Improving Robustness of Image Tampering Detection for Compression

Boubacar Diallo, Thierry Urruty, Pascal Bourdon, Christine Fernandez-Maloigne

Improving Robustness of Image Tampering Detection for Compression

Boubacar Diallo, Thierry Urruty, Pascal Bourdon, and Christine Fernandez-Maloigne

XLIM Research Institute (UMR CNRS 7252), University of Poitiers, France
firstname.lastname@univ-poitiers.fr

Abstract. The task of verifying the originality and authenticity of images puts numerous constraints on tampering detection algorithms. Since most images are acquired on the internet, there is a significant probability that they have undergone transformations such as compression, noising, resizing and/or filtering, both before and after the possible alteration. Therefore, it is essential to improve the robustness of tampered image detection algorithms for such manipulations. As compression is the most common type of post-processing, we propose in our work a robust framework against this particular transformation. Our experiments on benchmark datasets show the contribution of our proposal for camera model identification and image tampering detection compared to recent literature approaches.

Keywords: Image forensics, lossy compression, camera model identification, convolutional neural networks

1 Introduction

Nowadays, social networks have become affordable and powerful platforms for sharing, publishing any kind of images. Thus, with the advances of image editing techniques low-cost tampered or manipulated image generation processes have become widely available. Among these tampering techniques, copy-move, splicing and removal are the most common manipulations (see **Fig.1** for example).

- **Copy-move:** It copies and pastes of regions within the same image. This manipulation adds false information or hide information (covering it using other parts of the image).
- **Splicing:** It manipulates images by copying a region from one image and pasting it onto another. It can give the false impression that an additional element was present in a scene at the time that the photograph was captured.
- **Removal:** It eliminates regions from an authentic image followed by an inpainting technique that restores the image by filling holes using characteristics around the hole.

Even with careful inspection, non-expert users will have difficulties to recognize the tampered regions. Such images deliver misleading messages or even dangerous information causing important damage within the society.

Fig. 1. Examples of tampered images that have undergone different tampering techniques. From left to right are the examples showing manipulations of copy-move (*Adding missiles*), splicing (*Fake person*) and removal (*Missing person*).

Therefore, it is of primordial importance to develop forensic methods to validate the integrity of an image. For this reason, over the years, the forensic community has developed several techniques for image authenticity detection and integrity assessment [25, 12]. Among the many investigated forensic issues, great attention has been devoted to camera model identification [17, 18].

Indeed, detecting the model of an image source camera can be crucial for criminal investigations and legal proceedings. This information can be exploited for solving copyright infringement cases, as well as indicating the authors of illicit usages. Each camera model performs peculiar operations on image at acquisition time (e.g. different JPEG compression schemes, proprietary algorithms for Color Filter Array demosaicing, etc.). It leaves on each picture characteristic footprints which are exploited by the proposed approaches. Some authors in [22, 29] have used co-occurrence statistics in different domains coupled to a variety of supervised classification techniques. Most existing techniques use local parametric models of an image or handcraft features to provide sufficient pixel statistics.

Combining forensic methodologies and recent advancements established by deep learning techniques in computer vision, some researchers [29, 7, 2, 1] have proposed to learn camera identification features by using convolutional neural networks (CNN). The advantage of CNN is that they are capable of learning classification features directly from data, hence, they adaptively learn the cumulative traces induced by camera components.

While all of these methods have been very promising, CNN in their current form tend to learn only features related to image content. However, most images can experience unpredictable changes caused by content manipulations or geometric distortions such as lossy compression, noising, resizing and / or filtering, both before and after the possible alteration. It is therefore essential that tam-

pered image detection algorithms take into account the robustness faced with these manipulations. In this paper, motivated by the fact that lossy compression is the most relevant type of image post-processing, we propose a robust framework which contributes in improving camera model identification and image tampering detection. Our experiments will first demonstrate the importance of taking lossy compression into account and then highlight the performance of our proposal.

The remainder of this article is structured as follows: we provide a brief overview of the state-of-the-art of image tampering detection methods using camera model identification in Section 2. Then, we present our general framework for camera model identification and image tampering detection in Section 3. Section 4 presents our exhaustive experiments. It first discusses the importance of lossy compression manipulation before highlighting the robustness of our proposal against such manipulation. Section 5 concludes our work and gives some perspectives of this work.

2 Related work

2.1 Camera model identification

Forensic community researchers have developed "blindly" methods to determine an image camera model by identifying the fingerprints left when taking photographs [25]. Each camera model performs particular operations on image at acquisition pipeline time leaving characteristic fingerprints that can be exploited. These fingerprints are unique from one camera model to another and allow to study about the origin, processing history and authenticity of the captured images.

The first approaches developed used heuristically designed statistical metrics as features to measure and determine camera traces [17]. Then, other techniques use traces of specific physical components such as noise tracks left by camera sensors [28]. Other existing methods rely on the algorithmic components such as the unique implementation of JPEG compression [16] and traces left by demosaicing [26, 10, 11]. Given the difficulty of properly modelling typical operations of the image acquisition pipeline, other camera model identification methods exploit features mainly capturing statistical image properties paired with machine learning classifiers. A technique based on local binary patterns is proposed in [31]. Other researchers [22, 29] exploit the pixel co-occurrence statistics with a variety of supervised classification techniques. These methods guarantee very accurate results, especially on full-resolution images that provide sufficient pixel statistics. All these existing techniques are often designed by local parametric models of image data [26, 10] or use hand-crafted features [23].

However, recent works in research forensics suggest that learning camera features can be accomplished by using convolutional neural networks (CNN) [29, 7, 2, 1]. This was made possible by recent advancements established by deep learning techniques in computer vision [6, 20]. They showed the possibility to

improve the accuracy for detection and classification tasks by training on a great amount of data in order to learn characteristic features directly from the data itself.

2.2 Convolutional Neural Networks

Recent advances in deep learning have led to better performance because of the ability to learn extremely powerful image features with convolutional neural networks (CNN). In the late 1980s, CNN were first proposed by LeCun et al. [21] with the recognition of handwritten letters as an extended version of neural networks (NN). In 2012, with the availability of high-performance computing systems, such as GPUs or large-scale distributed clusters, CNN have become a widely used research tool. Thus, AlexNet [19], GoogLeNet [27] and ResNet [14] for example have become very popular CNN architectures because of impressive accuracy improvements for image classification and localization tasks.

In the last few years, many researchers showed a growing interest in image manipulation detection by applying different computer vision and deep-learning algorithms [29, 23, 7, 8, 5, 1]. In 2016, Bayar et al. [3] used the CNN and developed a new form of convolutional layer that is specifically designed to learn the manipulated features from an image. In this work, CNN are trained to detect multiple manipulations (Median filtering, Gaussian blurring, Additive white Gaussian noise, resizing) applied to a set of unaltered images. In [9], it is shown that both CNN and Long short-term memory (LSTM) based networks are effective in exploiting re-sampling features to detect tampered regions. The robustness against post-processing is not evaluated and it proposes in the future to detect image forgeries. The work in [4] examines the influence of several important CNN design choices for forensic applications, such as the use of a constrained convolutional layer or fixed high-pass filter at the beginning of the CNN. In [7, 8], two techniques are combined for image tampering detection and localisation, leveraging characteristic footprints left on images by different camera models. Firstly, it exploits a convolutional neural network (CNN) to extract characteristic camera model features from image patches. These features are then analysed by means of iterative clustering techniques in order to detect whether an image has been forged, and localise the affected region. Other methods are bound to specific problems as detecting specific tampering cues such as double-JPEG compression [2, 1], re-sampling and contrast enhancement [30]. A deep learning approach to identify facial retouching was also proposed by [24]. Recently, Huh et al. [15] propose a learning algorithm for detecting visual image manipulations that is trained only using a large dataset of real photographs. This model has been applied to the task of detecting and localising image splices.

While all of these methods have been very promising, CNN in their current form tend to learn only features related to image content. However, most images can experience unpredictable changes caused by content manipulations or geometric distortions such as compression, noising, and resizing. So it is essential that the tampered image detection algorithms need to take into account the robustness faced with these manipulations.

3 Proposed Method

In this section, we present our global framework which procures a robust solution for camera model identification and tampering image detection. The motivation of our work comes from the fact that most images are acquired on the internet. Among them, there is a significant proportion that has undergone some transformations such as lossy compression, noising, resizing and / or filtering, both before and after a possible alteration. It is therefore essential that tampered image detection algorithms strong robustness faced with these different manipulations. As compression is the most common and relevant type of post-processing when people share pictures on the internet, our experiments focus on this manipulation.

This work is divided in two parts as shown on Figure 2. In the first one, we detail our deep learning approach to identify camera models. The following part details how it is included in a global framework to obtain robustness against compression for tampering image detection.

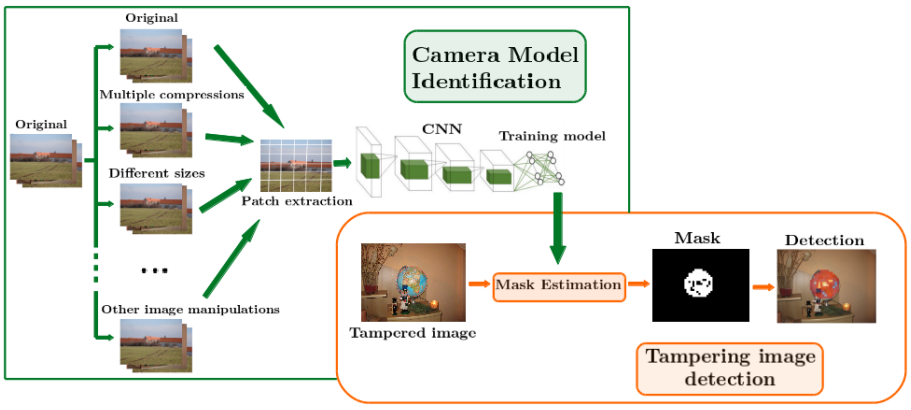


Fig. 2. The pipeline of our framework including the camera model identification learning phase and the tampering image detection method

3.1 Camera model identification

In this part, we will focus on camera model identification which is the main contribution of this paper (left part of Figure 2). The possibility of detecting which camera model has been used to shoot a specific picture is of importance for many forensic tasks as criminal investigations and trials. In the case a deeper source identification (e.g. use of footprints left on images for tampering detection and localization), camera model identification (CMI) can be considered an important preliminary step. The most effective methods for this task are based

on deep learning approaches. They extract distinctive features from the images of interest and use them to train a classifier. This approach requires a dataset of labelled images. In the next subsections, we detail each component of our framework presented on Figure 2.

Image transformations: The first and most important step for a deep learning strategy framework is the quality of the input data with respect to the desired application. As the objective is to detect image tampering on images shared on the internet, the trained CNN model needs to be fed with images that undergo similar transformations as any user could achieve such as lossy compression, noising, resizing and / or filtering. Thus, all original images have to be duplicated with transformed versions of itself. Experiments will show this step is of great importance to obtain good performance.

Patches extraction: As state-of-the-art methods [4, 8] for camera model classification gives promising results with small image patches, we also divide our image in small patches (64×64 pixels) as the second step of our robust framework for camera model identification. Indeed, the use of small image patches instead of full-resolution images better characterizes camera models in a reduced-size space. In order to avoid selecting overly dark or saturated regions, a threshold is used to exclude all patches containing saturated pixels. Each patch inherits the camera model label of its image before feeding the CNN.

Convolutional neural networks for camera model identification: Given its great potential, deep learning has become inevitable for camera model identification. In this section, we exploit convolutional neural networks (CNN) to extract characteristic camera model features from image patches. The first CNN architecture specifically dedicated to camera model identification has been proposed in [7]. In this work, we use a similar network. This choice is motivated with the aim to achieve a high camera model attribution accuracy with a fairly small network architecture. Note that modifying the used CNN is not in the scope of this paper.

The used network contains 11 layers namely 4 convolutional layers, 3 max-pooling layers, 2 fully-connected layers, 1 ReLU layer and 1 Softmax layer. Image patches are fed into the CNN through an input layer, also known as the data layer. The structure of the CNN architecture is described in **Table 1**:

Training: The training architecture is characterized by $340,462$ parameters, learned through Stochastic Gradient Descent on batches of 128 patches. Momentum is fixed to 0.9, weights decay is set to $7.5.10^{-3}$ while the learning rate is initialized to 0.015 and halves every 10 epochs. As trained CNN model M, we select the one that provides the smallest loss on validation patches within the first 50 training epochs.

Table 1. Structure of the CNN architecture [7]. N is the number of training classes

Layer	Input size	Kernel size	Stride	Num. filters	Output size
Conv1	$64 \times 64 \times 3$	4×4	1	32	$63 \times 63 \times 32$
Max-Pool1	$63 \times 63 \times 32$	-	2	-	$32 \times 32 \times 32$
Conv2	$32 \times 32 \times 32$	5×5	1	48	$28 \times 28 \times 48$
Max-Pool2	$28 \times 28 \times 48$	-	2	-	$14 \times 14 \times 48$
Conv3	$14 \times 14 \times 48$	5×5	1	64	$10 \times 10 \times 64$
Max-Pool3	$10 \times 10 \times 64$	-	2	-	$5 \times 5 \times 64$
Conv4	$5 \times 5 \times 64$	5×5	1	128	$1 \times 1 \times 128$
Fully1 (ReLU)	$1 \times 1 \times 128$	-	-	128	128
Fully2 (Softmax)	128	-	-	N	N

Classification: The problem of camera model identification consists in detecting a model L (within a set of known camera models) used to shoot an image I. When a new image I is under analysis, the camera model is estimated as follows: a set of K patches is obtained from image I as described above. The last layer (Softmax) assigns a label to each patch. The predicted model for image I is obtained through majority voting on existing labels.

3.2 Tampering image detection

Here, we briefly present the method for image forgery detection and localization in case of images generated through composition of pictures shot with different camera models. In this scenario, we draw inspiration from [8] by considering that pristine images are pictures directly obtained from a camera. Conversely, forged images are those created by taking patches of a pristine image, and pasting them on images with different camera models. Under these assumptions, the proposed method is devised to estimate whether the totality of image patches comes from a single camera (i.e. the image is pristine), or some portions of the image are not coherent with the rest of the picture in terms of camera attribution (i.e. the image is forged). If this is the case, a localization of the forged region is also done.

The proposed method is described on the right part of Figure 2. A tampered image I is first divided into non-overlapping patches. Each patch P is fed as input to a pretrained CNN to extract a feature vector f of N_{cams} elements corresponding to a number of cameras. This information is given as input to the clustering algorithm that estimates a tampering mask. The final output M is a binary mask, where black parts indicate patches belonging to the pristine region and white ones indicate forged patches. If no (or just a few) forged pixels are detected, the image is considered as pristine.

4 Experiments

In this section, we present our exhaustive experiment results. After detailing the experiment setup including chosen datasets and evaluation criteria, we propose

a preliminary study highlighting the importance of compression as image manipulation technique. Then, we detail the performance of our framework for camera model identification and tampering image detection.

4.1 Experiment Setup

Test datasets: Dresden dataset [13] is a publicly available dataset suitable for image source attribution problems. Dresden contains more than 13,000 images of 18 different camera models. Note that we selected only natural JPEG photos from camera models with more than one instance. This dataset is split into a training, validation, and evaluation sets, denoted DT, DV and DE respectively.

To evaluate tampering detection algorithm, we use the image sets proposed in [8]. These two separate sets of altered data represent a set of "known" data from DE images and an "unknown" dataset which contains images from another 8 camera models not included in the CNN training phase. The objective of using those sets is to study the differences in performance when using "known" and "unknown" cameras. Both sets contain 500 pristine images and 500 tampered images generated following the process given in [8].

Finally, to evaluate the influence of compression, all images from the chosen datasets are compressed with different factor qualities (FQ): 90%, 80% and 70%. The trained CNN with those FQ are named CNN90, CNN80, CNN70 and CNNm respectively for 90%, 80%, 70% and mixed compressed data.

Evaluation criteria: To evaluate the camera model identification performance, we use the average accuracy obtained with a majority voting. We evaluate detection performance on both "known" and "unknown" datasets in terms of accuracy, receiver operating characteristic (ROC) curves and Area Under the ROC Curve (AUC). These statistics are commonly known and used, they identify clearly the difference between the performance of studied approaches.

4.2 Influence of compression on CMI:

In this section we propose our preliminary study that highlights the importance of manipulation process on the CMI accuracy of our framework denoted CNNm compared to the one proposed by Bondi et al. [8].

Table 2. Influence of JPEG compression on Camera Model Identification

Accuracy	***Original***	***QF: 90%***	***QF: 80%***	***QF: 70%***
Bondi et al. [7]	**0.91**	0.19	0.12	0.12
CNNm	0.82	**0.80**	**0.75**	**0.72**

To make this robustness assessment, we consider the original images of the Dresden Test dataset (DT) using a JPEG compression with quality factor values ranging from 70 to 100 with a step of 10. Table 2 shows the influence of the

JPEG compression on the CMI accuracy. As one may observe, the performance of Bondi's approach is superior to ours on Original images. However, their framework decreases dramatically even with a close quality factor ($QF = 90\%$) which is not the case for our proposal.

This result shows us that the CNN trained only on "Original" image for camera model identification is not robust to compression. The reason behind this is that JPEG compression mitigates similar anomalies between block pairs, destroying clues for patch-based approaches such as CNNs. Indeed, it is well-known that JPEG compression is a lossy operation. Because of the rounding errors, it can not only change the original values of pixels, but also leads to information loss.

Figure 3 confirms the fact that a CNN trained model on a specific compression quality factor gives higher accuracy only for this quality factor. However, our framework gives performing results on average of all mixed compressed test images. This results also proves the fact that under a certain quality factor threshold, results will worsen. However, under this threshold, the image quality is too poor to be of any use.

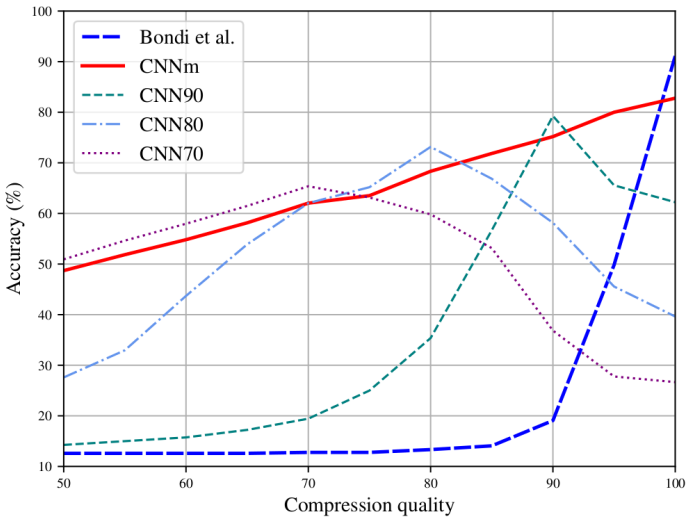


Fig. 3. Accuracy comparison curves of Camera Model Identification

4.3 Image tampering detection

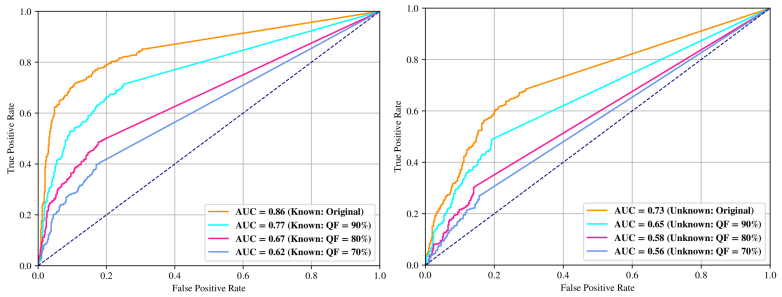
To confirm the first results, we study the influence of the JPEG compression on tampering detection algorithm. Similarly, the "Original" sets are also compressed with quality factors of 90%, 80% and 70%.

Table. 3 shows detection performance on both "known" and "unknown" datasets. Once again, we observe similar observations. Our framework is close to Bondi et al.'s framework for uncompressed ("Original") images. However, our performance outperforms for any other compression quality factor. This result was predictable as the detection is based on the CNN trained for CMI.

Table 3. Tampering detection results with compressed images

Dataset	Compression	Accuracy		TPR	
		Bondi [8]	CNNm	**Bondi** [8]	CNNm
Known	*Original*	**0.84**	0.77	**0.9**	0.83
	90%	0.56	**0.72**	0.47	**0.68**
	80%	0.52	**0.65**	0.24	**0.48**
	70%	0.52	**0.61**	0.15	**0.38**
Unknown	*Original*	**0.79**	0.7	**0.84**	0.68
	90%	0.56	**0.63**	0.38	**0.46**
	80%	0.52	**0.57**	0.17	**0.30**
	70%	0.51	**0.56**	0.11	**0.26**

The ROC and AUC values presented Figure 4 help us to study the effect of compression quality factor values on our framework only. Those figures that for tampering detection also, the loss of accuracy is closely linked to the quality of an image.



(a) "known" dataset (b) "unknown" dataset

Fig. 4. ROC curves of tampering detection algorithm tested on (a) "known" and (b) "unknown" datasets with different compression values (90% and 80%)

5 Conclusion

In this paper, we propose a deep learning framework robust for camera identification model and tampering image detection. It includes any kind of manipulations

that are commonly used by any user sharing images. We test our framework on the compression quality factor manipulation and show that our approach globally outperforms existing literature approaches. Our study emphasizes the fact that discriminant features from compressed images are harder to retrieve. Our future work will take this aspect into account to guarantee similar performance or very compressed data. We will also investigate neural network activation nodes to better understand the artifacts that help identifying camera models.

References

1. I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli. Localization of jpeg double compression through multi-domain convolutional neural networks. In *Proc. of IEEE CVPR Workshop on Media Forensics*, volume 3, 2017.
2. 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*, 49:153–163, 2017.
3. B. Bayar and M. 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*, pages 5–10. ACM, 2016.
4. B. Bayar and M. C. Stamm. Design principles of convolutional neural networks for multimedia forensics. *Electronic Imaging*, 2017(7):77–86, 2017.
5. B. Bayar and M. C. Stamm. Towards open set camera model identification using a deep learning framework. In *The 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE*, 2018.
6. Y. Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.
7. 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 Letters*, 24(3):259–263, 2017.
8. L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro. Tampering detection and localization through clustering of camera-based cnn features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1855–1864, 2017.
9. J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson. Detection and localization of image forgeries using resampling features and deep learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1881–1889. IEEE, 2017.
10. H. Cao and A. C. Kot. Accurate detection of demosaicing regularity for digital image forensics. *IEEE Transactions on Information Forensics and Security*, 4(4):899–910, 2009.
11. C. Chen, X. Zhao, and M. C. Stamm. Detecting anti-forensic attacks on demosaicing-based camera model identification. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 1512–1516. IEEE, 2017.
12. H. Farid. *Photo forensics*. MIT Press, 2016.

13. T. Gloe and R. Böhme. The'dresden image database'for benchmarking digital image forensics. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1584–1590. ACM, 2010.
14. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
15. M. Huh, A. Liu, A. Owens, and A. A. Efros. Fighting fake news: Image splice detection via learned self-consistency. *arXiv preprint arXiv:1805.04096*, 2018.
16. E. Kee, M. K. Johnson, and H. Farid. Digital image authentication from jpeg headers. *IEEE Trans. Information Forensics and Security*, 6(3-2):1066–1075, 2011.
17. M. Kharrazi, H. T. Sencar, and N. Memon. Blind source camera identification. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 1, pages 709–712. IEEE, 2004.
18. M. Kirchner and T. Gloe. Forensic camera model identification. *Handbook of Digital Forensics of Multimedia Data and Devices*, pages 329–374, 2015.
19. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
20. Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.
21. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
22. F. Marra, G. Poggi, C. Sansone, and L. Verdoliva. Evaluation of residual-based local features for camera model identification. In *International Conference on Image Analysis and Processing*, pages 11–18. Springer, 2015.
23. F. Marra, G. Poggi, C. Sansone, and L. Verdoliva. A study of co-occurrence based local features for camera model identification. *Multimedia Tools and Applications*, 76(4):4765–4781, 2017.
24. A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018.
25. M. C. Stamm, M. Wu, and K. R. Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013.
26. A. Swaminathan, M. Wu, and K. R. Liu. Nonintrusive component forensics of visual sensors using output images. *IEEE Transactions on Information Forensics and Security*, 2(1):91–106, 2007.
27. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
28. T. H. Thai, R. Cogranne, and F. Retraint. Camera model identification based on the heteroscedastic noise model. *IEEE Transactions on Image Processing*, 23(1):250–263, 2014.
29. A. Tuama, F. Comby, and M. Chaumont. Camera model identification with the use of deep convolutional neural networks. In *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*, pages 1–6. IEEE, 2016.
30. L. Wen, H. Qi, and S. Lyu. Contrast enhancement estimation for digital image forensics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(2):49, 2018.
31. G. Xu and Y. Q. Shi. Camera model identification using local binary patterns. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 392–397. IEEE, 2012.