DETECTING IMAGE SPLICING IN THE WILD (WEB)

Markos Zampoglou, Symeon Papadopoulos, Yiannis Kompatsiaris

Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece

ABSTRACT

As grassroots and social media-based journalism becomes more widespread, the need to verify information coming from such channels becomes imperative. In the past, there have been multiple occasions where forged pictures successfully passed as original news items, spreading misinformation or even panic. In this work, we investigate the potential for applying today's state of the art in image splicing detection in the context of images on the Web and images disseminated through social media. We investigate the alterations social media platforms apply on images and evaluate their impact on tampering detection. We further present a real-world dataset of forged images collected from various Web sources, and attempt to evaluate them using the current state-of-the-art in splicing detection. We present our results, and discuss their implications in real-world verification settings.

1. INTRODUCTION

There is inherent power in images, when used to accompany narratives. News reporters and services have been well aware of this fact, and have been using photographs to support their news reports since the birth of photography. Nowadays, following the spread of social media and micro-blogging, and the introduction of grassroots journalism, a similar trend can be observed on a massive scale, with laypeople taking the role of photo reporters and contributing their own news reports. In the resulting streams of information, false items, often in the form of tampered or fabricated images, are nowadays a common phenomenon and often lead to misinformation and confusion. Thus, from the standpoint of a news agency or authority following an event unfolding in social media news streams, there is a profound need for uncovering fake content.

Numerous algorithms have been proposed for tampering detection in digital images, with applications ranging from news items confirmation to courtroom evaluation of digital evidence. However, such algorithms tend to be sensitive to further alterations in the tampered image, e.g., recompression, scaling or histogram transformations. Indeed, a common assumption is that the suspect image at hand is the direct output of the (alleged) tampering process, and not a consecutive re-save of the image. While this is a reasonable assumption for a courtroom, where even the original raw camera images may be formally requested, it is hardly applicable in the context of the Web, where it is often difficult to trace an image to its original source, and users and services often modify an image before re-posting it. In this context, the breakdown of our fundamental assumptions drastically changes the performance of most detection algorithms (Figure 1).

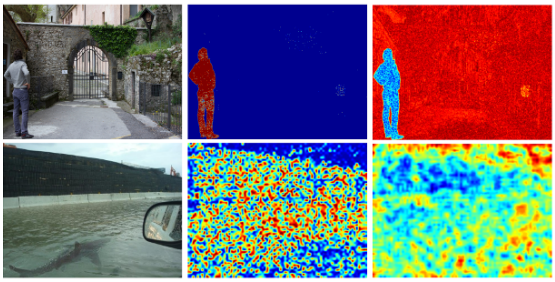

Fig. 1. Top: analysis of a forgery from an experimental dataset (VIPP). Bottom: an internet version of a well-known forgery. The algorithms used for the surfaces are [1] and [2].

This paper attempts to assess the real-world performance of state-of-the-art splicing detection techniques on news-related photographs appearing on the Web. To this end, we explore the types of modifications that major social media services apply to images posted on them, and proceed to evaluate the impact of such modifications on tampering detection. We further present a real-world dataset of known tampered images that spread on the Web in recent years, to serve as a realistic benchmark for the image forensics research community. We also propose a human-centered evaluation methodology for tampering localization algorithms, and use it to evaluate a number of state-of-the-art algorithms on our data. We close with our conclusions concerning the state of the art in real-world image tampering detection.

2. RELATED WORK

2.1. Dominant approaches

We can divide tampering detection methods into three broad categories, by the type of tampering they each detect: splicing, copy-move attack, and resampling/histogram operations. Image splicing refers to the practice of copying a part of one image and inserting it into another, so as to give the impression that an additional element was present in a scene.

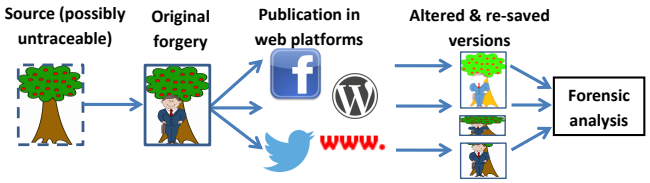

Fig. 2. Depiction of the stages between the original forgery and the forensic analysis in a real-world scenario.

Copy-moving is the practice of taking a part of an image and copying it within the same image. This can both be used to falsely add more information (e.g., make a crowd seem larger) or remove it (e.g., copy-moving the background over items). Finally, the third group includes alterations which generally tend to be well-intended and are rarely used to change an image's semantic content. With respect to detection, copy-move forgery detection algorithms are generally based on common image content search algorithms, by seeking internal replications of patches or keypoints. Common problems faced are achieving robustness with respect to transformations of the replicated patch, and tackling the high computational demands of exhaustive within-image search. A recent survey of the state-of-the-art can be found in [3]. Splicing detection, which is the focus of this work, relies on an entirely different premise: the assumption -and prerequisite- is that the spliced region essentially carries information that, whilst possibly invisible to the eye, is in some significant aspect different than in the rest of the image. There exist multiple cues that can be used to make this distinction. One trace left by the image capturing process are Color Filter Array (CFA) interpolation patterns, which tend to be camera model-related. Splicing can destroy CFA interpolation statistics, allowing for accurate splice localization [4]. A second camera-related characteristic of digital images is sensor noise, as different camera models and devices tend to introduce differently distributed noise in images. Inconsistencies in the distribution of local noise can thus be used to detect splicing [5]. Another large part of the bibliography attempts to leverage the particularities of JPEG compression to detect image alterations. Approaches look for disruptions of the JPEG block patterns [6] or take advantage of periodic patterns created in DCT coefficient histograms during re-quantization [1, 7]. Another approach is to detect DCT artifacts caused by non-aligned double JPEG compression [2], or to seek *JPEG Ghosts*, i.e. traces of past JPEG compressions [8]. Finally, certain methods seek discontinuities in the spatial features of the spliced image, using Gray-Level Run Length features [9] or Local Binary Patterns over the Steerable Pyramid transformed image [10].

2.2. Experimental datasets

Currently, there exist a number of benchmark datasets for evaluating tampering detection algorithms. Of these, the dataset of [3] and the CoMoFoD dataset [11] concern Copy-Move detection, while the Columbia Image Splicing Detection Evaluation Dataset[1], the Columbia Uncompressed Image Splicing Detection Evaluation Dataset [12], the CASIA Tampered Image Detection Evaluation Database[2] and the Visual Information Processing and Protection Group dataset [13] concern image splicing. Table 1 helps highlight certain limitations of the aforementioned datasets with respect to the task of Web image verification. One is that the image format in most datasets is different from JPEG, which is the prevalent format on the Web: for example, among the contents of the Common Crawl corpus[3], 87% of identifiable image suffixes correspond to JPEG (.jpg, .jpeg). With respect to our task, within the Wild Web dataset we collected for our experiments using an exhaustive search, described in Paragraph 3.2.2, 95% of a total of 13,577 forged images were in JPEG format. In contrast, only the VIPP datasets and a small part of CASIA v2.0 contain JPEG images. This leaves out a significant part of the bibliography that takes advantage of the effects of JPEG compression. Secondly, the absence of ground-truth masks in some datasets makes it very hard to evaluate algorithms producing localized results, which can be a crucial requirement for human investigators. Finally, a common characteristic of all datasets is their "neatness", meaning that the images contained in them have undergone at most two lossy compressions, one as original images, and one following the splice.

Table 1. Image splicing benchmark datasets. *Format*: the format of the images contained. *Masks*: the presence or absence of ground-truth binary masks giving the location of the splice (*Manual* means the masks were manually constructed by us using the original and spliced images). *# images*: the number of authentic and spliced images in the dataset.

Name	Format	Masks	# images
Columbia	BMP grayscale	No	933/912
Columbia Uncomp.	TIFF Uncomp.	Yes	183/180
CASIA TIDE v2.0	TIFF Uncomp., JPEG, BMP	No	7491/5123
VIPP Synth.	JPEG	Yes	4800/4800
VIPP Real.	JPEG	Manual	68/69

3. MULTIMEDIA VERIFICATION IN THE WILD

The tampering detection methods described above cover a wide variety of phenomena. However, as mentioned above, they often share one assumption: that the image under examination has not undergone further alterations following the

[1] http://www.ee.columbia.edu/ln/dvmm/downloads/AuthSplicedDataSet/AuthSplicedDataSet.htm

[2] http://forensics.idealtest.org

[3] http://commoncrawl.org/

Fig. 3. Examples of different forgeries. Left: original forgery. Center: post-splice. Right: cropping.

Fig. 4. Three versions of the same forgery.

tampering operation. Indeed, it is often assumed that the forensic algorithm has to decide between the image being either a camera original or a tampered and re-saved image. In practice, any trace of splicing can be erased given a sufficient number of alterations such as lossy (e.g., JPEG) recompressions, resampling (e.g., scaling) or filtering. However, images on the Web are likely to undergo one or more such operations before the forensic investigator has a chance to examine them (Figure 2). Thus, we are in need of an evaluation framework that reflects the realities of Web images.

3.1. Reverse-engineering the impact of social media platforms on uploaded images

Twitter is a major content source for journalists. Its real-time nature and convenient access to its content make it highly popular for disseminating on-the-spot information and images of developing news stories. The issue of fake news images posted on Twitter has been highlighted in the past [14, 15] and remains open. For the purposes of this work, it is important to investigate the ways Twitter alters uploaded images, and their impact on forensic analysis. To this end, we experimented extensively with PNG and JPEG images, since these are the most common image types, and found that Twitter follows these rules:

- If the image is larger than 2048×1024, rescale it to fit these dimensions whilst retaining the aspect ratio. The resampling function has been most closely simulated by a Lanczos-3 kernel.
- If it is in PNG format, and is $>$3MB, convert to JPEG.
- If it was originally in JPEG format, or was converted to JPEG, resave it at a quality factor of 75.

We also repeated a similar process for Facebook. Although Facebook is not the usual medium of choice for grassroots journalism, its high popularity means that there is a significant chance that an image arriving at our hands has passed through it. The corresponding rules for Facebook are highly similar to those of Twitter, with the main differences being that the image limit is 2048×2048, and the JPEG quality factor can vary between 70 and 90 depending on the image.

While the details may differ, the pattern is clear: social media images are most likely to have undergone one JPEG recompression, and quite possibly to have also undergone rescaling. Either of these operations could be enough to disrupt a forensic algorithm, rendering it practically useless for social media application. Furthermore, we also know that social media platforms tend to erase image metadata[4], which can be an invaluable tool for forensic analysis (especially Exif). Since many blogging and image hosting services also impose similar alterations on images, we ought to be prepared to deal with their overall impact.

3.2. Towards a real-world evaluation framework

3.2.1. Emulation of social media-like operations on benchmark datasets

We already know that the vast majority of algorithms proposed for forgery detection are, by their definition, not designed to work on images that have been re-compressed or, even worse, resampled due to scaling. We decided to experimentally verify this hypothesis by applying such operations on existing benchmark datasets and evaluating a number of state-of-the-art algorithms on them. Out of the existing datasets, we used Columbia Uncompressed, VIPP Synthetic and VIPP Realistic, since those offered binary masks for evaluations. On these datasets, we emulated two scenarios. The first was the re-compression of the images as JPEG of quality 75 to emulate the conversions applied by Twitter and Facebook, while the second included both resaving at that quality and re-scaling at 75% of the original image size. The degree of degradation of algorithm results on these datasets can give us an estimate of what to expect from a real-world application of state-of-the-art algorithms.

3.2.2. The Wild Web tampered image dataset

Besides our emulated datasets, we also created a collection of actual forged images from the Web, accompanied by manually constructed ground-truth binary masks, and the original, untampered sources, wherever those were available. This "real-world" dataset consists of 82 unique cases of confirmed forgeries. However, in gathering the dataset, it was often impossible to locate the original, first-posted forgery for each case. Instead, the investigator will often have access to one or more alternative versions of the same image, which will have already undergone various transformations. To emulate this

[4] `http://www.embeddedmetadata.org/social-media-test-procedure.php`

scenario, we downloaded all instances of each forgery that we could locate using Google reverse image search. The collected 13,577 unique images (with exact duplicates removed) featured a number of variants of each forgery, falling into one or more categories (Figure 3): a) versions of the same image at various scales, often at different aspect ratios, b) cropped versions of the original image, and c) post-splices: watermarks, frames, or further, generally obvious splices added over the original forged images. We decided to separate the obvious post-splices and croppings from the rest of the dataset, thus keeping 9,666 images that were most likely to resemble (or be) the forgery originally uploaded by the perpetrator. Especially for the croppings, the reason was not only the degrading they cause in detection performance, but also the difficulties in creating ground-truth binary masks for them. In the case of rescaling (even at different aspect ratios), the same binary masks could apply to all instances, following a proportionate rescaling of the mask. On the other hand, each cropping would require its own binary mask, of which the creation would be a very labour-intensive task. Exceptions to this practice were the cases where it was impossible to deduce which version of a forgery was closer to the original forgery (Figure 4). In these occasions, each version was kept in the dataset as a separate case with its own mask. Creating ground-truth masks was also a challenge, even when the originals were readily available. In many cases, multiple areas of the image contained different splices, possibly committed at different times. In those cases, we created binary masks for each case, and evaluated them separately. This brought our dataset to 101 unique masks for evaluation over 92 case variants of the original 82 cases.

3.2.3. *Evaluation protocol*

When evaluating tampering detection algorithms over a dataset, the criteria used for determining success are integral to the evaluation. While many algorithms return a binary result for the entire image [9, 10], thus simplifying evaluation at the cost of localization, many others return a surface, whose values correspond to local estimates, as in Figure 1. In the latter case, experimental evaluation requires the existence of a reference mask, signifying the actual location of the tampered region. Consecutively, a typical evaluation protocol is to contrast values of the surface under the mask with those outside it. For example, we can estimate the statistical median of the map values for the pixels/blocks under the mask with the median of those outside it [13], or evaluate the Kolmogorov-Smirnov statistic on the two value distributions [8]. While such protocols have served the research community well in the past, we consider them inappropriate for our use case for two reasons. One is their potential for poor localization, in cases where large areas outside the mask return high values, but not large enough for classification to be recognized as a failure (Figure 5). Second and most important is their unrealistic concept of a false positive: such protocols can declare a false positive *only* when they find significant differences in a predetermined area in a non-spliced image. From the perspective of a non- or semi-expert investigator, detection algorithms ought to return reliable, clearly interpretable results highlighting the forged areas of the image and only those. Any results whose area does not match the ground truth should be considered false, and so should non-spliced images returning *any* salient map area, anywhere on the image.

Fig. 5. Evaluation by comparing the medians inside and outside the mask (white circle). Due to the uniform background, the median outside the mask is very low, but from a real-world perspective this is a largely false output.

Table 2. Algorithm evaluation over our emulated dataset using a difference-of-medians criterion (top values) and our evaluation protocol (bottom values). In both approaches, the first value reflects the detection rate (True Positives) while the value in parentheses is the false alarm rate (False Negatives).

Dataset	[1]	[7]	[4]	[2]a	[2]b	[5]
Col. U.	-	-	0.89 (0.05)	-	-	0.39 (0.04)
Orig.	-	-	0.66 (0.16)	-	-	0.12 (0.57)
VIPP S.	0.47 (0.05)	0.51 (0.05)	0.15 (0.05)	0.57 (0.01)	0.28 (0.05)	0.13 (0.05)
Orig.	0.44 (0.27)	0.52 (0.00)	0.01 (0.23)	0.58 (0.09)	0.04 (0.25)	0.04 (0.74)
VIPP R.	0.54 (0.04)	0.58 (0.04)	0.04 (0.04)	0.70 (0.04)	0.28 (0.04)	0.20 (0.04)
Orig.	0.41 (0.46)	0.38 (0.09)	0.09 (0.22)	0.23 (0.30)	0.03 (0.39)	0.04 (0.90)
Col. U.	-	-	0.05 (0.05)	-	-	0.09 (0.05)
JPEG	-	-	0.00 (0.20)	-	-	0.02 (0.86)
VIPP S.	0.30 (0.04)	0.43 (0.04)	0.17 (0.05)	0.39 (0.05)	0.16 (0.05)	0.10 (0.05)
JPEG	0.26 (0.30)	0.30 (0.10)	0.01 (0.28)	0.23 (0.27)	0.01 (0.29)	0.04 (0.74)
VIPP R.	0.32 (0.04)	0.36 (0.04)	0.14 (0.04)	0.51 (0.04)	0.17 (0.04)	0.20 (0.04)
JPEG	0.13 (0.44)	0.17 (0.29)	0.00 (0.25)	0.14 (0.46)	0.01 (0.43)	0.02 (0.90)
Col. U.	-	-	0.03 (0.04)	-	-	0.11 (0.05)
resamp.	-	-	0.00 (0.24)	-	-	0.04 (0.79)
VIPP S.	0.05 (0.05)	0.05 (0.05)	0.05 (0.04)	0.05 (0.05)	0.05 (0.05)	0.06 (0.05)
resamp.	0.00 (0.23)	0.00 (0.00)	0.00 (0.23)	0.00 (0.15)	0.00 (0.29)	0.00 (0.84)
VIPP R.	0.13 (0.04)	0.12 (0.06)	0.03 (0.04)	0.23 (0.04)	0.17 (0.04)	0.23 (0.04)
resamp.	0.00 (0.47)	0.00 (0.00)	0.00 (0.28)	0.03 (0.25)	0.01 (0.47)	0.03 (0.47)

In our proposed evaluation approach, the output produced by each algorithm is first binarized using a method-specific threshold (i.e. for each algorithm, the possible thresholds reflect the meaning and range of the values returned), and then image morphological processing operations (such as opening and closing) are applied, to remove noise and retain connected regions. The resulting binary images are then compared to the ground truth mask, to evaluate the quality of the match. The criterion for comparing the mask produced by the algorithm to the ground truth binary mask is expressed by Equation 1.

$$E(A, M) = \frac{\Sigma(A \cap M)^2}{\Sigma(A) \times \Sigma(M)} \quad (1)$$

where A signifies the binary, processed algorithm output, M is the ground truth mask, and $\Sigma(x)$ is the area of a binary mask x. Experimentally, any value of $E(A, M)$ above 0.65 suggests a very good match which is very unlikely to have resulted by chance. The detection of false positives, on the other hand, follows a different approach. In contrast to other protocols, we are not expected to have a binary mask for each non-spliced image. Instead, following a similar thresholding and morphological processing step, we expect the algorithm output not to demonstrate *any* salient regions. The presence of such a region is classified as a false positive. While this means that, essentially, a different criterion is used for true positive and false positive detection, nonetheless we hold this protocol to more closely reflect our real-world task requirements.

4. EXPERIMENTAL STUDY

4.1. Tested approaches

We compared a number of well-established, state-of-the-art algorithms for image forensic analysis. The code for [7, 4, 2] was provided by the authors, while we further implemented [1, 5, 8] ourselves. As described in section 2.1, these algorithms cover a variety of phenomena, allowing us to tackle the tampering detection problem from multiple sides.

4.2. Results

As a preliminary step, we ran the above algorithms on the emulated data set we described in paragraph 3.2.1. For comparison, we present results both using a typical evaluation protocol (comparing the median feature values inside and outside the mask, Table 2, top values) and our thresholding protocol (bottom values). For the typical methodology of comparing the medians inside and outside the mask, the binary mask used for the non-spliced images was a parallelogram in the centre of the image, at 1/16 of the image area, similar to [13]. For the difference between medians, we chose thresholds that gave FN rates of < 0.05. For our own protocol, we applied multiple fixed and adaptive thresholds, and a battery of different morphological operations. For both positive and negative examples, we kept the best (i.e., closer to ground truth) result, to reflect the ability of a human inspector to detect the presence or absence of distinctive patterns. We then evaluated the forensic algorithms on the Wild Web dataset. The results are presented on Table 3. We present the number of cases where an algorithm achieved correct detection for at least one image of each case. The number in parentheses ("Unique") indicates the number of cases where a particular algorithm was the only one to achieve detection for that case.

Table 3. Algorithm evaluation over the Wild Web dataset.

	[1]	[7]	[4]	[2]a	[2]b	[5]	[8]
Detections	13	12	1	8	5	15	29
(Unique)	(4)	(1)	(0)	(1)	(2)	(6)	(10)

In this evaluation, we included one additional algorithm. [8] is more difficult to evaluate with respect to false positives, as its output tends to be rather noisy and consists of multiple images, many of which are bound to contain images that are, to an extent, salient. In an automatic detection framework, this method would produce numerous false positives, as *some* area is always expected to stand out at *some* level. On the other hand, visual inspection (with the semantic awareness of a human investigator) tends to give very good results, so we opted to include it separately. However, its localization also tends to be relatively poor, so we reduced the area matching threshold to 0.45 - still high enough to discard random findings. For fairness, we did not take the findings from [8] into account when counting unique hits from the other methods, while unique findings from [8] are counted against all findings from all other methods. Overall, out of 82 cases, 57 (47 when including [8]) gave no detection, while 25 (35) had at least one method correctly detect and localize the forgery in at least one image. However, visual inspection suggests that even our approach may be overestimating the output, especially for [8] and [5], with real recognition capabilities being even below that level.

4.3. Discussion

With respect to our emulation of social media image alterations, the comparison between the two tables offers a number of interesting observations: First, it becomes clear how simply comparing the values inside and outside the mask can lead to a serious underestimation of false positive rates. Secondly, the specialization of different algorithms in different cases becomes obvious: [4] seems to work only on the Columbia dataset (which is to be expected, as CFA patterns are particularly sensitive to JPEG compression), while [2]b and [5] seem to actually yield very poor results in all cases. Finally, the degradation due to image recompression and especially resampling is clear: while some algorithms show a degree of robustness to one recompression, resampling diminishes any possibility of detection (this becomes clear especially using our evaluation protocol).

Concerning the new Wild Web dataset, the first important observation is the low overall rate of detection. Having performed an exhaustive search over all available instances of a forgery in the real world, the likelihood of achieving correct detection using any of today's most successful approaches was, overall, relatively slim. This can either be attributed to the fact that the original forgeries are most often lost, and

only resampled versions remain online for us to investigate, or to the possibility that forgers tend to upload undetectable versions of their forgeries in the first place, having applied rescaling, blurring or other methods that make localization impossible. A second observation is the relative success of simpler methods against more complex ones. [1], [2]b and [5] significantly contributed to detection, by accurately localizing a number of unique forgeries. Especially the success of [5], which is a noise-based method, might suggest that, in real-world situations, differences in image noise might provide a valuable clue for forgery detection. Similarly, [8], without overlooking its tendency to generate noisy outputs, demonstrated high success rates, which we confirmed by visual inspection. It is possible that such approaches are in the long run more robust for Web image verification compared to more complex approaches, which may be more appropriate for controlled verification settings (such as a courtroom).

5. CONCLUSIONS

We presented an investigation of the performance of state-of-the-art image tampering detection methods on realistic Web image verification problems. By reverse-engineering and reproducing the image transformations applied by two major social media platforms, we confirmed the theoretical intuition that such operations can completely diminish the possibility of successful detection of image forgeries. We further proceeded to organize and present a real-world dataset of well-established forgeries, which we tried to detect using a set of state-of-the-art approaches. While a number of forgeries were detected, the algorithms we applied failed in the majority of cases. Overall, our investigation exposed a performance gap that today's approaches have to cover before we can claim to achieve successful forgery detection in the context of social media and the Web. As an immediate next step towards bridging this gap, we intend to make our Wild Web dataset publicly available, in order to foster real-world oriented research in image forensics.

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

6. REFERENCES

[1] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, 2009.

[2] Tiziano Bianchi and Alessandro Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.

[3] Vincent Christlein, Christian Riess, Johannes Jordan, Corinna Riess, and Elli Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[4] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, 2012.

[5] Babak Mahdian and Stanislav Saic, "Using noise inconsistencies for blind image forensics," *Image and Vision Computing*, vol. 27, no. 10, pp. 1497–1503, 2009.

[6] Yi-Lei Chen and Chiou-Ting Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 396–406, 2011.

[7] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *ICASSP*. 2011, pp. 2444–2447, IEEE.

[8] Hany Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 154–160, 2009.

[9] Xudong Zhao, Jianhua Li, Shenghong Li, and Shilin Wang, "Detecting digital image splicing in chroma spaces," in *IWDW*. 2010, vol. 6526 of *Lecture Notes in Computer Science*, pp. 12–22, Springer.

[10] Ghulam Muhammad, Muneer H. Al-Hammadi, Muhammad Hussain, and George Bebis, "Image forgery detection using steerable pyramid transform and local binary pattern," *Machine Vision and Applications*, vol. 25, no. 4, pp. 985–995, 2014.

[11] Dijana Tralic, Ivan Zupancic, Sonja Grgic, and Mislay Grgic, "Comofod - new database for copy-move forgery detection," in *ELMAR, 55th International Symposium*, 2013, pp. 49–54.

[12] Yu-Feng Hsu and Shih-Fu Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *ICME*. 2006, pp. 549–552, IEEE Computer Society.

[13] Marco Fontani, Tiziano Bianchi, Alessia De Rosa, Alessandro Piva, and Mauro Barni, "A framework for decision fusion in image forensics based on dempster-shafer theory of evidence," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 593–607, 2013.

[14] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy," in *22nd International World Wide Web Conference, WWW '13*. 2013, pp. 729–736, ACM.

[15] Christina Boididou, Symeon Papadopoulos, Yiannis Kompatsiaris, Steve Schifferes, and Nic Newman, "Challenges of computational verification in social multimedia," in *23rd International World Wide Web Conference, WWW '14*. 2014, pp. 743–748, ACM.