DF2023: The Digital Forensics 2023 Dataset for Image Forgery Detection

David Fischinger and Martin Boyer

Austrian Institute of Technology

Abstract

The deliberate manipulation of public opinion, especially through altered images, which are frequently disseminated through online social networks, poses a significant danger to society. To fight this issue on a technical level we support the research community by releasing the Digital Forensics 2023 (DF2023) training and validation dataset, comprising one million images from four major forgery categories: splicing, copy-move, enhancement and removal. This dataset enables an objective comparison of network architectures and can significantly reduce the time and effort of researchers preparing datasets.

Keywords: Image Manipulation Detection, Training Dataset, Benchmark Dataset, DF2023

1 Introduction

The proliferation of fake news presents a mounting concern in our society. Advances in technology have facilitated the swift and seamless production of convincing counterfeit digital media content, encompassing audio, video, and images. This impact spans from humorous satirical memes to organized political campaigns that disseminate fabricated news in order to manipulate public sentiment.

This paper addresses the issue of identifying local image forgeries. Over the past decade, several methods have been proposed in order to detect the main categories of image forgery: copy-move [Li et al., 2013], splicing [Lyu et al., 2013], inpainting [Li et al., 2017] and other specific filtering techniques, subsumed as enhancement [Sun et al., 2018]. However, these detection methods often concentrate on specific characteristics of each manipulation type. In recent years, more comprehensive approaches capable of detecting multiple types of manipulation have emerged, such as those presented in [Wu et al., 2019] and [Wu et al., 2022].

However, the research community is still in need of a large and more generalized dataset which enables training and hence an objective comparison of network architectures for the issue of image forgery detection. In this paper, we close this gap by introducing the Digital Forensics 2023 (DF2023) dataset. This training dataset is comprised of one million manipulated images specifically designed for image forgery detection and localization. By making the DF2023 dataset publicly available, it provides the research community with the means to conduct unbiased comparisons of network architectures and reduces the time and effort required for preparing training data.

2 Related Work

Many methods of detecting and localizing image forgery were published (see, for example, the reviews of [Zanardelli et al., 2022] and [Verdoliva, 2020] and references therein) in order to ensure visual information authenticity.

While there are a number of established benchmark datasets in the field of image forgery detection [Dong et al., 2013,Hsu and Chang, 2006,Carvalho et al., 2013,National Institute of Standards and Technology (NIST), 2016], proposed datasets are limited in size and manipulation diversity, and are therefore not appropriate as training datasets. Table 1, partly taken from [Novozamsky et al., 2020], gives an overview of available datasets designed for image forgery detection. Proposed datasets from literature which are not accessible anymore were removed from the table. As shown, the tampCoco [Kwon et al., 2022] dataset and the Defacto [Mahfoudi et al., 2019] dataset are by far the largest available datasets. The Defacto [Mahfoudi et al., 2019] dataset has about 190,000 images. However, 39,800 images of this dataset are very specific face morphing forgeries. The forgery type enhancement, on the other side, was not specifically included in the dataset. The tampCoco dataset has just been released on Kaggle on March 28, 2023. The dataset is derived from the MS-COCO dataset [Lin et al., 2014] and was generated by applying the manipulation techniques of splicing and copy-move operations.

Dataset of manipulated images	**Size**	**Manip-Types**
Coverage [Wen et al., 2016]	100	C
CoMoFoD [Tralic et al., 2013]	260	C
DSO [Carvalho et al., 2013]	100	SE
Columbia [Hsu and Chang, 2006]	160	S
CASIA [Dong et al., 2013]	920	SCE
CASIA v2.0 [Dong et al., 2013]	5,123	SCE
MICC-F220, MICC-F2000 [Amerini et al., 2011]	2,200	C
Zhou et al. [Zhou et al., 2017]	3,410	SE
NIST16 [National Institute of Standards and Technology (NIST), 2016]	564	SCR
OpenMFC20_Image_MD [Guan et al., 2019]	16,075	SCR
OpenMFC22_SpliceImage_MD [Guan et al., 2019]	2,000	S
IMD2020 Manually Created [Novozamsky et al., 2020]	2,010	SCRE
IMD2020 [Novozamsky et al., 2020]	35,000	R
Defacto [Mahfoudi et al., 2019]	189,387	SCR
tampCoco [Kwon et al., 2022]	800,000	SC
DF2023 Training (proposed)	1,000,000	SCRE
DF2023 Validation (proposed)	5,000	SCRE

Table 1: Examples of datasets designed for image manipulation detection with number of tampered images and manipulation types: (S)plicing, (C)opy-Move, (R)emoval, (E)nhancement

Considering the typical volume of training data required for deep neural networks to tackle complex tasks, the overview provided in Table 1 highlights the necessity for a sufficiently large training dataset that encompasses a diverse range of manipulations.

3 Digital Forensics Dataset - DF2023

The benefits of a large, diverse and public training dataset for detection of image forgeries are manifold: Researchers can save significant time by avoiding data collection, scripting and data generation. Using a pre-existing dataset prevents from consciously or unconsciously adjusting the training dataset to become too similar to the evaluation sets. Most importantly, such a dataset allows the decoupled evaluation and comparison of deep learning network architectures in an objective, transparent and (rather) reproducible way. For this reason, we introduce the Digital Forensics 2023 (DF2023) dataset, available from here: DF2023. The DF2023 training dataset contains one million forged images of the four main manipulation types. Specifically, the training dataset consists of 100K forged images produced by removal operations, 200K images produced by various enhancement modifications, 300K copy-move manipulated images and 400K spliced images. This distribution was selected based on our experience regarding the positive impact of each manipulation type on improving forgery detectors. The MS-COCO [Lin et al., 2014] 2017 training and validation datasets with 118K/5K images were facilitated as the source of pristine and donor images. Many other publicly available datasets in this research domain often lack comprehensive documentation, we on the other hand have chosen to provide a detailed description in the following sections on the meticulous process of creating the DF2023 dataset.

3.1 DF2023 - Dataset generation

1. Selection of pristine image:
A pristine image $\mathscr{I}_P$ was randomly selected from the MS-COCO 2017 training dataset, and respectively from the validation dataset. For the few images with width W or height H smaller than 256 pixels, the image was resized to the size $(max(W, 256), max(H, 256))$. For 50% of the images $\mathscr{I}_P$ in the training dataset, a proportion-preserving downscale was executed. This avoided extracting only small portions of larger images (like a monochrome patch depicting a part of the sky from the original image). The scaling of an image $\mathscr{I}_P$ with size (W, H) to (W_{new}, H_{new}) was done as follows:

$$\begin{aligned} W_{new} &= max(\lfloor(\frac{256 \cdot W}{min(W, H)})\rceil, 256) \\ H_{new} &= max(\lfloor(\frac{256 \cdot H}{min(W, H)})\rceil, 256) \\ \mathscr{I}_P &= \mathscr{I}_P.resize((W_{new}, H_{new})) \end{aligned} \tag{1}$$

Next, a patch of size $(256, 256)$ pixels was randomly chosen from the image $\mathscr{I}_P$ and used as a pristine image patch $\mathscr{P}$.

2. Selection of donor image:
A donor image $\mathscr{I}_D$ from MS-COCO (training/validation) was selected. For the splicing operation, a random image other than the pristine image $\mathscr{I}_P$ was selected. For the copy-move, removal and enhancement manipulations, the same pristine image was selected as a donor image ($\mathscr{I}_D = \mathscr{I}_P$).

3. Pre-processing of donor image:
Table 2 shows which preprocessing steps may be applied to the donor image $\mathscr{I}_D$ for each manipulation type. **Resample** rescaled the height and the width image dimensions independently by 70 to 130 percent. The size of the resulting image is at least $(256, 256)$. The preprocessing step **Flip** flipped the donor image horizontally with a likelihood of 50%, while **Rotate** rotated the image by either 90, 180 or 270 degrees with a likelihood factor controlled by a predefined parameter (for the DF2023 dataset, 30% of the donor images were rotated). **Blur** blurred the donor image with a likelihood of 50%. In case the blurring filter was applied, either `ImageFilter.BoxBlur` or `ImageFilter.GaussianBlur` from the Python package `PIL` were used, both with equal probabilities. The blur radius was set randomly between 1 and 7 pixels. **Contrast** used one of the ImageFilters `EDGE_ENHANCE`, `EDGE_ENHANCE_MORE`, `SHARPEN`, `UnsharpMask` or `ImageEnhance.Contrast` from the Python package PIL. **Noise** added Gaussian noise with mean and standard deviation $(\mu, \sigma) = (0, 12)$ with a likelihood of 1 out of 3. The **Brightness** was changed with a probability of 50% by a factor uniformly chosen in the range [0.5-1.5]. With 50% probability, a **JPEG-Compression** with quality factor $10x$ for $x \in [1, 2, 3, 4, 5, 6, 7]$ was employed. If the manipulation type was **Removal**, an inpainting filter from OpenCV [Bradski, 2000] was applied (either `cv2.INPAINT_TELEA` or `cv2.INPAINT_NS`) on the manipulation mask defined in step 5.
In case the chosen manipulation type was **Enhance** and none of the filters (blur, contrast, noise, brightness, JPEG compression) were applied to the donor image $\mathscr{I}_D$, the process was repeated.

Manipulation	**C**	**S**	**R**	**E**	**Pos.**	**values**	**e.g.**
Resample	×	×	–	–	1	0/1	0
Flip	×	×	–	–	2	0/1	0
Rotate	×	×	–	–	3	0/1/2/3	0
Blur	–	–	–	×	4-5	B/G, 0-9	G4
Contrast	–	–	–	×	6	0-5	0
Noise	–	–	–	×	7	0/1	1
Brightness	–	–	–	×	8	0/1	1
JPEG-Compression	–	–	–	×	9	0-9	7

Table 2: Preprocessing steps for donor image per manipulation types: Copy-Move (**C**), Splicing (**S**), Removal (**R**) and Enhancement (**E**). The column `Pos.` indicates the filenames encoding position for the corresponding manipulation (starting to count at the position for the manipulation type) as explained in Section 3.2. Column `values` and column `e.g.` show possible values and an example value, respectively, for the position in the name. For this example, a filename could be: COCO_DF_E000G40117_00200620.jpg

4. Cropping of donor patch:
Then, a donor patch $\mathscr{D}$ of size $(256,256)$ was randomly cropped from $\mathscr{I}_D$. For enhancement and removal (inpainting) manipulations, the donor patch $\mathscr{D}$ and the pristine patch $\mathscr{P}$ share the same location in $\mathscr{I}_D = \mathscr{I}_P$.

5. Creation of a binary manipulation mask:
Seven types of binary masks $\mathscr{M}$ were used to define the image region where manipulations were executed (see Table 3). In Table 4, various examples of the masks created and the resulting forged images are shown. Despite the five mask types which are based on geometric forms, we used Python's image processing toolbox scikit-image to segment the donor patch into Superpixels [Achanta et al., 2012] of appropriate size, and selected one Superpixel (connected set of pixels) as the defining mask where the manipulations would be applied on. Furthermore, the "object segmentation" used the segmentation ground truth from the MS-COCO dataset. All pixels from a donor image patch $\mathscr{D}$ which were marked corresponding to a specific object class (e.g. person) were selected and used as splicing input. The object category was randomly selected from the possible categories of the donor image, hence MS-COCO images with no labeled objects were excluded in case of object based mask creation.

Shape of Mask	**Parameters**	**Impact**
Triangle	p1, p2, p3	3 random points
Rounded Rectangle	X,Y, r	2 points for Bbox; radius of the corners
Ellipse	X, Y	2 points to define the bounding box
Polygon with 5 vertices	p1,...,p5	sequence of 5 random points
Ellipse + Polygon with 4 vertices	X,Y, p1,..,p4	ellipse + 4 vertex polygon
Superpixel Segmentation	[min, max]	range for number of Superpixels per image
Object Segmentation	obj. category	object category for segmentation (e.g. person)

Table 3: Types of mask shapes generated for local image manipulation

6. Creation of a non-binary manipulation mask:
For a smooth gradient at the edges of manipulation and as preparation for alpha blending, the manipulation masks $\mathscr{M}$ were blurred half of the time for splicing, copy-move and enhancement operations. This way, the transition from pristine image to manipulated patch is smooth and the forgery detection network is forced not to solely rely on sharp edges for identifying manipulated regions. Additionally, we applied alpha blending to make splicing manipulations more realistic and harder to detect. We achieved this by randomly setting an alpha value in the range [0.94, 1.0] and multiplying the manipulation mask with this floating point scalar value.

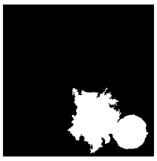
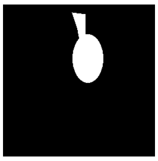
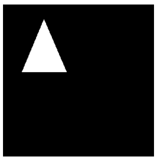
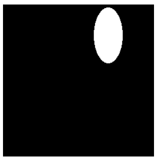
Forgery Type	Copy-Move 	Splicing 	Removal 	Enhance 	Removal 	Enhance 	Copy-Move
Shape of Forgery Mask	superpixel segmentat. 	object segmentation 	polygon 5 vertices 	ellipse + 4V polygon 	triangle 	rounded rectangle 	ellipse 

Table 4: Examples from the Digital Forensics 2023 (DF2023) dataset: The upper row shows the forged images and the applied manipulation type. The second row shows the corresponding manipulation mask and its shape.

Considerably stronger alpha blending led to worse results in our experiments. Finally, masks were recalculated if their size was below 5% or above 40% of the image patch.

7. Generation of forged image:
Given a pristine patch $\mathscr{P}$, a donor patch $\mathscr{D}$, a manipulation m and a binary manipulation mask $\mathscr{M}$, the forged image $\mathscr{X}$ is represented as

$$\mathscr{X} = (1 - \mathscr{M}) \cdot \mathscr{P} + \mathscr{M} \cdot m(\mathscr{D}) \tag{2}$$

meaning that each pixel of the resulting image $\mathscr{X}$ is taken either from the pristine patch $\mathscr{P}$ or the manipulated donor patch $\mathscr{D}$, depending on the binary mask $\mathscr{M}$. For alpha blending with a non-binary mask $\mathscr{M}$, the formula is still valid and combines pixels from the pristine and the donor image according to the mask values in the range $[0,1]$. In case of a copy-move manipulation, an additional translation of the copied image part $(1 - \mathscr{M}) \cdot m(\mathscr{D})$ is made towards another position in the pristine image patch.

8. Generation of ground truth:
Ground truth masks $\mathscr{M}_{GT}$ of (non-binary) manipulation masks $\mathscr{M}$, are defined as binary masks counting each non-zero value as 1, or as a Boolean matrix in NumPy notation:

$$\mathscr{M}_{GT} = (\mathscr{M} > 0) \tag{3}$$

3.2 DF2023 - Naming convention

The naming convention of DF2023 encodes information about the applied manipulations. The following convention is used for the image names:

$COCO\_DF\_0123456789\_NNNNNNNN.\{EXT\}$

For example:

$COCO\_DF\_E000G40117\_00200620.jpg$

After the identifier of the image data source ("COCO") and the self-reference to the Digital Forensics ("DF") dataset, there are 10 digits as placeholders for the manipulation. Position 0 defines the manipulation types copy-move, splicing, removal, enhancement ([C,S,R,E]). The following digits 1-9 represent donor patch manipulations according to column $Pos.$ in Table 2. For positions [1,2,7,8] (resample, flip, noise and brightness), a binary value indicates if this manipulation was applied to the donor image patch. In Position 3 (rotate) the values 0-3 indicate if the rotation was executed by 0, 90, 180 or 270 degrees. Position 4 defines if `BoxBlur` (B) or `GaussianBlur` (G) was used. Position 5 specifies the blurring radius. A value of 0 indicates that no blurring was executed. Position 6 indicates which one of the Python-PIL contrast filters `EDGE ENHANCE`, `EDGE ENHANCE MORE`, `SHARPEN`, `UnsharpMask` or `ImageEnhance` (values 1-5) was applied. If none of them was applied, this value is set to 0. Finally, position 9 is set to the JPEG compression factor modulo 10, where a value of 0 indicates that no JPEG compression was applied. The 8 characters `NNNNNNNN` in the image name template stand for a running number of the images.

4 Experimental results

For experimental results and in-depth evaluation, we refer to the publication [Fischinger and Boyer, 2023]. Here, the authors explain how using a simple network trained on the DF2023 dataset has led to state-of-the-art results in the area of image forgery detection.

5 Conclusion

This paper addresses the existing gap in the research area of image forgery detection and localization by providing a comprehensive and publicly accessible training dataset that encompasses a wide array of image manipulation types: We present the Digital Forensics 2023 (DF2023) dataset for training and validation (available from https://zenodo.org/record/7326540), comprised of more than one million images with diverse manipulations. We firmly believe that the availability of this dataset will not only save researchers valuable time but also facilitate easier and more transparent comparisons of network architectures.

Acknowledgement

Co-funded by the European Union

Project 101083573 — GADMO

References

[Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.

[Amerini et al., 2011] Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., and Serra, G. (2011). A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE transactions on information forensics and security*, 6(3):1099–1110.

[Bradski, 2000] Bradski, G. (2000). The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123.

[Carvalho et al., 2013] Carvalho, T., Riess, C., Angelopoulou, E., Pedrini, H., and Rocha, A. R. (2013). Exposing digital image forgeries by illumination color classification. *IEEE Trans. Inf. Forensics and Security*, 8(7):1182–1194.

[Dong et al., 2013] Dong, J., Wang, W., and Tan, T. (2013). Casia image tampering detection evaluation database. In *IEEE China Summit Inter. Conf. Signal Info. Proc.*, pages 422–426. IEEE.

[Fischinger and Boyer, 2023] Fischinger, D. and Boyer, M. (2023). DF-Net: The digital forensics network for image forgery detection. *Irish Machine Vision and Image Processing conference*.

[Guan et al., 2019] Guan, H., Kozak, M., Robertson, E., Lee, Y., Yates, A. N., Delgado, A., Zhou, D., Kheyrkhah, T., Smith, J., and Fiscus, J. (2019). Mfc datasets: Large-scale benchmark datasets for media forensic challenge evaluation. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 63–72. IEEE.

[Hsu and Chang, 2006] Hsu, Y. and Chang, S. (2006). Detecting image splicing using geometry invariants and camera characteristics consistency. In *IEEE Inter. Conf. Multim. Expo*, pages 549–552. IEEE.

[Kwon et al., 2022] Kwon, M.-J., Nam, S.-H., Yu, I.-J., Lee, H.-K., and Kim, C. (2022). Learning jpeg compression artifacts for image manipulation detection and localization. *International Journal of Computer Vision*, 130:1875 – 1895.

[Li et al., 2017] Li, H., Luo, W., and Huang, J. (2017). Localization of diffusion-based inpainting in digital images. *IEEE Transactions on Information Forensics and Security*, 12(12):3050–3064.

[Li et al., 2013] Li, L., Li, S., Zhu, H., Chu, S.-C., Roddick, J., and Pan, J.-S. (2013). An efficient scheme for detecting copy-move forged images by local binary patterns. *Journal of Information Hiding and Multimedia Signal Processing*, 4:46–56.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, L. (2014). Microsoft coco: Common objects in context. In *ECCV*. European Conference on Computer Vision.

[Lyu et al., 2013] Lyu, S., Pan, X., and Zhang, X. (2013). Exposing region splicing forgeries with blind local noise estimation. *International Journal of Computer Vision*, 110:202–221.

[Mahfoudi et al., 2019] Mahfoudi, G., Tajini, B., RETRAINT, F., Morain-Nicolier, F., Dugelay, J.-L., and Pic, M. (2019). Defacto: Image and face manipulation dataset. pages 1–5.

[National Institute of Standards and Technology (NIST), 2016] National Institute of Standards and Technology (NIST) (2016). Nist nimble 2016 datasets. `https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation/`.

[Novozamsky et al., 2020] Novozamsky, A., Mahdian, B., and Saic, S. (2020). Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In *2020 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 71–80.

[Sun et al., 2018] Sun, J.-Y., Kim, S.-W., Lee, S.-W., and Ko, S.-J. (2018). A novel contrast enhancement forensics based on convolutional neural networks. *Signal Processing: Image Communication*, 63:149–160.

[Tralic et al., 2013] Tralic, D., Zupancic, I., Grgic, S., and Grgic, M. (2013). Comofod—new database for copy-move forgery detection. In *Proceedings ELMAR-2013*, pages 49–54. IEEE.

[Verdoliva, 2020] Verdoliva, L. (2020). Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932.

[Wen et al., 2016] Wen, B., Zhu, Y., Subramanian, R., Ng, T.-T., Shen, X., and Winkler, S. (2016). Coverage—a novel database for copy-move forgery detection. In *2016 IEEE international conference on image processing (ICIP)*, pages 161–165. IEEE.

[Wu et al., 2022] Wu, H., Zhou, J., Tian, J., Liu, J., and Qiao, Y. (2022). Robust image forgery detection against transmission over online social networks. *IEEE Transactions on Information Forensics and Security*.

[Wu et al., 2019] Wu, Y., AbdAlmageed, W., and Natarajan, P. (2019). Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9535–9544.

[Zanardelli et al., 2022] Zanardelli, M., Guerrini, F., Leonardi, R., and Adami, N. (2022). Image forgery detection: a survey of recent deep-learning approaches. *Multimedia Tools and Applications*, 82:17521–17566.

[Zhou et al., 2017] Zhou, P., Han, X., Morariu, V. I., and Davis, L. S. (2017). Two-stream neural networks for tampered face detection. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 1831–1839. IEEE.