Rebroadcast Attacks: Defenses, Reattacks, and Redefenses

Wei Fan, Shruti Agarwal, and Hany Farid
Computer Science
Dartmouth College
Hanover, NH 03755
Email: {wei.fan, shruti.agarwal.gr, hany.farid}@dartmouth.edu

***Abstract*—A rebroadcast attack, in which an image is manipulated and then re-imaged, is a simple attack against forensic techniques designed to distinguish original from edited images. Various techniques have been developed to detect rebroadcast attacks. These forensic analyses, however, face new threats from sophisticated machine learning techniques that are designed to modify images to circumvent detection. We describe a framework to analyze the resilience of rebroadcast detection to adversarial attacks. We describe the impact of repeated attacks and defenses on the efficacy of detecting rebroadcast content. This basic framework may be applicable to understanding the resilience of a variety of forensic techniques.**

I. Introduction

A number of forensic techniques have been developed to detect various types of image manipulations [1]. Among these techniques, there exists what is often referred to as file-based techniques that are specifically designed to detect any modification of an original JPEG file, but not necessarily the nature of the manipulation. These include: (1) analyzing JPEG compression parameters, JPEG file markers, and EXIF format and content to determine if the overall JPEG packaging is consistent with the expected properties of the source camera [2], [3], [4]; (2) analyzing the embedded thumbnail image to determine if its construction and format are consistent with the source camera or that of a photo-editing software [7]; and (3) analyzing the encoded discrete cosine transform coefficients for evidence of multiple compressions that would arise, for example, after modifying and saving an image in a photo-editing software [9], [10]. Recent advances in machine learning have also been used to automatically detect changes to an original JPEG file [5], [6], [8], [11], [12].

Despite their efficacy, these techniques suffer from a simple rebroadcast attack in which an altered image is re-imaged, thus ensuring that all underlying camera properties will appear as original. We describe a technique for detecting this type of attack and its resilience to further adversarial attacks.

II. Rebroadcast Attack and Defense

There are two simple types of rebroadcast attacks generated by photographing a high-quality printed copy of an image, or photographing a displayed image on a high-resolution monitor. These approaches are relatively easy to execute and will result in an image file that is consistent with a camera original. Two other types of rebroadcast attacks are generated by scanning with a high-resolution flatbed scanner a printed copy of an image or capturing a screen-grab of a displayed image on a monitor. Unlike the first two approaches, these approaches will require some further manipulation to add the necessary JPEG file details to be consistent with a camera original.

Many techniques have been developed to detect rebroadcast attacks. These include the use of higher-order wavelet statistics to identify scanned images [13], local binary patterns to identify displayed images [14], Markov-based features to identify printed images [15], physics-based features to identify printed images [16], noise statistics and double JPEG compression to identify displayed images [17], aliasing patterns to identify displayed images [18], image-edge profiles to identify displayed images [19], and a convolutional neural network to identify displayed images [20]. A few other techniques attempt to simultaneously detect rephotographed printed and displayed images [21], [22], [23].

The simultaneous detection of all four types of rebroadcast attacks was first described in [24]. We will briefly summarize these results. The authors in [24] collect a dataset of $14,500$ original images from $1,294$ distinct cameras and $14,500$ rebroadcast images from a diverse set of distinct recapture devices: 234 displays, 173 scanners, 282 printers, and 180 recapture cameras. The performance of four different classification techniques is evaluated against this dataset: Three of the techniques are based on hand-crafted features [13], [14], [15] coupled with a non-linear support vector machine (SVM), and the fourth technique is based on a convolutional neural network (CNN). The CNN, described below, significantly outperforms the other approaches, so we will focus only on this classifier.

As proposed in [24], we train a CNN to classify small image blocks as original or rebroadcast, where a rebroadcast image block can be any of the four classes described above. The input to the network is a monochromatic (red channel) 64×64 image block I, and the output is a two-dimensional vector given by the function $\phi(I) \in \mathbb{R}^2$. The network consists of seven convolutional layers and two fully connected layers followed by a log-softmax layer. The first convolutional layer consists of 16 predefined Gaussian filters residuals with two different filter sizes and 8 different standard deviations. The detailed description of all hyper-parameters can be found in [24].

The set of $14,500$ original and $14,500$ rebroadcast images is randomly divided into 60:20:20 training, validation and testing sets. These images are partitioned into 4.35, 1.44, and 1.45 million training, validation, and testing image blocks. The overall training, validation, and testing accuracies are 98.85%, 98.46%, and 98.61%, with almost no difference in the detection of original or rebroadcast.

Here we analyze the vulnerability and resilience of this CNN-based approach towards multiple and repeated counter-forensic attacks that are designed to modify images to circumvent detection. A similar type of attack was proposed in [30] in which the authors described a gradient-based attack against SVM classifiers. Expanding on this basic idea, we explore the impact of repeated attacks and defenses on CNN classifiers.

III. A Second Attack and Defense

In the previous section, we see that a CNN can be trained to effectively distinguish original from rebroadcast images. In this section, we evaluate the resilience of this network to a counter-forensic attack.

Given an input image block I (we will refer to this block simply as an image), the output of our CNN is a two-dimensional vector $\phi(I)$. The input is classified as original or rebroadcast based on the sign of the following function:

$$f(I) \quad = \quad \vec{v}^T \phi(I), \tag{1}$$

where $\vec{v}^T = \begin{pmatrix} -1 & 1 \end{pmatrix}$. The function $f(\cdot)$ computes the difference between the two outputs and classifies an image as original if this difference is less than zero, $f(I) < 0$, and rebroadcast otherwise.

The goal of attacking this CNN is to modify a rebroadcast image I (with $f(I) \geq 0$) such that it will be classified as original ($f(I) < 0$). This attack can be formalized as an optimization of the following form:

$$\hat{I} \quad = \quad \arg\min_{I} f(I). \tag{2}$$

We solve this optimization problem using the gradient descent method with momentum which iteratively updates the solution according to the following update rule at the k^{th} iteration ($k = 0, 1, 2, \cdots$):

$$I^{k+1} \quad = \quad I^k - \alpha \left(m f'(I^{k-1}) + f'(I^k) \right), \tag{3}$$

where m is the momentum, α is the learning rate, and $f'(\cdot)$ is the gradient of Equation (1). Our CNN is implemented using the PyTorch framework [31]. PyTorch's *autograd mechanics* provides a reverse automatic differentiation system which yields the desired gradient $f'(\cdot)$. The gradient descent optimization is initialized with $I^0 = I$, $f'(I^{-1}) = 0$, and momentum $m = 0.9$. The learning rate is initialized to $\alpha = 1e^{-4}$ and is decreased by a factor of 0.9 when the loss plateaus. When the learning rate is reduced, the momentum is set to $m = 0$ for that iteration and reset to $m = 0.9$ in subsequent iterations.

The gradient descent iteration terminates under any of the following conditions: (1) the modified rebroadcast image is classified as original: $f(I^k) < 0$; (2) the learning rate α is less than a predefined threshold of $1e^{-8}$; or (3) the number of iterations k exceeds a predefined threshold of $1,000$.

A successful attack is one in which the modified rebroadcast images are mis-classified as original and the average difference between the rebroadcast and modified rebroadcast images is minimal (we measure image difference using mean-squared error, MSE). We do not explicitly penalize large deviations of MSE to give the attacker as much flexibility as possible. We have found, however, that a small learning rate typically (but not always) yields a modified rebroadcast image that is similar to the input rebroadcast image.

Starting with 0.63 million rebroadcast images, we generate a corresponding set of 0.63 million attack-rebroadcast images. The true positive rate (correctly classifying rebroadcast images) from the previous section is 98.54%. This rate plunges to 0.005% on the attack-rebroadcast images. At the same time, the average MSE between the rebroadcast and attack-rebroadcast images is only 0.96 (all images are integer-valued and span an intensity scale of $[0, 255]$).

IV. Iterative Attacks and Defenses

We have seen that a CNN is highly effective at detecting a broad range of rebroadcast attacks. We have also seen that this same CNN is vulnerable to a fairly simple counter-forensic attack in which a rebroadcast image can be slightly modified to evade detection. In this section we ask if a newly trained CNN can detect this new attack, and if repeated attacks against this defense are successful or not.

A. Single attack

The set of original and rebroadcast images described in Section II is denoted as $\mathcal{O}$ and $\mathcal{R}$. The CNN trained to discriminate between these images is denoted as $\mathcal{D}_1$. In the previous section, we describe how $\mathcal{D}_1$ can be attacked. In this section we explore whether this type of attack can be defended against repeated cycles of detect (D) and attack (A):

(D1) The first full detect/attack cycle starts with a defense against a rebroadcast attack. In particular, a CNN $\mathcal{D}_1$ is trained to distinguish between original $\mathcal{O}$ and rebroadcast $\mathcal{R}$ images as described in Section II.

(A1) The first cycle ends with an attack against $\mathcal{D}_1$ in which attack-rebroadcast images $\mathcal{R}_1$ are generated from $\mathcal{R}$ by attacking $\mathcal{D}_1$ using the gradient descent method described in Section III.

(D2) In the second defense, a new CNN $\mathcal{D}_2$ is trained on $\{\mathcal{O}, \mathcal{R}, \mathcal{R}_1\}$, where, all of the rebroadcast and attack-rebroadcast images are bundled together into a single class.

(A2) This cycle ends with an attack against $\mathcal{D}_2$ in which attack-rebroadcast images $\mathcal{R}_2$ are generated from $\mathcal{R}$ by attacking $\mathcal{D}_2$.

(Di) In the i^{th} defense, a CNN $\mathcal{D}_i$ is trained on $\{\mathcal{O}, \mathcal{R}, \mathcal{R}_1, \cdots, \mathcal{R}_{i-1}\}$.

(Ai) This cycle ends with an attack against $\mathcal{D}_i$ in which attack-rebroadcast images $\mathcal{R}_i$ are generated from $\mathcal{R}$ by attacking $\mathcal{D}_i$.

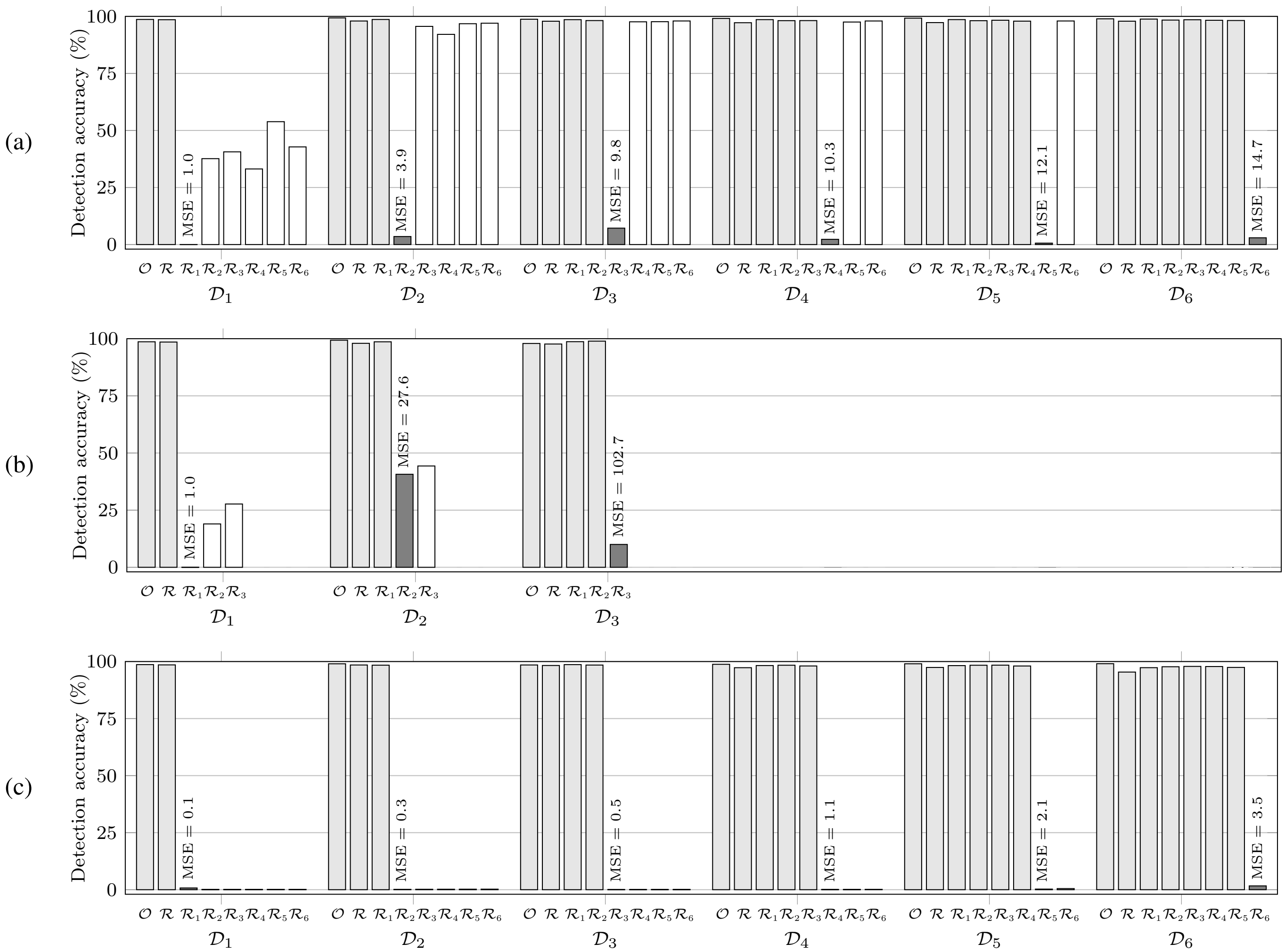

Fig. 1. Detection accuracy for detectors $\mathcal{D}_i$ against original $\mathcal{O}$, rebroadcast, $\mathcal{R}$, and attack-rebroadcast $\mathcal{R}_j$ images, corresponding to (a) a single attack; (b) multiple attacks; and (c) multiple (slow) attacks. The light gray bars ($i > j$) correspond to the accuracy against content for which the CNN detectors are trained. The dark gray ($i = j$) and white bars ($i < j$) correspond to the accuracy for attack-rebroadcast images against detectors in the current and previous detect/attack cycles. The dark gray bars are each annotated with an MSE value corresponding to the difference between the rebroadcast and attack-rebroadcast images (the integer-valued images are on an intensity scale of $[0, 255]$). All bars to the right of these dark gray bars are white.

In order to avoid a skewed training dataset, in the i^{th} cycle, the rebroadcast class is constructed from a randomly selected fraction of $1/i$ rebroadcast images $\mathcal{R}$ along with their corresponding attack-rebroadcast images in $\{\mathcal{R}_1, \cdots, \mathcal{R}_{i-1}\}$. This sampling ensures that the rebroadcast class size stays the same in each cycle.

We carry out six detect/attack cycles. The CNN training (detect) and the gradient descent (attack) are the same as described in the previous sections. Shown in Fig. 1(a) are the detection accuracies of these six CNNs on each subset of original, rebroadcast, and attack-rebroadcast images. Each detector $\mathcal{D}_i$ is trained on the images rendered as light gray bars. In each case, and as expected, detection accuracy remains high on these images (above 97%). We see, for example, that the CNN $\mathcal{D}_3$ can learn to discriminate original ($\mathcal{O}$) from rebroadcast ($\mathcal{R}$) as well as attack-rebroadcast ($\mathcal{R}_1$ and $\mathcal{R}_2$) images. This detector, however, is unable to defend against a new attack $\mathcal{R}_3$ as shown by the low detection accuracy rendered in dark gray. As shown in Fig. 1(a), this pattern continues for all detectors $\mathcal{D}_i$: CNN $\mathcal{D}_i$ detects $\{\mathcal{R}, \mathcal{R}_1, \cdots, \mathcal{R}_{i-1}\}$, but not the subsequent attack $\mathcal{R}_i$.

The value above each dark gray bar in Fig. 1(a) corresponds to the average MSE between the rebroadcast and attack-rebroadcast images. Although the classifier on repeated detect/attack cycles is not able to defend against new attacks, we do see that the attack does become more difficult as the MSE grows from 1.0 for $\mathcal{R}_1$ in $\mathcal{D}_1$ to 14.7 for $\mathcal{R}_6$ in $\mathcal{D}_6$. Despite the slight increase in MSE after repeated detect/attack cycles, it would appear as if the CNN cannot effectively defend against repeated attacks.

Note, however, that we only test the attack-rebroadcast images $\mathcal{R}_i$ against a single CNN $\mathcal{D}_i$. When we test $\mathcal{R}_i$ against other CNNs in the earlier detect/attack cycles, we find reason for hope. The white bars in Fig. 1(a) correspond to the detection accuracy of $\mathcal{R}_i$ against all classifiers in the earlier detect/attack cycles. The attack-rebroadcast images $\mathcal{R}_6$, for example, are thoroughly mis-classified by $\mathcal{D}_6$ but are correctly classified at a high rate by $\mathcal{D}_2$ through $\mathcal{D}_5$. Perhaps this shouldn't be surprising since the attack is designed to circumvent a single classifier, $\mathcal{D}_6$.

In the next section, we will explore a detect/attack cycle in which the attacker now has to attack all previous classifiers to avoid detection.

B. Multiple attacks

In the previous section we see that a gradient descent attack is successful at defeating a single detector but not all previous detectors in the cycle. In this section we will test the efficacy of attacking all detectors in the cycle. The training of each detector $\mathcal{D}_i$ is the same as in the previous section. The attack A1 in the first cycle is also the same, but subsequent iterations differ in that instead of attacking a single CNN, the attacker simultaneously attacks all previous CNNs in the cycle:

(A2) In this second attack, attack-rebroadcast images $\mathcal{R}_2$ are generated from $\mathcal{R}$ by attacking $\{\mathcal{D}_1, \mathcal{D}_2\}$ using the gradient descent method described below.

(Ai) In this attack, attack-rebroadcast images $\mathcal{R}_i$ are generated from $\mathcal{R}$ by attacking $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_i\}$.

The extension from attacking a single CNN to multiple CNNs is straightforward. In the i^{th} detect/attack cycle, a rebroadcast image I is modified such that it will be classified as original by all previous detectors: $f_j(I) < 0$, for $j = 1, 2, \cdots, i$. As before, the input to the j^{th} CNN is classified as original or rebroadcast based on the sign of the following function:

$$f_j(I) \quad = \quad \vec{v}^T \phi_j(I), \tag{4}$$

where $\vec{v}^T = \begin{pmatrix}-1 & 1\end{pmatrix}$, and $\phi_j(I)$ is the output of the j^{th} CNN $\mathcal{D}_j$.

Following a similar approach as in the previous section, the gradient descent method with momentum iteratively updates the solution according to the following update rule at the k^{th} iteration ($k = 0, 1, 2, \cdots$):

$$I^{k+1} \quad = \quad I^k - \sum_{j=1}^{i} \alpha_j \left(m_j f_j'(I^{k-1}) + f_j'(I^k) \right), \tag{5}$$

where m_j is the momentum, α_j is the learning rate, and $f_j'(\cdot)$ is the gradient of Equation (4).

The gradient descent is initialized with $I^0 = I$, $f_j'(I^{-1}) = 0$, and momentum $m_j = 0.9$. The learning rate is initialized to $\alpha_j = 1e^{-4}$ and is decreased by a factor of 0.9 when the loss $f_j(\cdot)$ plateaus. When the learning rate α_j is reduced, the momentum is set to $m_j = 0$ for that iteration and reset to $m_j = 0.9$ in subsequent iterations. The gradient descent iteration terminates under any of the following conditions: (1) the modified rebroadcast image is classified as original by all CNNs; (2) all of the learning rates α_j are less than predefined threshold of $1e^{-8}$; or (3) the number of iterations k exceeds a predefined threshold of $1,000$.

We carry out three detect/attack cycles. Shown in Fig. 1(b) are detection accuracies of three CNN detectors on different images. Each detector $\mathcal{D}_i$ is trained on the images rendered as light gray bars. As before, detection accuracy for each detector remains high on these images (above 97%). By only the second iteration, we see that the attacker is struggling to defeat the detectors. In particular, although the attack-rebroadcast images $\mathcal{R}_3$ are able to mostly circumvent detection by $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$, we see that it comes at the price of a high MSE of 102.7. That is, in order to circumvent detection, the images have to be significantly modified in appearance which presumably would be easily flagged as suspicious. We only perform three iterations because on the third iteration the MSE is so large that further iterations seem unlikely to yield an effective attack.

At this point, it seems that the defender has the upper hand. In the next section, we briefly explore strategies that the attacker can employ to defeat the defender.

C. Multiple (slow) attacks

In the previous section, the CNN learning rate is initialized to $\alpha_j = 1e^{-4}$. We hypothesized that a slower learning rate may benefit the attacker allowing them to both circumvent detection while minimizing the MSE between the rebroadcast and attack-rebroadcast images. Shown in Fig. 1(c) are the results of the detect/attack cycles described in the previous section with a learning rate of $\alpha_j = 1e^{-5}$. As before, the CNN detectors can accurately classify the content on which they are trained, but fail to detect future attacks. And, the slower learning rate yields significantly lower MSEs, between 0.1 and 3.5. With this lower learning rate, the attacker is victorious. It remains to be seen if even more detect/attack iterations will yield larger and prohibitive MSEs.

V. Conclusion

A CNN is able to reliably detect rebroadcast attacks. This CNN, however, is vulnerable to a simple counter-forensic attack in which a rebroadcast image is modified to appear as an original image. In repeated detect/attack cycles, the attacker seems to eventually succeed at circumventing detection. Across these cycles, however, the modified attack-rebroadcast image degrades in quality.

Although it appears that the attacker has the upper hand, we assume that the attacker has full knowledge of the defender (the CNN). It remains to be seen if the attacker can successfully circumvent detection with partial or no knowledge of the defender. Lastly, our attack only modifies a small image block. It remains to be seen if the attacker can seamlessly piece these blocks together to create a full-size adversarial image.

Acknowledgment

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA FA8750-16-C-0166). The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] H. Farid, *Photo Forensics*. MIT Press, 2016.

[2] J. Tešić, “Metadata practices for consumer photos,” *IEEE Multimedia Magazine*, vol. 12, pp. 86–92, 2011.

[3] E. Kee, M. K. Johnson, and H. Farid, “Digital image authentication from JPEG headers,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1066–1075, 2011.

[4] T. Gloe, “Forensic analysis of ordered data structures on the example of JPEG files,” in *IEEE International Workshop on Information Forensics and Security*, 2012, pp. 139–144.

[5] A. Tuama, F. Comby, and M. Chaumont, “Camera model identification with the use of deep convolutional neural networks,” in *IEEE International Workshop on Information Forensics and Security*, 2016, pp. 1–6.

[6] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, “First steps toward camera model identification with convolutional neural networks,” *IEEE Signal Processing Letter*, vol. 24, no. 3, pp. 259–263, 2017.

[7] E. Kee and H. Farid, “Digital image authentication from thumbnails,” in *Proc. SPIE, Media Forensics and Security II*, 2010.

[8] B. C. Chen, P. Ghosh, V. I. Morariu, and L. S. Davis, “Detection of metadata tampering through discrepancy between image content and metadata using multi-task deep learning,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1872–1880.

[9] A. C. Popescu and H. Farid, “Exposing digital forgeries in color filter array interpolated images,” *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, 2005.

[10] M. Kirchner, “Efficient estimation of CFA pattern configuration in digital camera images,” in *Proc. SPIE, Electronic Imaging, Media Forensics and Security*, 2010.

[11] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro, “Aligned and non-aligned double JPEG detection using convolutional neural networks,” *Journal of Visual Communication and Image Representation*, vol. 49, pp. 153–163, 2017.

[12] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli, “Localization of JPEG double compression through multi-domain convolutional neural networks,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1865–1871, 2017.

[13] H. Farid and S. Lyu, “Higher-order wavelet statistics and their application to digital forensics,” in *IEEE Workshop on Statistical Analysis in Computer Vision (in conjunction with CVPR)*, vol. 8, 2003, pp. 94–94.

[14] H. Cao and A. C. Kot, “Identification of recaptured photographs on LCD screens,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010, pp. 1790–1793.

[15] J. Yin and Y. Fang, “Markov-based image forensics for photographic copying from printed picture,” in *ACM International Conference on Multimedia*, 2012, pp. 1113–1116.

[16] X. Gao, T. T. Ng, B. Qiu, and S. Chang, “Single-view recaptured image detection based on physics-based features,” in *IEEE International Conference on Multimedia and Expo*, 2010, pp. 1469–1474.

[17] J. Yin and Y. Fang, “Digital image forensics for photographic copying,” in *Proc. SPIE, Media Watermarking, Security, and Forensics*, 2012, p. 83030F.

[18] B. Mahdian, A. Novozámskỳ, and S. Saic, “Identification of aliasing-based patterns in re-captured LCD screens,” in *IEEE International Conference on Image Processing*, 2015, pp. 616–620.

[19] T. Thongkamwitoon, H. Muammar, and P. Dragotti, “An image recapture detection algorithm based on learning dictionaries of edge profiles,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 953–968, 2015.

[20] P. Yang, R. Ni, and Y. Zhao, “Recapture image forensics based on Laplacian convolutional neural networks,” in *International Workshop on Digital Watermarking*, 2016, pp. 119–128.

[21] X. Zhai, R. Ni, and Y. Zhao, “Recaptured image detection based on texture features,” in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2013, pp. 234–237.

[22] Y. Ke, Q. Shan, F. Qin, and W. Min, “Image recapture detection using multiple features,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 5, pp. 71–82, 2013.

[23] H. Li, S. Wang, and A. C. Kot, “Image recapture detection with convolutional and recurrent neural networks,” *Proc. Electronic Imaging, Media Watermarking, Security, and Forensics*, vol. 2017, no. 7, pp. 87–91, 2017.

[24] S. Agarwal, W. Fan, and H. Farid, “A diverse large-scale dataset for evaluating rebroadcast attacks.” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2018.

[25] Z. Guo, L. Zhang, and D. Zhang, “Rotation invariant texture classification using LBP variance (LBPV) with global matching,” *Pattern Recognition*, vol. 43, no. 3, pp. 706–719, 2010.

[26] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[27] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[28] Y. Q. Shi, C. Chen, and W. Chen, “A Markov process based approach to effective attacking JPEG steganography,” in *International Conference on Information Hiding*, 2006, pp. 249–264.

[29] K. Wang, “A simple and effective image-statistics-based approach to detecting recaptured images from LCD screens,” *Digital Investigation*, vol. 23, pp. 75–87, 2017.

[30] Z. Chen, B. Tondi, X. Li, R. Ni, Y. Zhao, and M. Barni, “A gradient-based pixel-domain attack against SVM detection of global image manipulations,” in *IEEE International Workshop on Information Forensics and Security*, 2017, pp. 1–6.

[31] “PyTorch,” http://pytorch.org/.