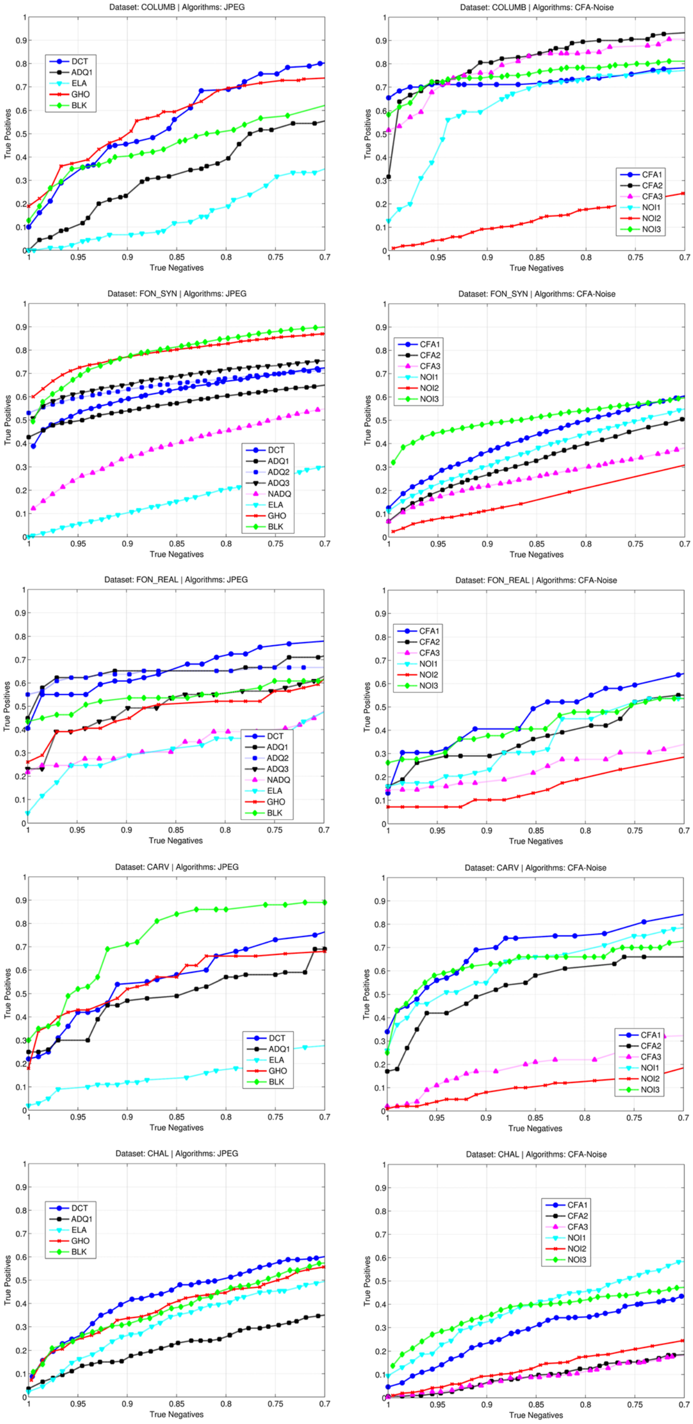

Fig. 5 ROC curves for all algorithms applied on the benchmark datasets. As for low TN rates the evaluation criteria become generally uninformative, algorithm performance is only presented up to a True Negative rate of 70 %, but we consider performance at above 95 % TN to be the most important indicator

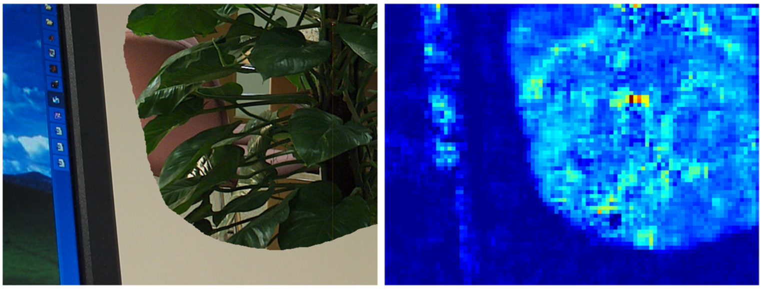

Fig. 6 Application of the JPEG-based BLK algorithm on an image that has never undergone JPEG compression. While we can be sure that there are no actual traces of JPEG blocking for the algorithm to detect, the algorithm still correctly localizes the tampered region, due to different image content characteristics in the spliced region

artifacts to be found anywhere in the image. Generally, when designing an evaluation study, it might seem reasonable not to even consider evaluating JPEG-based algorithms on uncompressed images. However, we decided to include these evaluations in our study, as they produce rather interesting results from a user perspective. In a real-world scenario, this would be the actual output a user would encounter –a correct localization, for the wrong reasons. It is open for discussion whether such behavior from a localization algorithm increases or decreases its utility for end users.

In this aspect, the performance of algorithms on the Synthetic dataset of Fontani et al. (FON_SYN) is a more reliable indicator of forgery detection performance. As the tampered area does not coincide with any visual phenomena, it is less likely to produce spurious true positives. JPEG-based algorithms perform reasonably robustly, especially if we take into account the internal splitting of the dataset into subsets of images that have undergone different splicing procedures (e.g. Aligned, Non-Aligned), not all of which can be detected by all algorithms. What is interesting, on the other hand, is that CFA and noise algorithms also perform significantly better than random in this dataset. There exist a small but not insignificant set of cases that are accurately detected using these algorithms, which most likely suggests that not all relevant traces (e.g., CFA interpolation patterns) disappear during lossy compression. This is most prominent in NOI3, which yields a large number of clear TP detections at the 100 % TN rate. This is a very important feature of the algorithm, as it shows that, while based on spatial information, the algorithm is equally effective at locating the effects of different compression histories, even in cases where no other modification has taken place.

Algorithm performance in the Realistic dataset of Fontani et al. (FON_REAL) follows similar patterns. With the exception of BLK and ADQ3 which seem to perform significantly better in the synthetic cases than in the realistic ones, most algorithms show comparable performance to the one achieved in the synthetic set. Non-JPEG algorithms actually perform slightly better on the realistic set, which is partly to be expected as the realistic photo editing process is likely to have generated more pattern discrepancies than the automatic generation process used for creating the synthetic set.

As was mentioned above, the CARV dataset was created to test an illuminant-based face splicing detection algorithm, and we were interested in evaluating whether traces detectable by the algorithms tested here had unintentionally remained –this created a relatively more realistic scenario in comparison, e.g. with FON_REAL, where forgeries were also realistically created, but explicitly in order to test some of the JPEG-based algorithms evaluated here.

Results were in fact interesting for both sets of algorithms. JPEG-based methods gave more than 20 % recognition rates, with the exception of ELA which performs significantly worse, achieving zero correct detections at 100 % TN and reaching about 10 % True Positives at the 95 % TN mark. On the other hand, CFA1 and NOI1 achieve a True Positive detection of about 50 % for 95 % True Negatives. However, looking more closely, one important observation can be made on the behavior of the CFA1 algorithm. Figure 7 shows one image from the CARV dataset that is successfully detected even at a 100 % True Negatives threshold. While the overall value distribution does distinguish between the tampered area and the rest, especially if we already know the mask, the algorithm produces a very spurious output. CFA1 produces probabilistic per-block values in the range [0, 1]. It may be problematic that in this example the mean value within the mask is around 0.6 while the mean outside the mask is around 0.5, and in many regions it locally deviates even above 0.65. While, after the fact, visual inspection of the output mask intuitively gives the sense of a correct detection, the extent to which this raw output would be informative to a human analyst is an open question. This is important as practically all correct detections achieved at very low False Positive rates in this dataset feature similar ambiguities.

The final dataset used was the training dataset from the First Image Forensics Challenge (CHAL). The observed results clearly demonstrate why, in the presence of so many image splicing localization methods in literature, the most successful proposals during the Challenge were in fact based on other types of information, such as clustering the PRNU fingerprints in the dataset, or copy-move detection. Indeed, there are very few detections for most algorithms at TN = 100 %, and as we shift the threshold, the TP detection rate increases very slowly alongside the TN rate, suggesting that no clear, visible localizations are actually achieved. NOI1, ADQ1, BLK and GHON yield roughly 10 % True Positives for TN = 100 %, with a notable exception in NOI3, which, while still far from yielding desirable performance, clearly outperforms all other algorithms.

Following these first-level performance evaluations, we turned our interest to the particularities of the problem we are currently focused on: Web images that reach our hands have commonly undergone at least one further JPEG resave, either by some intermediate user, or automatically by the publishing platform. Thus, it would be important to evaluate the robustness of existing algorithms against this assumption –we can generally assume that a further resave will lead to performance degradation, but different algorithms should be expected to exhibit different levels of robustness.

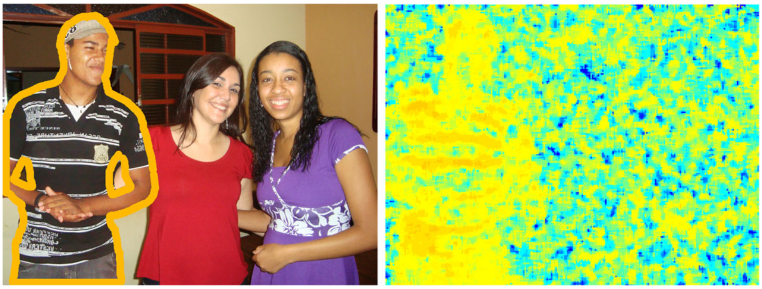

Fig. 7 An image from the CARV dataset and its output using CFA1. Darker hues correspond to lower values. The mean probability of tampering within the mask is around 0.6 while the mean outside the mask is around 0.5. However, there are many local deviations, especially with some high-value regions outside the mask

Thus, all images from the datasets were resaved five times, at qualities 100, 95, 85, 75 and 65, and the algorithms were run again on the resulting images. For compactness, we do not present the ROC curves for each algorithm, but instead estimate the threshold value for which the algorithm returns a true negative rate of 95 %, and calculate the percentage of true positives for the same threshold. Results are presented in Fig. 8.

Perhaps the most striking observation is the way the ELA and Ghost algorithms behave in the Columbia Uncompressed dataset. In the uncompressed dataset, the algorithm expectedly achieves practically no detections, which is reasonable as the images are not supposed to feature any JPEG traces for these algorithms to detect. However, following recompression, and especially at a high quality, algorithm performance sharply increases. These two algorithms are supposed to identify different compression histories in different parts of the image, which clearly does not apply in this case. What instead happens is that they end up operating as a noise filtering method, and essentially localize different noise content in different parts of the image, either as an overall phenomenon, or on specific image edges. Thus, in Fig. 9 left, we see that by applying ELA on an image from Columbia, the spliced region returns an overall higher residual, which is obviously not due to different JPEG histories, but rather due to different image content. On the right, we instead see that edges in the spliced region are a lot sharper than in the rest of the image, and cause a very different value distribution within the mask than outside it.

The reason this phenomenon only appears in re-compressed images is due to the fact that, when applying ELA on the uncompressed image, we subtract a JPEG-compressed version of the image, which has been inevitably chroma-subsampled. The difference between the original and the subtracted image is thus dominated by the effects of chroma subsampling. When applying ELA or Ghost on images that have already been compressed at least once, even at quality 100, chroma subsampling has already taken place, and the comparison focuses on the local discrepancies caused by rounding and DCT quantization. This observation sheds light on another aspect of this algorithm. While its aim is to identify local discrepancies in the image compression history, it seems that, at least for high JPEG qualities, some percentage of the output is due to different noise content, potentially caused by different capturing conditions (e.g. device or settings). We can be sure of that, as all images in the Columbia dataset have identical (i.e. zero) compressions in their history. Thus, noise content is the only aspect in which each splice differs from the recipient image.

A second interesting observation we can make is that, in the three datasets where JPEG based methods perform relatively well (FON_SYN, FON_REAL, CARV), while their performance generally degrades as the recompression quality drops, it does not do so monotonically –instead, we encounter small increases, especially at qualities 95 to 75. This especially applies to the double quantization detectors, ADQ1, ADQ2 and NADQ. The reason this happens is that these datasets contain JPEG images at various qualities. For most algorithms, a resave even at the same quality will cause a degree of feature degradation due to rounding errors. For the double quantization algorithms, a resave at the same quality will leave the detection result almost unchanged –thus, resaving an image of quality 85 at quality 95 may prove more detrimental for double quantization traces than actually resaving it at quality 85.

Apart from that, algorithm behavior is generally as expected, in the sense that performance degrades as the JPEG compression quality drops. It should be noted however, that the overall effect of resaving up to quality 95 does not seem to be as damaging as one would expect, even for the sensitive CFA algorithms. Indeed, most algorithms seem to retain their performance up to that level in all datasets. However, for qualities 85 and below, performance drops significantly for all datasets.

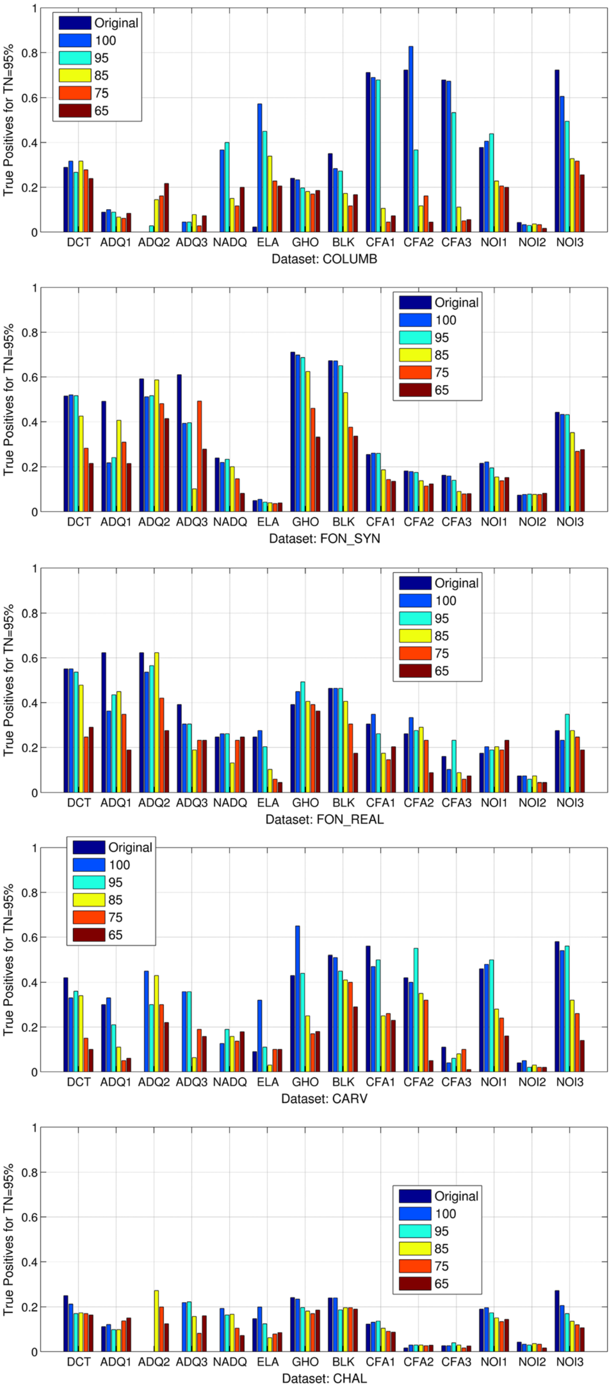

Fig. 8 Evaluation of the effects of JPEG resaves of various qualities on splicing detection. For ADQ2, ADQ3 and NADQ, no value is given for the original images in datasets COLUMB, CARV, and CHAL, as they do not contain images in JPEG format

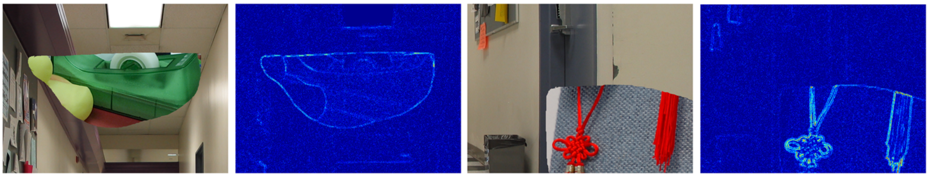

Fig. 9 Application of ELA on spliced images formed from uncompressed sources that have undergone one (uniform) recompression at Quality 100. In the left case, the entire spliced region features a higher residue than the rest, possibly due to different source noise. In the right, edges inside the spliced area leave much more prominent residue than edges outside it

Overall, Fig. 8 offers a comparative view of the current state-of-the-art: with respect to realistic data, at a reasonable True Negative rate of 95 %, success rates range from about 50 % to a maximum of 62 % for ADQ1 and ADQ2 in FON_REAL. While this performance is modest in itself, success rates in the more complex CHAL dataset are at best around 20 %, which, given the difference between the masks used for positive detection and negative detection, may be rather close to random –the only possible exception being NOI3. The same applies to images recompressed below a certain quality, generally at 75 % for JPEG methods and 85 % for noise and CFA-based methods. As we observed in our past investigations [54], social media platforms resave images at qualities as low as 70, which seem to render many algorithms entirely ineffective. Furthermore, social media platforms also rescale large images, a process that, in theory, will cause even more intense damage, as the resampling will most likely destroy most tampering traces. In order to evaluate the extent of damage caused by rescaling, we ran a focused evaluation on a random subset of each dataset consisting of 100 tampered and 100 untampered images each –with the exception of FON_REAL which consisted of less than 100/100 images to begin with. This focused approach ought to give us a good estimate of the damage caused by rescaling, without having to run the evaluations over the entire data. The selected images were rescaled at 95 %, 75 % and 50 % their original size, and resaved at JPEG quality 90.

Figure 10 shows the effects of rescaling. Again, the rate of True Positives is given for TN = 95 %. It is clear that, in the vast majority of cases, regardless of the degree of scaling, the resampling procedure destroys most tampering traces in the image.

3.3.2 Computational cost

Besides the detection performance, one other aspect of tampering localization algorithms that we have to take into account is their computational cost. The 14 algorithms evaluated here differ significantly in this aspect, and we chose an empirical approach to evaluating the cost of each algorithm: 200 images were randomly chosen from all datasets, and the average time (in seconds) taken by each algorithm for a single image were measured and are presented in Table 3. It can be seen that the results differ significantly, from the near-zero cost of ELA to the significant times taken by ADQ3, CFA3, NOI3. In the case of the ADQ3, this is because in our implementation the image is segmented into a very large number of overlapping blocks, each of which has to be processed independently. In the case of the CFA3, the calculation of the local features is a demanding process. For both these algorithms, we could have opted for a non-overlapping segmentation, but this would have significantly reduced the localization accuracy due to the coarseness of the result. In all cases, it should be taken into account that the results presented are implementation- and dataset-dependent. While all efforts have been

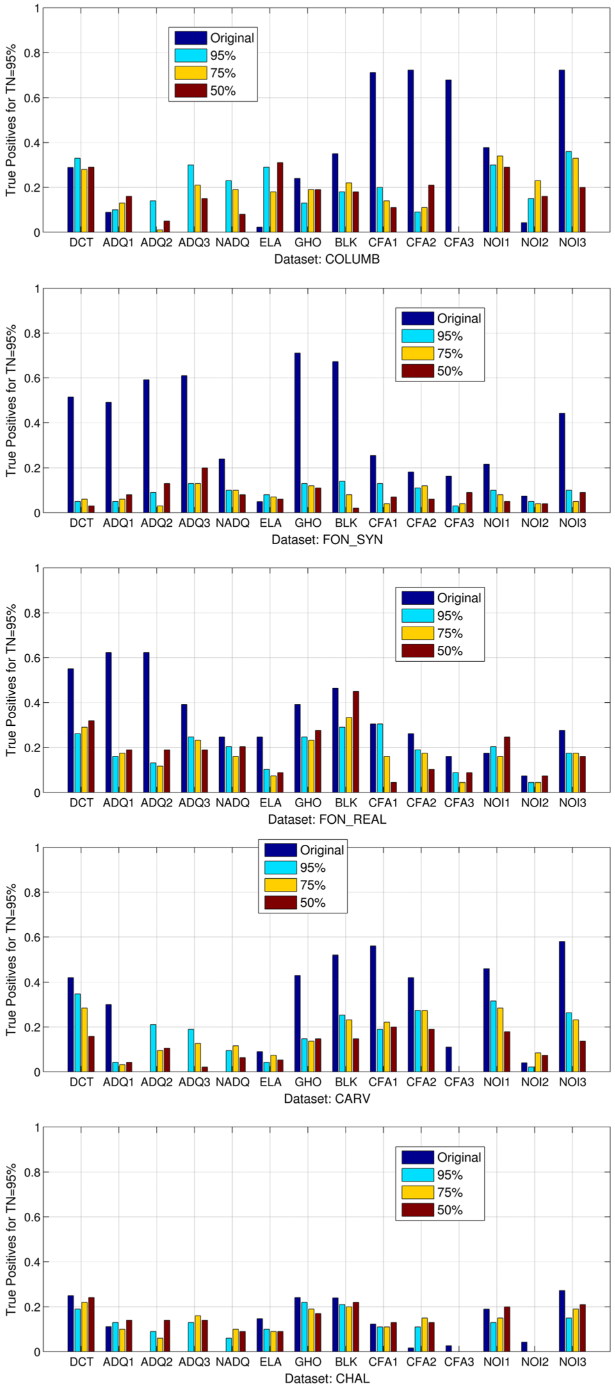

Fig. 10 Evaluation of the effects of JPEG rescales for various scales on splicing detection. For ADQ2, ADQ3 and NADQ, no value is given for the original images in datasets COLUMB, CARV, and CHAL, as they don't contain images in JPEG format

Table 3 The average time taken by each of the 14 algorithms to process a single image

Algorithm	ADQ1	ADQ2	ADQ3	NADQ	ELA	GHO	DCT	BLK	CFA1	CFA2	CFA3	NOI1	NOI2	NOI3
Time/image (seconds)	0.24	0.28	14.33	1.32	0.06	4.72	0.08	1.46	3.60	2.85	20.91	0.10	2.07	7.16

made to make our own implementations as efficient as possible, it is possible that faster implementations are feasible. Furthermore, while we have tried to take a representative sample of the available image size distribution, most images are small (around 1-2MP), while modern capturing devices often produce images of 10MP or more. As a result, the numbers presented in Table 3 should be considered merely comparative.

3.3.3 The Wild Web dataset

Our investigation so far focused on a one-step modification and demonstrates how even a single resave at a medium-to-low quality can seriously hurt the performance of most algorithms. However, we are fully aware that the situation in the real world is far more complex than this. To this end, the final and most challenging set of evaluations of the state-of-the-art concern the application on the Web-based task we are interested in, in the form of the Wild Web dataset.

In subsection 3.2 we presented two different evaluation methodologies depending on whether a dataset also contains untampered images, or consists entirely of tampered ones. As the Wild Web dataset falls into the latter case, it is more appropriate to use the second approach presented, i.e. evaluation based on the binarization of the output masks and their pixel-wise comparison to the ground truth masks. Thus, following application of all algorithms on the dataset images, the output maps were binarized using multiple threshold values covering the entire range of possible map values. Each of the resulting binary maps was then processed using different combinations of binary morphological operations (opening and closing), thus generating multiple versions of the output binary map for each threshold value. Finally, a measure of mask similarity was calculated for each image, threshold value and mask version, counting the per-pixel retrieval performance:

$$E(A,M) = \frac{\sum \left(A \bigcap M\right)^2}{\sum (A) \times \sum (M)} \tag{2}$$

In Eq. (2), A signifies the binary, processed algorithm output and M is the ground-truth map, while $\Sigma(x)$ indicates the area of a binary mask x. Of all the E values generated for each image, we keep the highest for each algorithm, in order to reflect a human investigator's ability to spot regions that stand out in an output map due to their difference from the rest of the image, and combine the map output with semantic cues to deduce where the splicing has taken place. This measure was chosen in our previous work for its ability to filter out false detections. Experimentally, we consider any output map achieving an $E > 0.7$ as an accurate detection.

Table 4 Algorithm performance on the Wild Web dataset. *PENS* corresponds to a theoretical perfect ensemble classifier, here corresponding to the number of classes where at least one algorithm achieved detection. *Detections* corresponds to the number of classes where at least one image was correctly localized using the corresponding algorithm, while *Unique* corresponds to the number of cases detected exclusively by that algorithm

Algorithm	ADQ1	ADQ2	ADQ3	NADQ	ELA	GHO	DCT	h	CFA1	CFA2	CFA3	NOI1	NOI2	NOI3	PENS
Detections	3	8	2	1	2	12	1	3	1	0	0	3	2	7	18
Unique	0	1	0	1	0	2	0	0	0	0	0	0	2	0	-

As we described in Section 2.3, the Wild Web dataset is separated in a number of cases, for which we have identified various sub-cases, each with possibly multiple binary masks. In a realistic investigative scenario, any accurate detection in any of these should qualify as an overall success for the case. We thus evaluate the 14 algorithms at the level of the 78 unique cases, and seek instances, where an algorithm achieves at least one detection in a case.

Table 4 presents our results for the 78 classes. Besides the algorithm outputs, we added a column labeled PENS, corresponding to a theoretical Perfect ENSemble classifier (PENS), essentially summing up all successes from all algorithms and counting the number of classes where at least one successful detection took place. In analyzing the results, the most striking observation is the low performance of practically all algorithms on the dataset. Out of 78 cases, only 18 seem to be possible to detect correctly. However, even this value is likely an overestimation, due to the thresholding approach used: when thresholding output maps that do not produce probabilistic values but instead follow an arbitrary range (an issue which is most evident in noise-based methods), it is always possible to find some threshold for which the output map, having its values affected by the shapes present in the image, produces a binary map that

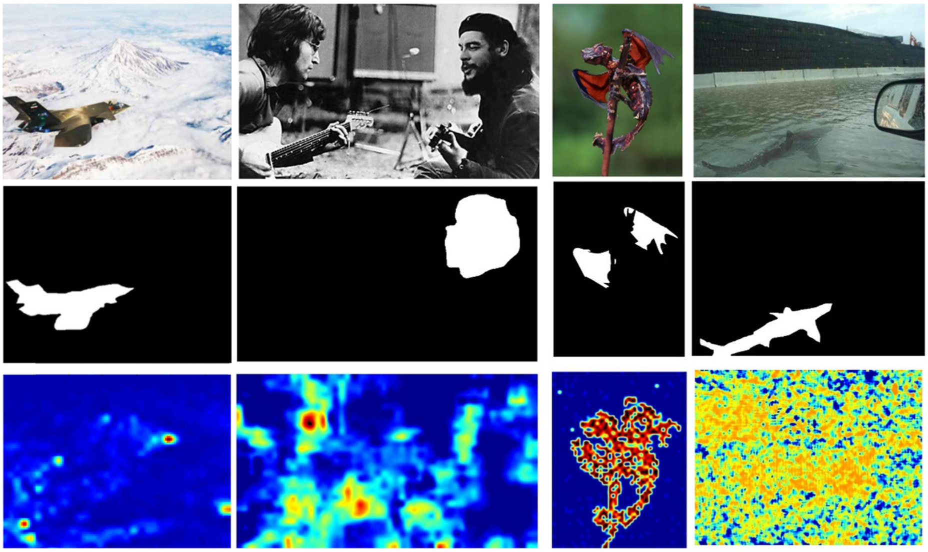

Fig. 11 Examples of failed detections from the Wild Web dataset. Top: the forged images. Middle: the ground truth masks. Bottom: algorithm output. From left to right: NOI1 (first column), DCT (second column), ADQ1 (third column), CFA1 (fourth column)

coincides with the spliced region. Essentially, we should be generally wary of claimed successes, with the possible exception of ADQ1, ADQ2 and CFA1 whose probabilistic output makes them relatively more reliable. As a result, we are faced with a serious challenge: in a real world application, since the majority of algorithms do not produce a probabilistic output that we can threshold at a reasonable value (say, 0.5 or 0.7), our only other option is comparing arbitrary thresholds. However, in the absence of negative examples, we are unable to evaluate the extent in which this results in overestimations, and is difficult to evaluate what a human investigator would deduce by visually inspecting the output maps produced.

Another general observation is the relative superiority of JPEG-based algorithms over CFA- and noise-based ones, with the exception of NOI3. Indeed, all the latter methods yield very few successful detections. On the other hand, double quantization and Ghost methods seem to dominate detections, while JPEG Blocking artifact detection seems to also contribute a number of unique detections. This is reasonable, since we are dealing with images that have undergone multiple modifications during their lifecycle, and the more sensitive traces such as CFA interpolation patterns are more likely to have been destroyed.

Figure 11 shows a few of the many failed detections from the Wild Web dataset, while Fig. 12 presents some cases of successful detections from the Wild Web dataset. Although the overall performance of splicing detections in the real world leaves a lot to be desired, it becomes clear from the results that, for some cases, there exist instances where splicing localization can work, and will provide an invaluable tool in assisting investigators. The issue remains open, however, on how to increase the robustness of such algorithms in real-world situations, and make them reliable and widely applicable for content found on the Web.

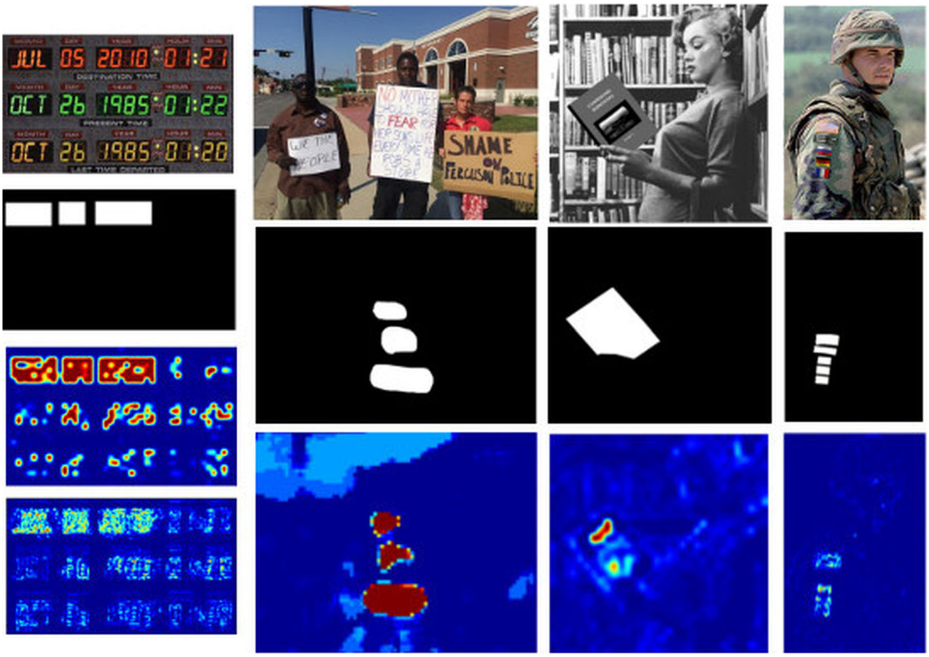

Fig. 12 Examples of successful detections from the Wild Web dataset. From left to right: ADQ1 and GHO (first column), ADQ2 (second), NO1 (third), and GHO (fourth column)

4 Conclusions

In this paper, we conducted a comprehensive evaluation of the state-of-the-art in splicing localization algorithms. Our work focused on the performance of verification systems for journalists in usage settings where the image content is sourced from the Web, which means that classes of algorithms that exploit explicit knowledge, such as knowledge of a device's PRNU patterns, were excluded. It also means that robustness to subsequent image alterations is a major consideration when evaluating algorithm performance, a fact which we attempted to emulate and take into account. Evaluations were also run on the Wild Web dataset, consisting of forged images collected from the Web, which is the closest benchmark we have to the real-world task.

One clear observation is the discrepancy between real-world data and the experimental datasets typically used in evaluations to date such as [25, 28]. With the exception of the dataset used in the First IFS-TC Image Forensics Challenge, all other datasets were found to be significantly easier than the Wild Web dataset, even for different algorithms than the ones they were designed for. On the other hand, the Challenge dataset proved to be an extremely hard benchmark, to the point of being tougher than some cases from the Wild Web dataset. This is potentially due to the images having undergone rescaling, thus losing most traces of the forgery, whereas the Wild Web dataset also contained some images that maintained their original forgery traces.

Another important observation is the sensitivity of some JPEG-based algorithms to non-JPEG features, such as noise. Algorithms based on blocking artifact aberrations or recompression residue (i.e. Block [35], Ghost [22], and ELA [33]) were able to localize forgeries even in cases where no JPEG compression had taken place –especially following one resave of the image to high-quality JPEG. It is arguable whether this can be considered a feature of such algorithms, or an undesirable or unreliable outcome. No such instance was observed for Double Quantization algorithms, as they operate directly on the low-frequency DCT coefficients and cannot locate high-frequency noise patterns.

In terms of performance, we can observe that, in many cases the differences between algorithms detecting similar traces were generally small and difficult to generalize. ADQ1 [37] performed slightly worse than ADQ2 [5], but is also computationally simpler, while the much more demanding ADQ3 [1] showed much better performance for the artificial cases than the realistic ones. CFA3 [21] was found in many cases to be relatively weaker and less robust than CFA1 [23] and CFA2 [21], despite the former's increased computational complexity. On the other hand, NOI3 [16] not only showed in most cases superior performance to other noise-based methods, but also proved to be an all-round detector, able to give good results on many uncompressed images, JPEG benchmark images and actual real-world cases. One final note to take into account is that, as we opted for a massive, automatic evaluation framework, it is possible that the efficiency of NOI2 has been relatively underestimated. This is because NOI2 depends heavily on a number of parameters which, while we made our best effort to calibrate optimally for all datasets, meant that the algorithm could have performed better if we had the option of calibrating the parameters for each image separately.

Finally, one striking observation is the unsuitability of ELA for automatic localization. Indeed, this partly expected outcome is due to the fact that ELA was designed for use by appropriately trained investigators, who would use broader clues to localize forgeries. Still, its algorithmic similarity to the Ghosts algorithm meant it merited an inclusion in the family of approaches tested here, and in fact led to the interesting observation on its behavior as a noise discrepancy detector for uncompressed images.

Overall, and with the Web image forensics task in mind, one could conclude that the field is far from mature for real-world use. In the vast majority of the real Wild Web cases, but also in the realistic Challenge dataset, most algorithms failed to detect any trace, most likely because the traces sought are too fragile to survive the typical image processing operations such images undergo. If splicing localization algorithms are to meet real-world application in the future, the field must take significant steps forward from its current stage, and multiple directions arise from our evaluations. One such potentially fruitful direction was mentioned in Section 2.1, and was also made apparent by our experiments on the Wild Web dataset: the combination of image forensics algorithms with a reverse image search engine. While the possibility of an algorithm properly localizing the forgery in any single image in the dataset was minuscule, we can simultaneously see that a number of cases contain at least one image where at least one localization algorithm gave an unambiguously correct detection. Thus, when presented with a potential forgery, a forensics system could first run a reverse search and collect all possible near-duplicates of the image, and run the forensics algorithms on all of them, in the hope that one will return a clear detection. Furthermore, application of methods such as multiple parenting phylogeny [19] could solve many additional cases, provided we tackle the burden of near-duplicate retrieval on the Web.

A second research direction would be to base our forensic analysis on trainable spatial features. Machine learning methods have been used in the past to successfully detect forgeries. It is indicative that, in contrast to the extreme difficulty of localizing forgeries in the Image Forensics Challenge datasets, near-perfect accuracy was achieved by more than one teams in the detection phase of the Challenge using such methods [14, 51]. The degree in which these methodologies can be converted to operate on a local scale and in a generalizable, non-dataset-specific manner is under debate, but this direction could lead to algorithms that are resistant to image modifications –even resampling. In the recent past, spatial features that have shown good performance for detection were extended to operate on a local scale [13], while the SpliceBuster (NOI3) [16] that we evaluated in this work is a further refinement of those algorithms. Its relatively good performance in our evaluations suggests that these approaches may be on the right path towards a method that will be widely applicable, and robust with respect to the problems of real-world application.

In concluding our investigation, while the field has made significant progress so far, we observe that most of the splicing localization algorithms available today are more suited to relatively controlled environments than to Web and social media content. On the other hand, our work shows that multiple cases of Web images already exist, where existing algorithms can offer significant assistance to an investigator. Furthermore, previous work on classifier fusion has shown that it can further boost the overall reliability of a system [24]. Overall, however, automatic evaluations have a severe limitation in that they only try to emulate what human investigators would deduce from the algorithm outputs. One approach which could provide enlightening results would be to directly evaluate the state-of-the-art using human-computer interaction studies on actual users. In our recent work [55] we have presented an open, web-based tool featuring a number of state-of-the-art algorithms, intended to provide a live testbed for human investigators to evaluate the usefulness of algorithms. We believe that such inter-disciplinary research may also be necessary before we can reach the point where image forensics can be used by professionals without special forensics training to verify Web sourced content. It is our hope that the framework and results we presented in this investigation will encourage further research to this end.

Acknowledgments This work was supported by the REVEAL project, partially funded by the European Commission (contract no. FP7-610928).

References

1. Amerini I, Becarelli R, Caldelli R, Del Mastio A (2014) Splicing forgeries localization through the use of first digit features. IEEE International Workshop on Information Forensics and Security (WIFS) 143–148.
2. Ardizzone E, Bruno A, Mazzola G (2015) Copy-move forgery detection by matching triangles of keypoints. IEEE Transactions on Information Forensics and Security 10:2084–2094
3. Bianchi T, Piva A (2012a) Detection of nonaligned double JPEG compression based on integer periodicity maps. IEEE Transactions on Information Forensics and Security 7:842–848
4. Bianchi T, Piva A (2012b) Image forgery localization via block-grained analysis of JPEG artifacts. IEEE Transactions on Information Forensics and Security 7:1003–1017
5. Bianchi T, De Rosa A, Piva A (2011) Improved DCT coefficient analysis for forgery localization in JPEG images. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2444–2447
6. Birajdar GK, Mankar VH (2013) Digital image forgery detection using passive techniques: a survey. Digit Investig 10:226–245
7. "CASIA TIDEv2.0" (2009) http://forensics.idealtest.org/casiav2/ (Accessed 20–03-2016)
8. Chang IC, Yu JC, Chang CC (2013) A forgery detection algorithm for exemplar-based inpainting images using multi-region relation. Image Vis Comput 31:57–71
9. Chen M, Fridrich J, Goljan M, Lukás J (2008) Determining image origin and integrity using sensor noise. IEEE Transactions on Information Forensics and Security 3:74–90
10. Chierchia G, Poggi G, Sansone C, Verdoliva L (2014) A Bayesian-MRF approach for PRNU-based image forgery detection. IEEE Trans Inf Forensics Secur 9:554–567
11. Christlein V, Riess C, Jordan J, Riess C, Angelopoulou E (2012) An evaluation of popular copy-move forgery detection. IEEE Transactions on Information Forensics and Security 7:1841–1854
12. "Columbia image splicing detection evaluation dataset". (2004) http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm (Accessed 20–03-2016)
13. Cozzolino D, Gragnaniello D, Verdoliva L (2014a) Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques. IEEE International Conference on Image Processing (ICIP) 5302–5306
14. Cozzolino D, Gragnaniello D, Verdoliva L (2014b) Image forgery detection through residual-based local descriptors and block-matching. IEEE International Conference on Image Processing (ICIP), pp. 5297–5301
15. Cozzolino D, Poggi G, Verdoliva L (2015a) Efficient dense-field copy-move forgery detection. IEEE Trans Inf Forensics Secur 11:2284–2297
16. Cozzolino D, Poggi G, Verdoliva L (2015b) Splicebuster: A new blind image splicing detector. IEEE International Workshop on Information Forensics and Security (WIFS) 1–6.
17. de Carvalho T, Riess C, Angelopoulou E, Pedrini H, de Rezende Rocha A (2013) Exposing digital image forgeries by illumination color classification. IEEE Transactions on Information Forensics and Security 8:1182–1194
18. de O. Costa F, Oikawa MA, Dias Z, Goldenstein S, de Rocha AR (2014) Image phylogeny forests reconstruction. IEEE Transactions on Information Forensics and Security 10:1533–1546
19. de Oliveira AA, Ferrara P, de Rosa A, Piva A, Barni M, Goldenstein S, Dias Z, Rocha A (2016) Multiple parenting phylogeny relationships in digital images. IEEE Transactions on Information Forensics and Security 2:328–343
20. Diane N, Nanda W, Xingming S, Moise FK (2014) A survey of partition-based techniques for copy-move forgery detection. Sci World J 1:1–13
21. Emir D, Memon N (2009) Image tamper detection based on demosaicing artifacts. International Conference on Image Processing 1497–1500
22. Farid H (2009) Exposing digital forgeries from JPEG ghosts. IEEE Transactions on Information Forensics and Security 4:154–160
23. Ferrara P, Bianchi T, De Rosa A, Piva A (2012) Image forgery localization via fine-grained analysis of CFA artifacts. IEEE Transactions on Information Forensics and Security 7:1566–1577
24. Ferrara P, Fontani M, Bianchi T, de Rosa A, Piva A, Barni M (2015) Unsupervised fusion for forgery localization exploiting background information. IEEE International Conference on Multimedia & Expo Workshops 1–6.
25. Fontani M, Bianchi T, de Rosa A, Piva A, Barni M (2013) A framework for decision fusion in image forensics based on Dempster–Shafer theory of evidence. IEEE Transactions on Information Forensics and Security 8:593–607

26. Gaborini L, Bestagini P, Milani S, Tagliasacchi M, Tubaro S (2015) Multi-clue image tampering localization. IEEE National Conference on Parallel Computing Technologies (PARCOMPTECH):125–130
27. He Z, Wei L, Sun W, Huang J (2012) Digital image splicing detection based on Markov features in DCT and DWT domain. Pattern Recogn 45:4292–4299
28. Hsu YF, Chang SF (2006) Detecting image splicing using geometry invariants and camera characteristics consistency. IEEE International Conference on Multimedia and Expo 549–552.
29. Irene A, Becarelli R, Caldelli R, Casini M (2015) A feature-based forensic procedure for splicing forgeries detection. Math Probl Eng. doi:10.1155/2015/653164
30. Kakar P, Sudha N, Ser W (2011) Exposing digital image forgeries by detecting discrepancies in motion blur. IEEE Transactions on Multimedia 13:443–452
31. Kee E, O'Brien J, Farid H (2014) Exposing photo manipulation from shading and shadows. ACM Trans Graph 5:165:1–165:21
32. Kennedy L, Chang S-F (2008) Internet image archaeology: automatically tracing the manipulation history of photographs on the web. Proceedings of the 16th ACM international conference on Multimedia 349–358
33. Krawetz N (2007) A picture's worth: Digital image analysis and forensics, Online article on: http://www.hackerfactor.com/papers/bh-usa-07-krawetz-wp.pdf (Accessed 20–03-2016).
34. Li CT, Li Y (2012) Color-decoupled photo response non-uniformity for digital image forensics. IEEE Trans Circuits Syst Video Technol 22:260–271
35. Li W, Yuan Y, Yu N (2009) Passive detection of doctored JPEG image via block artifact grid extraction. Signal Process 89:1821–1829
36. Li J, L X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. IEEE Transactions on Information Forensics and Security 3:507–518
37. Lin Z, He J, Tang X, Tang CK (2009) Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. Pattern Recogn 42:2492–2501
38. Liu Q, Cao X, Deng C, Guo X (2011) Identifying image composites through shadow matte consistency. IEEE Transactions on Information Forensics and Security 6:1111–1122
39. Luo W, Qu Z, Huang J, Qiu G (2007) A novel method for detecting cropped and recompressed image block. International conference on Accoustics speech and. Signal Process 2:117–220
40. Lyu S, Pan X, Zhang X (2014) Exposing region splicing forgeries with blind local noise estimation. Int J Comput Vis 110:202–221
41. Mahdian B, Saic S (2009) Using noise inconsistencies for blind image forensics. Image Vis Comput 27: 1497–1503
42. Muhammad G, Al-Hammadi MH, Hussain M, Bebis G (2014) Image forgery detection using steerable pyramid transform and local binary pattern. Mach Vis Appl 25:985–995
43. Pasquini C, Brunetta C, Vinci A, Conotter V, Boato G (2015) Towards the verification of image integrity in online news. IEEE International Conference on Multimedia & Expo Workshops (ICMEW) 1–6.
44. Patel H, Patel M (2015) An improvement of forgery video detection technique using Error Level Analysis. Int J Comput Appl. doi:10.5120/19615-1508
45. Qureshi MA and Deriche M (2014) A review on copy move image forgery detection techniques 11th International Multi-Conference on Systems, Signals & Devices (SSD) 1–5.
46. Redi J, Taktak W, Dugelay JL (2011) Digital image forensics: a booklet for beginners. Multimedia Tools and Applications 51:133–162
47. "Report on the IEEE-IFS challenge on image forensics", (2013) http://www.signalprocessingsociety.org/news/581/607/Report-on-the-IEEE-IFS-challenge-on-image-forensics/ (Accessed 20–03-2016).
48. Ryu SJ, Kirchner M, Lee MJ, Lee HK (2013) Rotation invariant localization of duplicated image regions based on zernike moments. IEEE Transactions on Information Forensics and Security 8:1355–1370
49. Stamm CM, Wu M, Ray Liu KJ (2013) Information forensics: an overview of the first decade. IEEE Access 1:167–200
50. Wang W, Jing D, Tieniu T (2011) Tampered region localization of digital color images based on JPEG compression noise. Digital Watermarking. Springer Berlin Heidelberg 120–133.
51. Xu G, Ye J, Shi YQ (2014) New developments in image tampering detection. Springer International Publishing, Digital Forensics and Watermarking, pp. 3–17
52. Ye S, Sun Q, Chang EC (2007) Detecting digital image forgeries by measuring inconsistencies of blocking artifact. IEEE International Conference on Multimedia and Expo 12–15.
53. Yerushalmy I, Hel-Or H (2011) Digital image forgery detection based on lens and sensor aberration. Int J Comput Vis 92:71–91

54. Zampoglou M, Papadopoulos S, Kompatsiaris Y (2015) Detecting image splicing in the wild (web). IEEE International Conference on Multimedia & Expo Workshops (ICMEW) 1–6.
55. Zampoglou M, Papadopoulos S, Kompatsiaris Y, Bouwmeester R, Spangenberg J (2016) Web and social media image forensics for news professionals. Social Media in the Newsroom (#SMNews@ICWSM), Tenth International AAAI Conference on Web and Social Media Workshops
56. Zhao X, Li J, Li S, Wang S (2011) Detecting digital image splicing in chroma spaces. Digital Watermarking, Springer Berlin Heidelberg, pp. 12–22

Markos Zampoglou received a Degree in Applied Informatics in the University of Macedonia, Thessaloniki, Greece, in 2004, and an MSc in Artificial Intelligence from the University of Edinburgh in 2005. In 2011 he received a Ph.D. degree from the Department of Applied Informatics, University of Macedonia, on the subject of semantic video retrieval. He is currently a Postdoctoral Research Associate with the Information Technologies Institute (ITI), part of the Centre for Research and Technology Hellas (CERTH). His research interests include multimedia forensics and semantic multimedia analysis and retrieval

Symeon Papadopoulos received the Diploma degree in Electrical and Computer Engineering in the Aristotle University of Thessaloniki (AUTH), Greece in 2004. In 2006, he received the Professional Doctorate in Engineering (P.D.Eng.) from the Technical University of Eindhoven, the Netherlands. Since September 2006, he has been working as a research associate with the Information Technologies Institute (ITI), part of the Centre for Research and Technology Hellas (CERTH), on a wide range of research areas such as information search and retrieval, social network analysis, data mining and web multimedia knowledge discovery. In 2009, he completed a distance-learning MBA degree in the Blekinge Institute of Technology, Sweden. In 2012, he defended his Ph.D. thesis in the Informatics department of AUTH on the topic of large-scale knowledge discovery from social multimedia. He is currently Chair of the IEEE Special Technical Community on Social Networking (STCSN)

Ioannis (Yiannis) Kompatsiaris is a Senior Researcher (Researcher A') with the Information Technologies Institute / Centre for Research and Technology Hellas, Thessaloniki, Greece. His research interests include semantic multimedia analysis, indexing and retrieval, social media and big data analysis, knowledge structures, reasoning and personalization for multimedia applications, eHealth, security and environmental applications. He received his Ph.D. degree in 3-D model based image sequence coding from the Aristotle University of Thessaloniki in 2001. He is the co-author of 90 papers in refereed journals, 38 book chapters, 8 patents and more than 320 papers in international conferences. He has been the coorganizer of various international conferences and workshops and has served as a regular reviewer for a number of journals and conferences. He is a Senior Member of IEEE and member of ACM