Training CNNs in Presence of JPEG Compression: Multimedia Forensics vs Computer Vision

Sara Mandelli, Nicolò Bonettini, Paolo Bestagini, Stefano Tubaro
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

***Abstract*—Convolutional Neural Networks (CNNs) have proved very accurate in multiple computer vision image classification tasks that required visual inspection in the past (e.g., object recognition, face detection, etc.). Motivated by these astonishing results, researchers have also started using CNNs to cope with image forensic problems (e.g., camera model identification, tampering detection, etc.). However, in computer vision, image classification methods typically rely on visual cues easily detectable by human eyes. Conversely, forensic solutions rely on almost invisible traces that are often very subtle and lie in the fine details of the image under analysis. For this reason, training a CNN to solve a forensic task requires some special care, as common processing operations (e.g., resampling, compression, etc.) can strongly hinder forensic traces. In this work, we focus on the effect that JPEG has on CNN training considering different computer vision and forensic image classification problems. Specifically, we consider the issues that rise from JPEG compression and misalignment of the JPEG grid. We show that it is necessary to consider these effects when generating a training dataset in order to properly train a forensic detector not losing generalization capability, whereas it is almost possible to ignore these effects for computer vision tasks.**

I. Introduction

Thanks to the increasing availability of digital data and computational resources, CNNs have greatly outperformed multiple classical approaches for a wide variety of tasks in different fields [1], [2]. For instance, they have been used with outstanding results to solve several computer vision problems related to image analysis and understanding. This is the case of object detection [3], image segmentation [4], image retrieval [5] and many other tasks. All these tasks are characterized by two properties: (i) they are extremely challenging to solve using a model-based solution (i.e., it is hard to define objective properties that an image must have in order to belong to a given class); (ii) they can be reasonably well solved by human operators through visual inspection (i.e., a human can classify or segment an object in a scene using visual semantic cues, given that the quality of the image under analysis is decent). In other words, we often know what we are looking for, but we cannot describe it easily.

The rise of CNN-based solutions over classical methods is not just a computer vision prerogative. In the last few years, also the multimedia forensics community has started replacing more classical detectors with CNN-based ones. Camera model identification [6], [7], image tampering localization [8], [9] and deepfake detection [10], [11] are just a few examples of problems whose accurate solutions are nowadays based on the use of CNNs. However, despite forensic and computer vision tasks share some similarities, they also have some differences. On one hand, forensic problems are often hard to solve through purely model-based methods as computer vision tasks. As an example, considering an image tampering detection problem, defining a complete model accommodating for all possible image forgery operations is far from being practical. On the other hand, forensic tasks cannot be typically solved using visual cues by human operators. As an example, it is almost impossible to tell how many compression steps an image underwent or which is the specific camera model used for the shooting, even if the image quality is extremely high.

The impossibility of easily extracting forensic information by simple visual inspection is due to the fact that forensic traces are often hidden in tiny details of the image under analysis. For instance, forensic information can lie in high-pass components, or in low-power and noisy-like signals hardly detected by human eyes. For this reason, training a CNN for a computer vision task or for a forensic one can be strongly different. As an example, we expect that typical image processing and enhancement operations do not impact on object detection tasks (unless operated as an adversarial attack [12]). In other words, the class of an image should not change even if the image is resized, compressed, undergone some color correction or other operations, as far as a human operator can still recognize the object. Conversely, it is hard to tell whether processing operations impact on forensic traces of any kind, as we do not always clearly understand which are the important cues that a CNN captures.

In this paper, we specifically focus on problems introduced by JPEG compression while training a CNN for forensic tasks. Indeed, JPEG is one of the most widely used image compression standard and is well-known to hinder forensic traces. We analyze the effect of working with JPEG compressed images which have been cropped with respect to their original size and the consequence of applying JPEG compression with different quality factors during training and/or testing of a CNN. We compare results achieved on two forensic tasks (i.e., camera model identification and detection of synthetically generated images) with those achieved on two computer vision tasks (i.e., image classification on ImageNet [13] and Lsun [14] datasets). For each task, we use four different CNNs in order to avoid possible biases due to some specific architectures.

WIFS'2020, December, 6-11, 2020, New York, USA.
978-1-7281-9930-6/20/$31.00

Results confirm that training a CNN for a forensic task needs some special care that is not necessary for computer vision tasks.

We hope this analysis can be helpful to practitioners and researchers willing to use powerful data-driven approaches in forensic scenarios, as they might risk getting tricked in learning something not related to the task under analysis. We also hope our experiments will motivate some more research in this area.

II. Background

A. JPEG Compression Forensics

JPEG is one of the most widely used compression scheme on the web, and it approximately works as follows. Input images are partitioned into 8×8 non-overlapped pixel blocks. 2D Discrete Cosine Transform (DCT) is computed for each block and transform coefficients are quantized into integer-valued levels according to the selected quantization matrix and quality factor. Quantization is the step causing information loss. A lower quantization quality factor indicates a stronger quantization, thus lower quality of the final decompressed image. Quantized values are then converted into a binary stream by means of lossless coding. At decoding time, the binary stream is decompressed, coded blocks are reconstructed by applying inverse 2D DCT on quantized coefficients and the image is re-built in the pixel domain.

JPEG compression leaves peculiar traces which have been studied in the forensics literature for many years. For instance, the 8×8 block processing leaves characteristic blocking artifacts that can be exploited to tell whether an image underwent JPEG compression [15]. It is also well known that quantization affects DCT coefficients' histograms, influencing in peculiar way their shape depending on the number of compressions an image has undergone. This can be exploited to detect aligned and non-aligned JPEG compressions [16], [17], [18]. Alternatively, the authors of [19], [20] propose methods based on the first digits' law to detect multiple compressions. The same task is accomplished in [21] by means of non-negative matrix factorization, which is particularly useful given the non-negative nature of histograms. Primary quantization step in double compression is estimated by statistical model of DCT in [22]. Since the JPEG compression pipeline has several degrees of freedom in its definition (e.g., the quantization rule, the quantization matrix corresponding to a certain quality factor, etc.), it is also possible to distinguish among different implementations (i.e., different software used for the compression) as shown in [23], [24].

B. CNNs in Multimedia Forensics

In the last few years, following a common trend in many fields, CNNs have outperformed many classical forensics detectors. Camera model identification is considered in [7], [6], [25], where deep features are extracted from images by means of CNNs and then fed to model based algorithms to perform the classification task. In [26], [27] authors show how it is possible to train a CNN to detect double JPEG compression, easing its work by pre-computing DCT coefficients. In [28] authors makes use of a CNN approach for detecting contrast adjustment on images in presence of JPEG compression. CNNs have successfully outperformed classical detectors also for the source device identification problem. In [29], a CNN-based image denoising is proposed instead of classical denoising procedures for boosting device attribution performance, while in [30] the CNN is forced to learn a way of comparing camera fingerprint and image noise at patch level. Given the flourish of generation techniques such as Generative Adversarial Networks (GANs) and DeepFakes [31], the forensics community has recently focused on spotting media generated in such manner. In [32] authors show how different GANs could leave different traces in generated images, giving the investigator a tool to detect them. The problem of incremental learning on new kind of GANs is considered in [33]. A method for detecting DeepFake videos using Recurrent Neural Network is proposed in [10], while in [11] an ensemble of networks with an attention mechanism is employed for the same task. In this vein, a study on different preprocessing and augmentations techniques (including JPEG compression) when dealing with detection of CNN generated images has been proposed in [34].

III. Case Studies

In order to investigate the impact of JPEG when training CNNs for multimedia forensics and computer vision tasks, we consider two realistic scenarios as case studies: (i) the dataset under analysis contains some JPEG images cropped with respect to their original size; (ii) the dataset contains some JPEG images that have been compressed with unknown quality factor. It follows an exhaustive description of each scenario.

A. JPEG Grid Misalignment

While editing a photograph or simply uploading a profile picture over social networks, it often occurs that images are cropped with respect to their original size. As a matter of fact, this operation is performed most of the times without paying attention to the precise pixel coordinates of the cropped area, as it is more important to prevent the picture subjects being canceled by the cropping. As a consequence, it generally happens that JPEG-compressed images are cropped without respecting the 8×8 characteristic pixel grid introduced by JPEG compression. If the image is then further saved as JPEG, a new 8×8 grid non-aligned with the original one is generated.

When training a CNN to solve an image classification problem, the presence of JPEG grid misalignment on images can cause issues depending on the specific task. In Section IV, we perform some experiments showing when JPEG misalignment can be problematic considering both image forensics and computer vision tasks.

B. Quality factor of JPEG Compression

When collecting images "in the wild", the fact that images may come from different cameras, may be post-processed or uploaded on social media multiple times has to be taken

for granted. Likewise, it is well known that each single camera model, image editing software or social media may compress images using different JPEG implementations and/or parameters (e.g., quantization matrix, quality factors, etc.). As a result, we can state that a wide variety of differently compressed JPEG images can be found on the internet.

Compressing images with different quality factors surely leaves peculiar forensic traces. For this reason, we aim at investigating how the quality factor of JPEG compressed images affects a CNN performance. In Section IV, we analyze computer vision and image forensics tasks, eventually showing how the training phase can be tuned in order to improve results in case different JPEG compressions are used.

IV. Experimental Setup

A. Datasets

In order to provide a sufficiently general idea of the impact of JPEG compression on multimedia forensics and computer vision problems, we consider two tasks per area.

Multimedia Forensics. Regarding multimedia forensics, we tackle two problems: (i) camera model identification, i.e., identifying the source camera model of a query image; (ii) detection of CNN-generated images from original photographs. If the first problem is common in forensics investigation, the latter is taking the trend in the last few years, due to the widespread diffusion of images and videos with fake content produced by means of CNN-based technologies. To investigate these problems, we consider well known datasets available in the literature. For camera model identification task, we exploit images selected from the Vision dataset [35], considering the 28 available camera models. For each camera model, we pick all the natural images (more than 150 images per model), extracting 10 random patches per image with a common size of 512×512 pixels. Concerning the task of CNN-generated images, we select images from the dataset released in [32], which provides pairs of generated/pristine images for multiple CNN-generated image categories (e.g., apple2orange, summer2winter, etc.). Specifically, we select images generated from CycleGAN [36] (~ 1000 images per category), extracting one patch per image and cropping to 224×224 pixels.

Computer Vision. As extremely common computer vision task, we considered image object recognition, i.e., learning to classify images according to the specific subject depicted. We investigate image classification performances of CNNs over two datasets. Images of the first dataset are selected from 16 different synsets of the well known ImageNet database [13]. The second dataset is extracted from Lsun dataset [14], where images from 20 different categories are selected. In both cases, we extract one patch per image, cropping to a common resolution of 224×224 pixels. In detail, we pick around 1000 and 2000 images per class for ImageNet and Lsun, respectively.

B. Network training and evaluation

We investigate four different networks. Two networks are selected from the recently proposed EfficientNet family of models [37], which achieves very good results both in computer vision and multimedia forensics tasks. Specifically, we select EfficientNetB0 and EfficientNetB4 models. The other networks are known in literature as ResNet50 [38] and XceptionNet [39].

Following a common procedure in CNN training, we initialize the network weights using those trained on ImageNet database. All CNNs are trained using cross-entropy loss and Adam optimizer with default parameters. The learning rate is initialized to 0.001 and is decreased by a factor 10 whenever the loss does not decrease for 10 epochs. Training is stopped if loss does not decrease for more than 20 epochs, and the model providing the best validation loss is selected.

Concerning the dataset split policy, we always keep 80% of the images for training phase (further divided in $85\% - 15\%$ for training and validation sets, respectively), leaving the remaining 20% to the evaluation set. All the experiments are performed on the evaluation set in closed-set scenario, i.e., given a query image, the correct image category is always present in the given set of possible answers. After the network prediction, the category returning the highest CNN score is associated with the image. We use the average accuracy of correct predictions per category as evaluation metrics.

All tests have been run on a workstation equipped with one Intel® Xeon Gold 6246 (48 Cores @3.30 GHz), RAM 252 GB, one TITAN RTX (4608 CUDA Cores @1350 MHz), 24 GB, running Ubuntu 18.04.2. We resort to Albumentation [40] as data augmentation library for compressing images in JPEG, while we use Pytorch [41] as Deep Learning framework. The experimental code is available at https://github.com/polimi-ispl/wifs2020-forensics-vs-cv.

C. Experiments

The four tasks and four networks have been tested considering the impact of two different JPEG compression aspects.

JPEG-grid misalignment. In this experiment, for both training and test, we consider the combinations of two possible scenarios: (i) images are always aligned to the 8×8 JPEG grid that starts from the top-left pixel; (ii) images are non-aligned with the JPEG grid, which may start from any pixel position. To simulate this effect, we JPEG compress all images using a quality factor of 100. This guarantees that images contain the typical JPEG lattice, not worsening their visual quality in any way. To obtain image patches coherently aligned to the JPEG lattice, we select patches by cropping the images in a way that patches are aligned to the 8×8 pixel grid. To obtain patches not aligned with the grid, we extract them in random positions.

Quality factor of JPEG compression.

In this experiment, we consider that evaluation images may be compressed with different quality factors $\mathrm{QF} \in \{50, 60, 70, 80, 90, 99\}$. We perform all possible combinations

of training and testing under different hypothesis on the used quality factor. Specifically, we train the CNNs in two ways: (i) using image patches directly selected from the original datasets, not applying any kind of data augmentation; (ii) performing data augmentation in training phase, including in the training dataset half of the images compressed with a quality factor selected from the above reported list. In the first training situation, we suppose to know nothing about the JPEG compression parameters of the evaluation images. In the second scenario, we assume some knowledge on the JPEG quality factor and exploit this to potentially improve final results. In this setup, patches are extracted in random positions, being the JPEG-grid alignment a nuisance parameters for the evaluation of accuracy versus JPEG quality factor.

V. Results

JPEG-grid misalignment. Fig. 1 reports JPEG grid misalignment results for the problems of camera model identification (a), CNN-generated image detection (b) and image classification on the extracted subsets of ImageNet (c) and Lsun (d) databases.

Considering multimedia forensics tasks, we notice that being careful to the JPEG grid alignment of the extracted patches is paramount for achieving good accuracy. When we train a detector on JPEG-aligned patches and we test it on JPEG-misaligned ones, results drop consistently. This is especially true for the camera model identification problem (Fig. 1a), where all CNNs report accuracies always higher than 0.88, except for the case of training on JPEG-aligned images and testing on random cropped ones. In this scenario, none of the proposed CNN architectures is able to overcome 0.71 as average accuracy. For the task of CNN-generated image detection (Fig. 1b), average accuracy worsens as well only in this particular situation.

On the contrary, if we consider the computer vision tasks (Fig. 1c and Fig. 1d), training and/or testing on patches aligned or not with the JPEG grid does not change the achieved results in a systematic way. Results are almost uniform for all the networks. This probably means that CNNs are really capturing some information related to visual cues which are not hindered by a simple JPEG-grid misalignment.

Quality factor of JPEG compression. Figs. 2, 3, 4 and 5 depict results related to investigations on the JPEG quality factor for EfficientNetB0, EfficientNetB4, ResNet50 and XceptionNet, respectively.

In these experiments as well, notice that multimedia forensics tasks suffer from JPEG quality factor mismatching much more than computer vision ones. For instance, looking at the camera model identification goal, not performing augmentation in training phase can strongly hinder CNN performances: by training on augmented data with $\mathrm{QF} = 50$, evaluation accuracy can pass from less than 0.2 to more than 0.85. Regarding all the multimedia forensics tasks, selecting a low quality factor for training augmentation seems to allow a better coverage in testing phase with respect to high quality

Fig. 1. Accuracy as a function of JPEG-grid alignment in training and/or testing phases, for the tasks of: (a) camera model identification; (b) CNN-generated image detection; (c) image classification on ImageNet subset; (d) image classification on Lsun subset.

factors. Indeed, training on data augmented with $\mathrm{QF} = 99$ almost corresponds to absence of augmentation and achieves acceptable results only when the test QF matches the training one. On the contrary, as training QF decreases, evaluation results present a flat behavior for all the possible test QF.

This phenomenon does not occur or is extremely reduced in computer vision problems, where the dynamic range of output accuracy is much more limited around the same average value in all the experiments. Again, this probably means that computer vision goals are not influenced by training/testing on specific JPEG configurations, and results maintain accurate whenever the image quality is preserved and objects remain detectable. Conversely, in multimedia forensics, image visual quality is not the only thing to take care of.

VI. Conclusions

In this work, we studied the effect of JPEG compression on CNNs applied to multimedia forensic tasks. In particular, we considered the effect of training and testing CNNs considering different JPEG-grid alignment and JPEG quality factors. We compared the achieved results to those obtained in the same conditions on two computer vision tasks.

Results show that CNNs are extremely delicate in the multimedia forensic scenario. If we train a CNN only considering uncompressed images, it fails when applied to compressed ones. If we train a CNN only considering a specific JPEG-grid alignment, it will fail on randomly cropped images. Conversely, computer vision tasks involving image analysis and understanding are inherently more robust to JPEG compression, given that image visual quality remains decent.

In the light of these results, we will pay particular attention whenever we train a CNN or another sophisticated data-driven

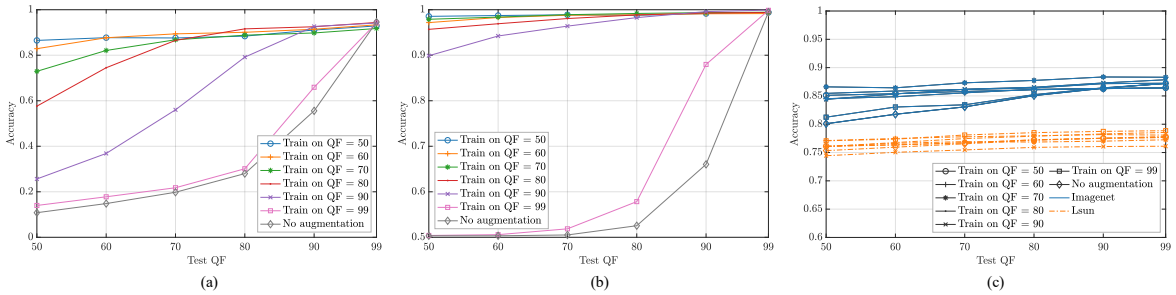

Fig. 2. Accuracy of EfficientNetB0 as a function of training augmentation for the tasks of (a) camera model identification, (b) CNN-generated image detection, (c) image classification on ImageNet and Lsun subsets.

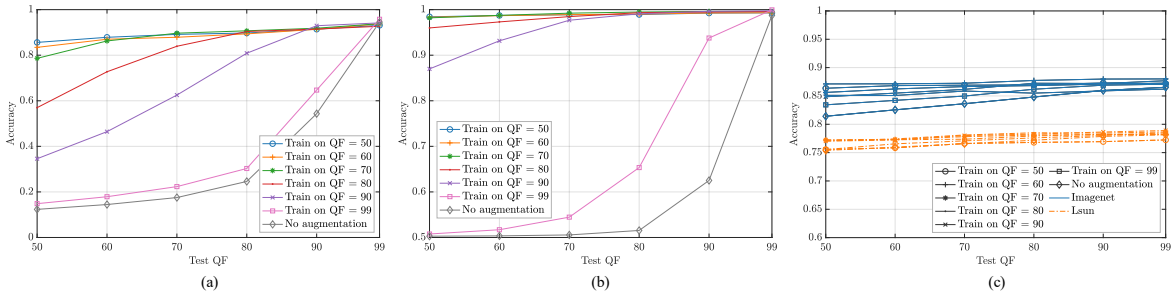

Fig. 3. Accuracy of EfficientNetB4 as a function of training augmentation for the tasks of (a) camera model identification, (b) CNN-generated image detection, (c) image classification on ImageNet and Lsun subsets.

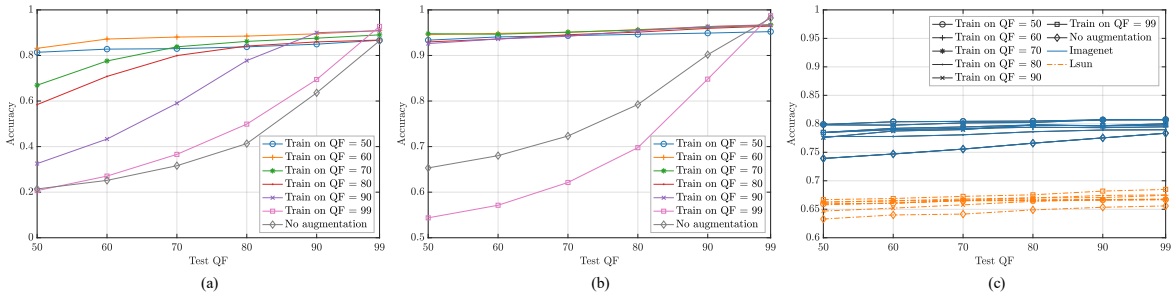

Fig. 4. Accuracy of ResNet50 as a function of training augmentation for the tasks of (a) camera model identification, (b) CNN-generated image detection, (c) image classification on ImageNet and Lsun subsets.

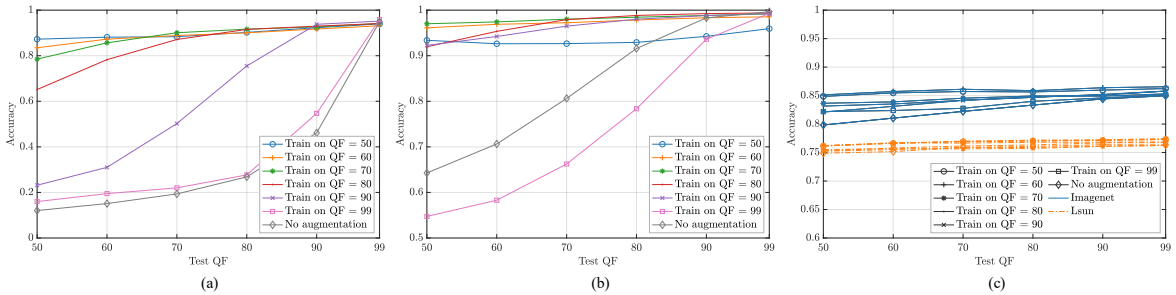

Fig. 5. Accuracy of XceptionNet as a function of training augmentation for the tasks of (a) camera model identification, (b) CNN-generated image detection, (c) image classification on ImageNet and Lsun subsets.

method for forensic purpose, as we want to avoid getting biased by some specific global processing traces.

ACKNOWLEDGMENT

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement num-

ber FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government. Hardware support was generously provided by NVIDIA Corporation.

REFERENCES

[1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[2] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais *et al.*, "Deep learning and process understanding for data-driven Earth system science," *Nature*, vol. 566, pp. 195–204, 2019.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[5] J. Yue-Hei Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 53–61.

[6] 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 (WIFS)*, 2016.

[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*, vol. 24, pp. 259–263, 2016.

[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 *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1855–1864.

[9] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-Based Camera Model Fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2020.

[10] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018.

[11] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video Face Manipulation Detection Through Ensemble of CNNs," *arXiv preprint arXiv:2004.07676*, 2020.

[12] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *arXiv preprint arXiv:1702.02284*, 2017.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[14] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop," *arXiv preprint arXiv:1506.03365*, 2015.

[15] Z. Fan and R. L. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Transactions on Image Processing (TIP)*, vol. 12, pp. 230–235, 2003.

[16] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE transactions on information forensics and security*, vol. 4, pp. 154–160, 2009.

[17] T. Bianchi and A. Piva, "Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 7, pp. 842–848, 2012.

[18] M. Barni, A. Costanzo, and L. Sabatini, "Identification of cut paste tampering by means of double-JPEG detection and image segmentation," in *IEEE International Symposium on Circuits and Systems*, 2010.

[19] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple JPEG compressions using first digit features," *APSIPA Transactions on Signal and Information Processing*, vol. 3, p. e19, 2014.

[20] C. Pasquini, G. Boato, and F. Perez-Gonzalez, "Multiple JPEG compression detection by means of Benford-Fourier coefficients," in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014, pp. 113–118.

[21] S. Mandelli, N. Bonettini, P. Bestagini, V. Lipari, and S. Tubaro, "Multiple Jpeg Compression Detection Through Task-Driven Non-Negative Matrix Factorization," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2106–2110.

[22] T. H. Thai and R. Cogranne, "Estimation of Primary Quantization Steps in Double-Compressed JPEG Images Using a Statistical Model of Discrete Cosine Transform," *IEEE Access*, vol. 7, pp. 76 203–76 216, 2019.

[23] S. Agarwal and H. Farid, "Photo forensics from JPEG dimples," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE, 2017, pp. 1–6.

[24] N. Bonettini, L. Bondi, P. Bestagini, and S. Tubaro, "JPEG Implementation Forensics Based on Eigen-Algorithms," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018, pp. 1–7.

[25] B. Bayar and M. C. Stamm, "Towards Open Set Camera Model Identification Using a Deep Learning Framework," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2007–2011.

[26] 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.

[27] W. Ahn, S. . Nam, M. Son, H. . Lee, and S. Choi, "End-to-end double JPEG detection with a 3D convolutional network in the DCT domain," *Electronics Letters*, vol. 56, no. 2, pp. 82–85, 2020.

[28] M. Barni, A. Costanzo, E. Nowroozi, and B. Tondi, "Cnn-Based Detection of Generic Contrast Adjustment with Jpeg Post-Processing," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 3803–3807.

[29] M. Kirchner and C. Johnson, "SPN-CNN: Boosting Sensor-Based Source Camera Attribution With Deep Learning," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2019.

[30] S. Mandelli, D. Cozzolino, P. Bestagini, L. Verdoliva, and S. Tubaro, "CNN-Based Fast Source Device Identification," *IEEE Signal Processing Letters*, vol. 27, pp. 1285–1289, 2020.

[31] L. Verdoliva, "Media Forensics and DeepFakes: an overview," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2020.

[32] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Do GANs Leave Artificial Fingerprints?" in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2019, pp. 506–511.

[33] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, "Incremental learning for the detection and classification of GAN-generated images," in *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2019, pp. 1–6.

[34] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 7, 2020.

[35] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "VISION: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 15, 2017.

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[37] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[38] 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*, 2016, pp. 770–778.

[39] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[40] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.