CNN-BASED DETECTION OF GENERIC CONTRAST ADJUSTMENT WITH JPEG POST-PROCESSING

*M.Barni[#], A.Costanzo[†], E.Nowroozi[#], B.Tondi[#]

[#] Department of Information Engineering and Mathematics
University of Siena
[†] CNIT - Consorzio Nazionale Interuniversitario per le Telecomunicazioni

ABSTRACT

Detection of contrast adjustments in the presence of JPEG post processing is known to be a challenging task. JPEG post processing is often applied innocently, as JPEG is the most common image format, or it may correspond to a laundering attack, when it is purposely applied to erase the traces of manipulation. In this paper, we propose a CNN-based detector for generic contrast adjustment, which is robust to JPEG compression. The proposed system relies on a patch-based Convolutional Neural Network (CNN), trained to distinguish pristine images from contrast adjusted images, for some selected adjustment operators of different nature. Robustness to JPEG compression is achieved by training a JPEG-aware version of the CNN, i.e., feeding the CNN with JPEG examples, compressed over a range of Quality Factors (QFs). Experimental results show that the detector works very well under a wide range of QFs and scales well with respect to the adjustment type, yielding very good performance under a large variety of unseen tonal adjustments.

Index Terms— Adversarial multimedia forensics, adversarial learning, deep learning for Multimedia Forensics, transfer-learning, contrast manipulation detection, cybersecurity.

I. INTRODUCTION

Adjustment of contrast and lighting conditions of image subparts is often performed during forgery creation. Therefore, the problem of detecting such manipulation has been widely studied in image forensics, and, more recently, in scenarios encompassing the presence of an adversary [1], [2]. Due to the peculiar traces left by contrast adjustment operators, most early works were based on the analysis of first order statistics [3]–[5]. Such approaches, however, are easily circumvented by the adversary, by means of both targeted [6] and also universal approaches [7]. To cope with such attacks, countermeasures were developed in turn, based on a second-order analysis [8], [9]. However, in most cases, the attack is of laundering-type, consisting in the application of a post-processing operation, e.g., a geometric transformation, filtering or compression. Laundering attacks are shown to be very powerful against manipulation detectors in general [10]. In particular, the performance of contrast manipulation detectors proposed so far decreases significantly in the presence of even mild post-processing and, above all, they all exhibit a poor robustness against JPEG compression [3], [5], [8], [10], [11], even when the compression is very weak. Since images are often stored and distributed in JPEG format, JPEG compression is also one of the most common post-processing images are subject to. Therefore, designing a JPEG-robust contrast adjustment detector is of primary importance.

In this paper, we face with the above problem by resorting to adversary-aware data-driven classification [12], that is, by designing a data driven detector for contrast adjustment which is trained to recognise the specific class of JPEG laundering attacks. In particular, we look for a generic detector of contrast adjustment, that is, a detector which generalizes well to a wide variety of types of tonal adjustments. Such adversarial detection task is not trivial and requires that highly descriptive features are adopted. This is hard to accomplish with standard machine learning classifiers trained on rich image representations (e.g. the rich feature models [13]). Then, in this paper, we propose to rely on a Convolutional Neural Network (CNN) architecture. The CNN is directly fed by the pixel image (with no pre-processing), hence the discriminative features for our problem are self-learned by the CNN. Specifically, the proposed detector relies on a JPEG-aware, patch-based CNN, which is used to classify image regions, i.e. image patches. A test image is then divided into patches which are tested separately by feeding them to the CNN. The soft patch scores (CNN outputs) are collected and the global decision on the image is performed on the score vector.

All the compression QFs inside a range of values are considered to train the aware CNN. Noticeably, the performance of the CNN could be improved by exploiting the knowledge of the QF, which can be estimated from the image header, and specializing the CNN to work for one QF (hence training several CNNs). However, such an approach has the drawback of being easily prone to attacks: just re-saving the image in uncompressed format (e.g., PNG,..) or compressing again the image with a different QF would prevent the correct identification of the QF used to compress the image. Therefore, for our global manipulation detection task, we considered only one CNN model; the final detection accuracy is raised by exploiting the fact that patches coming from a same image are generated under the same hypothesis (being all pristine or contrast adjusted patches), and hence should all result in a small (or large) soft value as CNN output.

Experiments show that our system achieves good performance over a wide range of QFs. Thanks to the fact that the CNN is simultaneously trained with different contrast adjustments, our detector achieves good scalability with respect to the contrast adjustment types, yielding good performance with a large variety of contrast, brightness and tonal adjustments, i.e. under processing mismatch conditions. Good performance are maintained in the absence of JPEG, that is, when the contrast adjustment is the last step of the manipulation chain.

The rest of the paper is organized as follows: Section II, we define the detection task addressed, describe the proposed CNN-based detector and the network architecture. In Section III-A, we first detail the methodology followed for conducting our experiments, then we report and discuss the results.

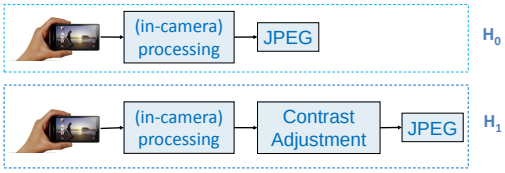

Fig. 1. Detection task considered in this paper.

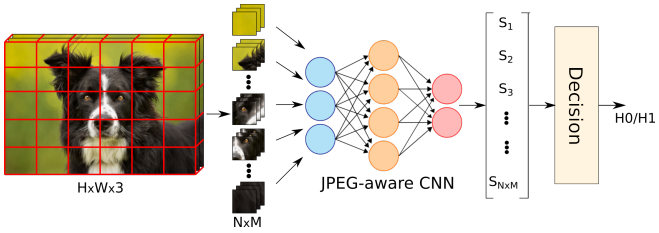

Fig. 2. Pipeline of the proposed generic, JPEG-aware, contrast adjustment detector. Adaptive histogram equalization, gamma correction (both compression and expansion) and histogram stretching are used to train the network.

II. PROPOSED SYSTEM

Figure 1 schematizes the problem addressed in this paper. We let hypothesis H_0 correspond to the case of pristine image and H_1 to the case of contrast adjusted images In both cases, the image is JPEG compressed at the end, with a given QF. In this scheme, JPEG compression can also be viewed as a counter-forensic, laundering-type, attack, due to its known effectiveness in erasing the traces of contrast manipulations [3], [5], [8], [10], [11].

The architecture of the proposed detection scheme is reported in Figure 2. The color image is divided into non-overlapping patches of size 64×64 which are fed to a JPEG-aware version of a CNN. The patch scores, i.e. the CNN soft outputs for all the patches, are then collected and the final decision is based on the score vector $s = (s_1, s_2, ...s_{N \times M})$ (where $N \times M$ is the total number of blocks). The decision is made by simply thresholding the sum of the scores, i.e. according to the statistic[1] $\sum_i s_i/(M \cdot N)$. Since patches coming from a same image are drawn under the same hypothesis, such normalized sum is expected to be large in one case (contrast adjusted image) and small in the other (pristine image).

The JPEG-aware CNN is trained with JPEG compressed images on one hand (H_0) and images subject to a contrast adjustment followed by JPEG compression on the other (H_1). The network architecture and the training strategy are detailed in the following sections In the attempt to build a detector which generalizes to unseen adjustments (transfer-learning), we consider contrast adjustments of different nature to train the network. Specifically, the processing we selected are: adaptive histogram equalization, gamma correction (both compression and expansion of the contrast) and histogram stretching.

Regarding the compression QF, we focus on values ranging from medium-high to high values (i.e., $QF \geq 80$), which are common values in many practical applications.

[1]This is a simple and non optimized choice. Other possible fusion strategies could be adopted.

II-A. CNN architecture

First attempts to train a network for our problem by using architectures similar to those adopted for other forensic tasks [14]–[16] were unsuccessful. A possible explanation is the following: while some processing operations, e.g. local filtering and double JPEG, introduces some local patterns that a properly trained CNN with few layers is able to 'easily' learn, common contrast adjustments do no leave local visual artifacts, thus making self CNN learning harder and calling for the adoption of deeper models. We were in fact able to get higher accuracies by switching to deeper architectures, inspired by those adopted in image classification and pattern recognition applications [17].

The structure of our network for patch classification (see Figure 3) is detailed as follows: it takes a color patch of size 64×64 as input and consists of

- **5 convolutional layers** followed by a **max-pooling** layer. In the first convolutional layer 32 filters are applied. The number of filters increases by 32 at each layer. For all the filters, the kernel size is 3×3 and the stride is always 1. Max-pooling is applied with kernel size 2×2 and stride 2 producing a final $27 \times 27 \times 160$ feature map.
- **3 convolutional layers** followed by a **max-pooling** layer. As before, the number of filters of size 3×3 and stride 1 increases by 32 at each layer. The pooling is the same as before, yielding a $10 \times 10 \times 256$ feature map.
- A final **convolutional layer** with a filter of size $1 \times 1 \times 128$ generating a $10 \times 10 \times 128$ feature map.
- A **fully-connected layer** with 250 input neurons and 2 output neurons, followed by a softmax layer.

Some comments regarding the main features of the above architecture are in order: the use of many convolutions (5) before the first pooling layer permits to consider a large receptive field for each neuron, which is good to capture relationships among pixels in large neighborhoods; the stride 1 permits to retain as much spatial information as possible. The purpose of the final convolutional layer is to reduce the number of parameters by halving the number of maps (from 256 to 128), without affecting spatial information. The adoption of only one fully connected layer also permits to reduce the number of parameters without affecting too much the performance. Finally, we observe that using small patches (64×64) permitted us to increase the depth of the network for the same number of parameters.

II-B. CNN training strategy

We obtained the JPEG-aware CNN model by training the network in two steps. First, the network is trained to recognize between patches coming from pristine and contrast-adjusted images for the uncompressed case (unaware pre-training). Then, the aware model is obtained by fine-tuning the unaware network, by feeding the CNN with JPEG compressed examples of the above classes.

Since the network is very deep and then the number of images used for training is very large, we perform compression on-the-fly by augmenting the data inside the network; hence, the compression is performed directly on the 64×64 patches (that is, after the image splitting). Such a strategy is viable because the JPEG compression is a local operator which can be applied separately on multiples of 8×8 image patches producing the same result as applied on the entire image.

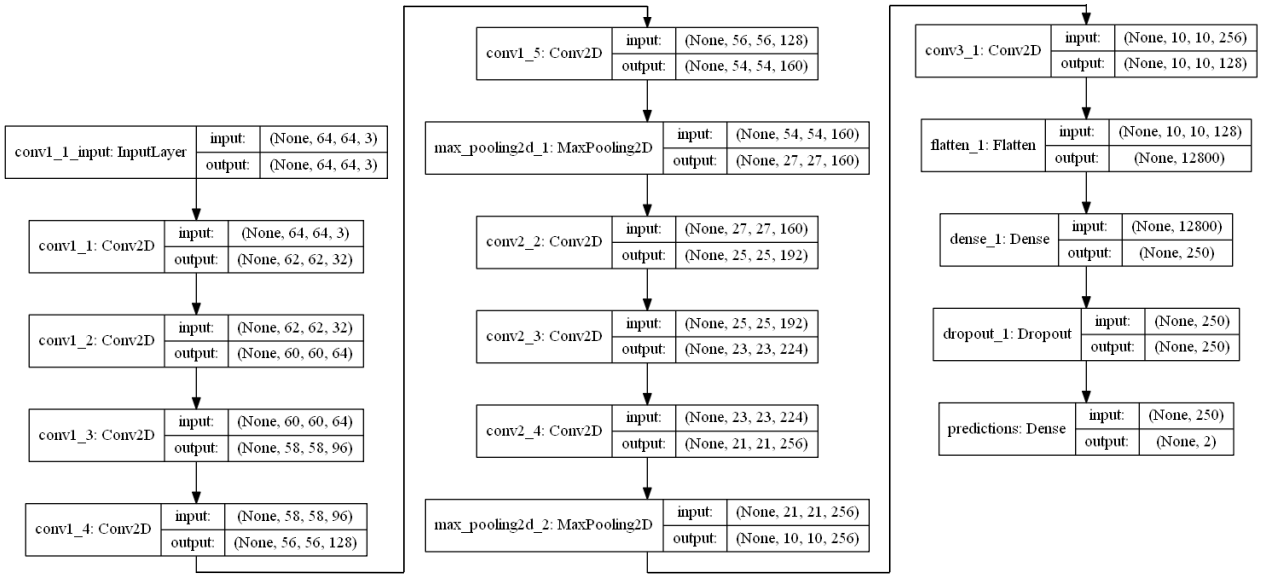

Fig. 3. Architecture of the proposed network.

III. EXPERIMENTS

III-A. Methodology

We built the training and testing sets by starting from color images in uncompressed format. The images for the H_0 and H_1 classes were produced as detailed in Figure 1. The adjustment of the contrast under H_1 is obtained by considering several algorithms. As we said, to generate the images used for training, we considered the following operators: Adaptive Histogram Equalization (in particular, its refined, Contrast Limited, implementation, `CLAHE` [18]), Gamma Correction (γ `Corr`), and Histogram Stretching (`HS`). Such operators are designed for one-channel images; to make them work on color images, we apply them as follows: for the images processed with `CLAHE`, we first converted the image from the RGB to the HSV color space, we applied the enhancement to the luminance channel only, namely the V channel, and converted back to the RGB domain [2]. The same strategy is adopted to generate the images processed with `HS`. Finally, for the γ `Corr`, the contrast is modified by applying the operator to each channel (R, G and B) separately. The above operators are applied in equal percentage to generate the class of contrast adjusted images (H_1). Regarding the parameters, the clip-limit parameter for `CLAHE` is set to 0.005, the γ value to 1.5 and 0.7 (randomly chosen with probability 0.5), and the saturation percentage of the `HS` to 5% for both black and white values. The above choices do not introduce visually unpleasant artifacts.

For generating the test images, we also considered different values of the parameters for the same operators, in order to assess the performance under parameter mismatch, and different operators, by processing the images with adjustment tools provided by Photoshop. In particular, we considered the following tonal adjustments:

- `AutoContrast`, `AutoColor` and `AutoTone`; algorithms which operate differently with respect to the color channels. The clipping is set to 7% for `AutoContrast` and `AutoColor` and to 5% for `AutoTone`; the snap neutral midtones option is selected for the `AutoColor`;
- `Curves_S`; a (hand-made) smooth S-curve is applied to enhance the contrast in the midtones;
- `Brightness` and `Contrast`; generic tools for enhancing and reducing brightness and contrast; for the enhancement, we set `Brightness` to 50 (`Brightness+`) and `Contrast` to 70 (`Contrast+`), while for the reduction, we set `Brightness` to -70 (`Brightness-`) and `Contrast` to -50 (`Contrast-`);
- Histogram Equalization (`HistEq`).

The `HistEq` manipulation is considered for completeness: although its visual impact is much stronger with respect to that of the other manipulations considered, and hence is rarely adopted in practice, the `HistEq` manipulation is often considered in multimedia forensic literature.

Regarding JPEG compression, we random selected the QFs (uniformly) in the range $[90, 100]$ to compress the images used for training. For testing, we also considered images compressed with QF 85 and 80.

The images used for training were all processed with the OpenCV library for Python. For the tests, the Photoshop software was also adopted. We used TensorFlow, via the Keras API [19], to train and test the CNN.

III-B. Results

Uncompressed, camera-native, images (.tiff) are taken from the RAISE8K dataset [20] (of size 4288×2848), splitted into training and test set, and then contrast-adjusted to produce the images for H_1 in the unaware case (i.e., without the final JPEG). The images are then divided into 64×64 patches for CNN training and testing: 2×10^6 patches per class (coming from more than 1000

[2]The straightforward application of `CLAHE` (and `HS`) to each channel separately unnaturally changes the color balance and produces visually unpleasant images.

Table I. Performance (AUC) of the detector under matched processing. The matched parameters are in bold.

		QF						
		no jpg	**100**	**98**	**95**	**90**	**85**	**80**
CLAHE	0.003	100	99.9	99.8	98.9	97.6	97.1	96.8
	0.005	100	99.9	99.9	99.4	98.9	98.8	98.5
	0.007	100	99.9	100	99.6	99.1	98.9	98.7
γ Corr	**1.5**	98.8	98.5	94.2	89.2	87	84	81.2
	1.7	99.4	98.9	95.7	91.8	90.4	89.7	89.2
	0.7	99.1	97.1	92.3	87.3	85.6	81	78
	0.6	99.7	99.5	97.3	91.6	86.7	83.7	80.1
HS (%)	3	99.6	98.1	95.8	91.4	87.8	85.7	83.5
	5	99.5	98.9	97.6	93.7	92.6	91.5	90.3
	7	100	99.3	98.3	95.5	94	93.7	93.6

Table II. Performance (AUC) of the detector for different tonal adjustments.

	QF						
	no jpg	**100**	**98**	**95**	**90**	**85**	**80**
HistEq	100	99.9	99.9	99.5	98.3	96.9	94.8
Brightness+	97.5	97.7	95.2	93.6	91.2	87.8	85.6
Contrast+	99.1	100	99.6	97.9	94.7	91.9	87.1
Brightness-	96.7	97.3	93.3	90.1	84.2	78.8	75.6
Contrast-	98.8	99.6	96.4	91.2	87	82	80
Curve_S	99.6	99.8	99.8	99.1	97.7	96	93.6
AutoContrast	95.9	94.7	93	91.9	90.2	89	86.5
AutoColor	98.2	98.6	96.8	95.3	93.7	91.8	89.1
AutoTone	99.5	99.5	99	98.2	97.2	96.1	94.5

training images) were selected to train the CNN, whereas 2×10^5 patches were used for testing. In the aware case, the patches are JPEG compressed with QF $\in [90, 100]$. The overall performance of the detector are tested on 300 images from the test set, both uncompressed and compressed with QF = $\{100, 98, 95, 90, 85, 80\}$.

When training and testing are performed with uncompressed images (unaware case), the average test accuracy of the CNN on image patches is $93,5\%$, where the average is taken on the 3 manipulations, i.e., CLAHE, γ Corr (compression and expansion) and HS, and on all the QFs inside the training range. For the overall system, we get almost perfect classification, that is, the Area Under Curve (AUC) is $99,8\%$, which is in line with the state of the art [10]. A noticeable strength is that here these performance are achieved by one (generic) system only, rather than using separate systems each one specialized on one manipulation only. By testing the unaware detector with JPEG compressed images, the performance drop to AUC = 56% (the CNN accuracy on image patches is around 50% even for weak compression), thus showing that, as it is the case with SVM detectors [10], the CNN model is not robust to the JPEG laundering attack.

Concerning the aware case, the average accuracies that we obtained at the patch level in the range of QFs $[90, 100]$ are: 0.84 for CLAHE, 0.72 for γ Corr and 0.79 for HS. These accuracies are not high; however, we stress that we do not need them to be very high, since the detection accuracy is then raised by the final decision stage (see Figure 2). What is it more important is that the performance are moderately good with respect to all the contrast adjustment operators. We also observe that specializing our network to work with one QF only, we could have obtained much higher performance on a patch level; however, as motivated in the introduction, to avoid easy attacks (as recompression and saving in uncompressed formats), we want a detector of generic contrast adjustments which works well on a range of QFs (and also in the absence of JPEG).

The overall performance of the detector on full images are reported in Table I in terms of AUC values, for both matched and mismatched processing parameters. The most difficult case corresponds to γ Corr, where the AUC is below 90% for QF ≤ 95. This behavior is due to the fact that such kind of adjustment is difficult to detect by itself and above all to the fact that the CNN is simultaneously trained with values smaller and larger than 1, corresponding to a compression and an expansion of the contrast.[3] We observe that the performance are good in the presence of a mismatch in the processing parameters, obtaining better classification when the adjustment is stronger than in the matched case, and worse when it is weaker. The performance remain very good in the absence of JPEG, in line with those achieved by the unaware detector (AUC = 99.6% on the average in the matched case). Expectedly, performance decreases as QF decreases. However, good robustness to JPEG compression is achieved (at least for CLAHE and HS) also when the QF is 85 and 80, which are outside the training range. Table II shows the results under various contrast/brightness adjustment performed with Photoshop. Based on these results, the CNN-based detector scales very well with respect to the adjustment type maintaining good performance when the tones of the image are adjusted in different ways and, possibly, selectively in different tonal ranges, and when the adjustment operates differently on the color channels.

IV. CONCLUSIONS

In this paper, we proposed an adversary-aware CNN-based approach to cope with the well known problem of detection of contrast adjusted images in the presence of JPEG post-processing. To accomplish this task, we exploited the superior capabilities of CNN architectures with respect to classical machine learning tools. In order to build a detector which works well for generic contrast adjustment, we trained the CNN with a certain number of adjustments of different nature. Results show that our detector achieves good performance over a wide range of QFs and generalizes well to unseen tonal adjustments. As further research, it would be interesting to see if the performance with respect to the most difficult cases can be improved by refining the composition of the training, i.e., the types of contrast adjustments considered, their proportions, and the distribution of the QF over the range, and also the fusion strategy at the final stage. As a future work, we would like to improve the performance on a patch level to move from detection to localization.

V. REFERENCES

[1] R. Böhme and M. Kirchner, "Counter-forensics: Attacking image forensics," in *Digital Image Forensics*, H. T. Sencar and N. Memon, Eds. Springer Berlin / Heidelberg, 2012.

[3] We verified that if the detector is trained with $\gamma = 1.5$ only (gamma expansion), the AUC for the γ Corr is above 97% for every QF ≥ 85. In this case, however, the performance with respect to the gamma compression ($\gamma < 1$) are very poor even with large QF (e.g. AUC $= 78\%$ for QF $= 95$).

[2] M. Barni and F. Pérez-González, "Coping with the enemy: advances in adversary-aware signal processing," in *ICASSP 2013, IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 26-31 May 2013, pp. 8682–8686.

[3] Matthew C Stamm and KJ Ray Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 492–506, 2010.

[4] Gang Cao, Yao Zhao, and Rongrong Ni, "Forensic estimation of gamma correction in digital images," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 2097–2100.

[5] Gang Cao, Yao Zhao, Rongrong Ni, and Xuelong Li, "Contrast enhancement-based forensics in digital images," *IEEE transactions on information forensics and security*, vol. 9, no. 3, pp. 515–525, 2014.

[6] Gang Cao, Yao Zhao, Rongrong Ni, and Huawei Tian, "Anti-forensics of contrast enhancement in digital images," in *Proceedings of the 12th ACM Workshop on Multimedia and Security*. ACM, 2010, pp. 25–34.

[7] Mauro Barni, Marco Fontani, and Benedetta Tondi, "A universal technique to hide traces of histogram-based image manipulations," in *Proceedings of the on Multimedia and security*. ACM, 2012, pp. 97–104.

[8] Xunyu Pan, Xing Zhang, and Siwei Lyu, "Exposing image forgery with blind noise estimation," in *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security*. ACM, 2011, pp. 15–20.

[9] Alessia De Rosa, Marco Fontani, Matteo Massai, Alessandro Piva, and Mauro Barni, "Second-order statistics analysis to cope with contrast enhancement counter-forensics," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1132–1136, 2015.

[10] Haodong Li, Weiqi Luo, Xiaoqing Qiu, and Jiwu Huang, "Identification of various image operations using residual-based features," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[11] Neetu Singh and Abhinav Gupta, "Analysis of contrast enhancement forensics in compressed and uncompressed images," in *Signal Processing and Communication (ICSC), 2016 International Conference on*. IEEE, 2016, pp. 303–307.

[12] M. Barni, E. Nowroozi, and B. Tondi, "Higher-order, adversary-aware, double jpeg-detection via selected training on attacked samples," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 281–285.

[13] Jessica Fridrich and Jan Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[14] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849–1853, Nov 2015.

[15] Belhassen Bayar and Matthew C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, New York, NY, USA, 2016, IH&MMSec '16, pp. 5–10, ACM.

[16] Mauro Barni, Luca Bondi, Nicolò Bonettini, Paolo Bestagini, Andrea Costanzo, Marco Maggini, Benedetta Tondi, and Stefano 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.

[17] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] Karel Zuiderveld, "Graphics gems iv," chapter Contrast Limited Adaptive Histogram Equalization, pp. 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

[19] François Chollet et al., "Keras," https://github.com/keras-team/keras, 2015.

[20] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato, "Raise: A raw images dataset for digital image forensics," in *Proceedings of the 6th ACM Multimedia Systems Conference*, New York, NY, USA, 2015, MMSys '15, pp. 219–224, ACM.