Deep Learning Methods and Advancements in Digital Image Forensics

Thèse de doctorat d'Informatique, Télécommunications et Électronique de Sorbonne Université, préparée à Eurecom.

École doctorale n° 130, Sciences et technologies de l'information et de la communication.
Spécialité de doctorat: Sécurité Numérique
Unité de recherche: Eurecom
Référent: : Jean-Luc Dugelay

Thèse présentée et soutenue à Biot, le 26 Septembre 2022, par

Alexandre Berthet

Composition du jury:

CHAUMONT Marc Maître de conférence - université de Nîmes, LIRMM	Rapporteur
FONTAINE Caroline Directrice de recherche, CNRS	Présidente
PIC Marc Directeur technique, Surys	Examinateur
RETRAINT Florent Professeur des universités, UTT	Rapporteur
SAHBI Hichem Chargé de recherche, CNRS	Examinateur
DUGELAY Jean-Luc Professeur des universités, Eurecom	Directeur
GALDI Chiara Chercheuse associée, Eurecom	Co-encadrante

Thèse de doctorat

NNT:

SORBONNE UNIVERSITÉ

ÉCOLE DOCTORALE
Sciences et technologies
de l'information et de la
communication

Titre : Méthodes d'apprentissage profond et avancements pour la criminalistique des images.

Mots-clés : *Criminalistique des Images, Reconnaissance de Caméras, Compression basée sur l'Intelligence Artificielle, Protocoles d'Évaluation, Contre-Criminalistique.*

Résumé : Le volume de données visuelles numériques augmente considérablement d'année en années. En parallèle, l'édition d'images est devenue plus facile et plus précise. Les modifications malveillantes sont donc plus accessibles. La criminalistique des images fournit des solutions pour garantir l'authenticité des données visuelles numériques. La reconnaissance de la caméra source et la détection des images falsifiées sont notamment les tâches principales. Tout d'abord, les solutions étaient des méthodes classiques basées sur les artéfacts produits lors de la création d'une image numérique. Puis, comme pour d'autres domaines du traitement d'images, les méthodes sont passées à l'apprentissage profond. Dans un premier temps, nous présentons une étude de l'état de l'art des méthodes d'apprentissage profond pour la criminalistique des images. Notre étude de l'état de l'art souligne le besoin d'appliquer des modules de pré-traitement pour extraire les artéfacts cachés par le contenu des images. Nous avons aussi mis en avant les problèmes concernant les protocoles d'évaluation de la reconnaissance d'image. De plus, nous abordons la contre-criminalistique et présentons la compression basée sur l'intelligence artificielle, qui pourrait être pris en compte comme une attaque. Dans un second temps, cette thèse détaille trois protocoles d'évaluation progressifs qui abordent les problèmes de reconnaissance de caméras. Le protocole final, plus fiable et reproductible, met en avant l'impossibilité des méthodes de l'état de l'art à reconnaître des caméras dans un contexte difficile. Dans un troisième temps, nous étudions l'impact de la compression basée sur l'intelligence artificielle sur deux tâches analysant les artéfacts de compression : la détection de falsifications et la reconnaissance du réseau social. Les performances obtenues montrent d'une part que cette compression doit-être prise en compte comme une attaque, mais qu'elle mène à une baisse plus importante que d'autres manipulations pour une dégradation d'image équivalente.

Eurecom, centre de recherche en sciences numérique
450 Route des Chappes, 06410 Biot, France

EURECOM
Sophia Antipolis

SORBONNE UNIVERSITÉ

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication

Title: Deep Learning Methods and Advancements in Digital Image Forensics.

Keywords: Digital Image Forensics, Camera Recognition, AI-based Compression, Evaluation Protocols, Counter-Forensics.

Abstract: The volume of digital visual data is increasing dramatically year after year. At the same time, image editing has become easier and more precise. Malicious modifications are therefore more accessible. Image forensics provides solutions to ensure the authenticity of digital visual data. Recognition of the source camera and detection of falsified images are among the main tasks. At first, the solutions were classical methods based on the artifacts produced during the creation of a digital image. Then, as in other areas of image processing, the methods moved to deep learning. First, we present a state-of-the-art survey of deep learning methods for image forensics. Our state-of-the-art survey highlights the need to apply pre-processing modules to extract artifacts hidden by image content. We also highlight the problems concerning image recognition evaluation protocols. Furthermore, we address counter-forensics and present compression based on artificial intelligence, which could be considered as an attack. In a second step, this thesis details three progressive evaluation protocols that address camera recognition problems. The final protocol, which is more reliable and reproducible, highlights the impossibility of state-of-the-art methods to recognize cameras in a challenging context. In a third step, we study the impact of compression based on artificial intelligence on two tasks analyzing compression artifacts: tamper detection and social network recognition. The performances obtained show on the one hand that this compression must be taken into account as an attack, but that it leads to a more important decrease than other manipulations for an equivalent image degradation. Future perspectives could be: i) the use of loss functions dedicated to pair similarity, such as contrastive of triplet; ii) the creation of a database including AI-based compression for the detection of double compression.

Eurecom, centre de recherche en sciences numérique
450 Route des Chappes, 06410 Biot, France

EURECOM
Sophia Antipolis

Dédicace

Je tiens à dédicacer mes parents, source inépuisable de soutien et d'amour. Sans eux, l'idée d'une thèse ne me serait pas venue et ils sont donc les premiers que je remercie et à qui je la dédie.

Je tiens aussi à dédier cette thèse à ma compagne, qui m'a accompagné au quotidien dans cette épreuve et a su me pousser dans les moments les plus durs.

Avant-Propos

Pour faciliter une lecture rapide de la thèse, un résumé de chaque sous-section est fourni. Ils sont facilement reconnaissables dans un encadré bleu.

Remerciements

Je tiens à remercier mon directeur de thèse, Jean-Luc Dugelay, qui m'a suivi pendant ces trois années. Il a été force de proposition, d'inspiration et de travail.

Je remercie aussi ma co-encadrante, Chiara Galdi, qui m'a accompagné de manière plus quotidienne et technique durant la moitié de ma thèse. Elle m'a permis de pousser à bout certaines recherches et d'en finir d'autres.

Je voudrais aussi remercier mes collègues doctorants qui ont rendu ce séjour plus joyeux et festifs. Ces moments passés avec eux ont été très appréciables et m'ont permis de m'aérer l'esprit en temps voulu.

Acronyms

A-DJPEG-C Aligned-Double JPEG-Compression. 91, 93

ACF Anti-Counter-Forensics. 52, 53, 54, 56, 57

AI Artificial Intelligence. 2, 11, 18, 46, 57, 58, 91, 92, 94, 95, 96, 97, 98, 99, 116, 117, 118, 119, 120, 122, 123, 127, 128, 130

AP Average Precision. 14, 96, 97

AuC Area under the Curve. 14, 15, 48, 50, 86, 87

AWGN Additive White Gaussian Noise. 15, 16, 42, 51, 52, 53, 95, 96, 97, 98, 119, 120, 121, 123, 127, 128

BMP BitMaP. 42, 43, 44

CE Contrast Enhancement. 51, 54, 55, 57

CF Counter-Forensics. 39, 52, 53, 54, 55, 57, 58, 91

CFA Color Filter Array. 15, 19, 56, 59

CNN Convolutional Neural Network. 13, 18, 22, 28, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 55, 56, 60, 61, 62, 63, 66, 71, 83, 93, 94, 100, 101, 103, 104, 106, 107, 110, 117, 118, 119

CORESA COmpression et Representation des Signaux Audiovisuels. 130

CW CyberWorlds. 130

DCT Discrete Cosine Transform. 15, 20, 26, 29, 30, 31, 34, 39, 52, 53, 56, 90, 91, 92, 93, 94, 95, 99, 100, 101, 102, 103, 104, 105, 106, 110, 111, 114, 115, 116, 117, 118, 119, 127, 130

DenseNet DenseNet Neural Network. 36, 37, 38, 63, 65, 66, 94, 124

DIF Digital Image Forensics. 13, 15, 17, 18, 20, 21, 22, 25, 26, 27, 28, 29, 30, 33, 34, 36, 37, 38, 39, 40, 41, 42, 45, 52, 53, 55, 57, 58, 59, 66, 73, 90, 91, 92, 93, 94, 97, 98, 99, 101, 115, 116, 117, 118, 119, 122, 123, 127, 128, 129, 130

DJPEG Double JPEG. 93

DJPEG-C Double JPEG-Compression. 14, 44, 90, 92, 93, 94, 95, 99, 101, 128

DL Deep Learning. 18, 22, 25, 27, 28, 29, 30, 33, 34, 36, 38, 40, 41, 46, 51, 52, 55, 57, 58, 59, 60, 61, 62, 63, 66, 68, 70, 77, 78, 80, 81, 89, 91, 92, 93, 99, 100, 105, 116, 117, 124, 125, 126, 128, 129

DNN Deep Neural Network. 15, 22, 27, 28, 29, 30, 34, 35, 36, 40, 41, 52, 53, 55, 58, 59, 99, 108

DWT Discrete Wavelet Transform. 53

EI Electronic Imaging. 130

ET Extremely randomized Trees. 35, 36

FC Fully-Connected. 23, 24, 35, 36, 37, 38, 39, 71, 72, 93, 107, 117, 118

FGSM Fast Gradient Sign Method. 55, 56

FN False Negative. 46, 47, 86, 96

FP False Positive. 46, 47, 86, 96, 110

FPR False Positive Rate. 47, 48

GAN Generative Adversarial Networks. 55, 56, 57, 58, 96, 119

GB Gaussian Blurring. 16, 42, 51, 52, 95, 119, 120, 121, 123, 128

GPU Graphics Processing Unit. 27, 33

GRU Gated Recurrent Unit. 96

HiFiC High-Fidelity Compression. 15, 16, 95, 96, 97, 98, 119, 120, 121, 122, 127, 128

HPF High-Pass Filter. 29, 31, 32, 62, 71, 72, 79

ICIP International Conference on Image Processing. 130

ICPRAM International Conference on Pattern Recognition Application & Methods. 130

IDCT Inverse Discret Cosine Transform. 101

ILSVRC Image Large-Scale Visual Recognition Challenge. 15, 27, 28, 40, 63

JPEG Joint Photographic Experts Group. 16, 20, 42, 43, 44, 45, 49, 51, 52, 53, 57, 58, 59, 71, 90, 91, 92, 94, 95, 96, 97, 98, 99, 100, 101, 105, 108, 116, 117, 119, 120, 121, 122, 123, 127, 128, 130

JSMA Jacobian based Salience Map Attack. 55

k-NN k-Nearest Neighbors. 55, 56, 100

L-BFGS Limited-memory Broyden-Fletcher-Goldfarb-Shanno. 55, 56

LSTM Long-Short Term Memory. 39, 96

MF Median Filter. 16, 30, 51, 52, 54, 55, 56, 75, 95, 119, 120, 121, 123, 128

MFC Media Forensics Challenge. 44

MFCN Multi-task Fully Connected Network. 37

MFR Median Filtering Residual. 30, 33, 54, 55, 56, 71, 72

MLP MultiLayer Perceptron. 37

MPEG Moving Picture Experts Group. 129

MS-SSIM Multi-Scale Structural Similarity. 96, 119

MSE Mean Squared Error. 24

MWSF Media, Watermarking and Security Forensics. 130

NA-DJPEG-C Non Aligned-Double JPEG-Compression. 91, 93, 94

Nadam Nesterov-Accelerated Adaptive Moment Estimation. 24, 107

NC Nimble Challenge. 36, 38, 39, 44

NEF Nikon Electronic File. 42, 45, 46

NIST National Institute of Standards and Technology. 44

NRCS Natural Resources Conservation Service. 45, 46

PGD Projected Gradient Descent. 55, 56

PGM Portable GrayMap. 45

PNG Portable Network Graphics. 42, 43, 44

PRNU Photo-Response Non-Uniformity. 19, 53, 59, 61, 82, 91, 99, 100, 104, 105, 106, 110, 114, 116, 117, 118, 119, 127, 130

PSNR Peak Signal to Noise Ratio. 14, 96, 97, 99, 119

QF Quality Factor. 15, 30, 71, 90, 92, 93, 94, 95, 96, 97, 98, 108, 120

ReLU Rectified Linear Unit. 23, 107

ResNet Residual Neural Network. 36, 37, 38, 63, 64, 65, 66, 72, 78, 79

RGB Red Green Blue. 31, 32, 39, 95, 101

RNN Recurrent Neural Networks. 95

ROC Receiver Operating Characteristic. 14, 15, 48, 86, 87

RTD Realistic Tampered Dataset. 44

SGD Stochastic Gradient Descent. 24, 33, 64

SIFT Scale-Invariant Feature Transform. 54

SN Social Network. 15, 21, 22, 27, 58, 74, 91, 92, 95, 98, 99, 100, 101, 102, 103, 104, 105, 106, 108, 109, 110, 111, 112, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 127, 128, 130

SNN Siamese Neural Network. 38, 39, 41, 50, 70, 71, 75, 82, 83, 125

SOTA State-Of-The-Art. 13, 14, 21, 31, 34, 48, 50, 51, 52, 60, 62, 66, 67, 69, 70, 74, 78, 79, 81, 82, 83, 86, 87, 88, 89, 93, 94, 95, 96, 97, 100, 106, 108, 110, 111, 112, 114, 115, 116, 117, 119, 124, 125, 126, 127, 130

SPN Sensor Pattern Noise. 59, 78, 105

SRM Spatial Rich Model. 29, 31, 32, 33

SRRSV Stratified Repeated Random Sub-sampling Validation. 110, 111

SSIM Structural Similarity. 14, 97, 99, 120, 121, 122, 123, 127, 128

SUN Scene UNderstanding. 45

SVM Support Vector Machine. 35, 36, 54, 62

TanH Hyperbolic Tangent. 23, 71

TIFF Tagged Image File Format. 42, 43, 44, 45, 46

TN True Negative. 46, 47, 86, 96

TP True Positive. 46, 47, 86, 96, 110

TPR True Positive Rate. 47, 48

t-SNE t-Distributed Stochastic Neighbour Embedding. 15, 80, 81, 87, 88

TSN Two-Stream Network. 13, 14, 15, 16, 34, 38, 39, 41, 82, 101, 106, 107, 108, 109, 110, 111, 112, 114, 115, 116, 118, 119, 120, 122, 123, 127, 130

UCID Uncompressed Color Image Database. 14, 36, 38, 39, 45, 46, 51, 108, 110, 111, 112, 114, 115, 120, 121, 122, 127

VCIP Visual Communications and Image Processing. 130

Video-ACID Video Authentication and Camera Identification Database. 129

Contents

List of Tables

List of Figures

1 Introduction

Nowadays, the volume of digital visual data (images and videos) increases significantly year after year. Image statistics have tripled between the beginning and the end of the last decade, from photos taken around the world to images posted on social networks (see Fig. 1). This growth in digital visual data is due in part to the development of new technologies (e.g., smartphones) and social media, which are increasingly present in our daily lives. At the same time, photo editing software are becoming more powerful and easier to use. With the rise of social media, these tools are even integrated into some smartphone apps, such as Instagram or Snapchat. When image editors first appeared, their purpose was to modify and improve the quality of images by changing their mathematical statistics. With their improvement, they were used to modify the content of digital visual data (e.g., facial filters on Instagram). Thus, these image editors can be used to tamper images by changing the semantic meaning of a photo or video (e.g., fake news). Notably, deep fakes have been quite famous in recent years, with some examples altering the reputation of American presidents.

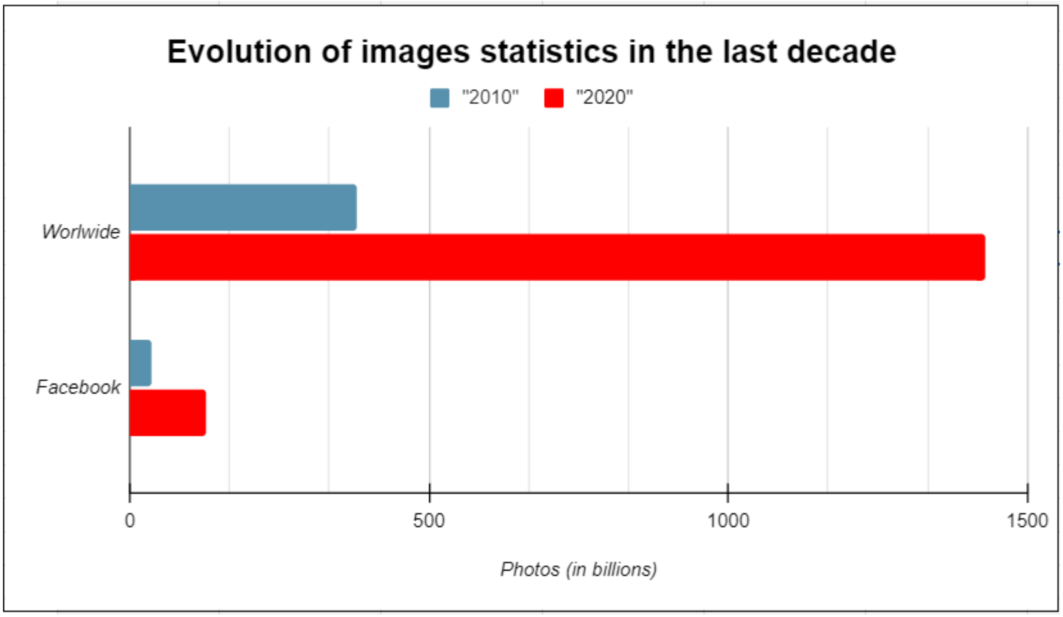


Figure 1: Evolution of images statistics between the beginning and the end of the last decade. Photo taken around the world and posted on Facebook. (numbers from Statista)

Digital Image Forensics (DIF) (Redi et al. 2011) represents a relatively new field whose objective is to guarantee the authenticity of digital visual data. DIF provides tools to blindly analyze images and give information about their origin or content. In fact, among the tasks of DIF, two main objectives stand out, namely: detection of forgeries and recognition of the source camera (i.e. sensor recognition). Originally (i.e., around 2005), to achieve these goals, many classical methods were designed, implemented and tested on different databases. Most of them were based on the analysis of mathematical statistics of images. As in several other areas of image processing, such as object detection or face recognition, new approaches based on Deep Learning (DL) and mainly on Convolutional Neural Network (CNN) have emerged and have surpassed the traditional approaches. The emergence of Artificial Intelligence (AI) architectures for DIF applications appeared later than for the other domains of image processing. However, the literature is already vast and this thesis analyzes the impact and advances of the DL on the DIF.

Thesis Outline

This chapter presents the background of the two domains that are addressed in this thesis: Deep Learning (DL) and Digital Image Forensics (DIF). The chapter 2 presents a comprehensive description of DL based approaches in terms of preprocessing step (unlike most other domains, raw images are rarely used directly as inputs), model architectures, databases, performances and associated metrics. Anti- and counter-forensic methods, classical and based on DL, are also discussed to provide a comprehensive overview. The chapter 3 is entirely devoted to camera recognition, and notably protocols to address issues from the literature. The chapter 4 tackles the impact of AI-based compression on DIF tasks, such as forgery detection or social network recognition. The thesis ends in chapter 5, with conclusions and future perspectives.

1.1 Digital Image Forensics

Images and videos play an essential role in digital communication, and they can be used as evidence for personal (social networking), legal (trial) or security (surveillance, police investigation) purposes. Therefore, verifying their source and authenticity is a crucial aspect to avoid any malicious use. However, as editing software are easy to access and use, falsified contents are becoming more common and increasingly difficult for humans to distinguish. Digital Image Forensics (DIF) emerged as a new solution to this type of malicious image editing, providing tools to blindly examine images and their mathematical statistics. The methods were first based on the analysis of these mathematical statistics, called artifacts in the literature. This section presents the main artifacts used by classical methods and the main applications of DIF that we have discussed in this manuscript.

1.1.1 Artifacts

Early approaches to the tasks were based on mathematical statistics of images. In particular, these methods analyze the artifacts that remain inside the digital image during its creation process, which consists of three steps: acquisition, post-processing and storage (see Fig. 2).

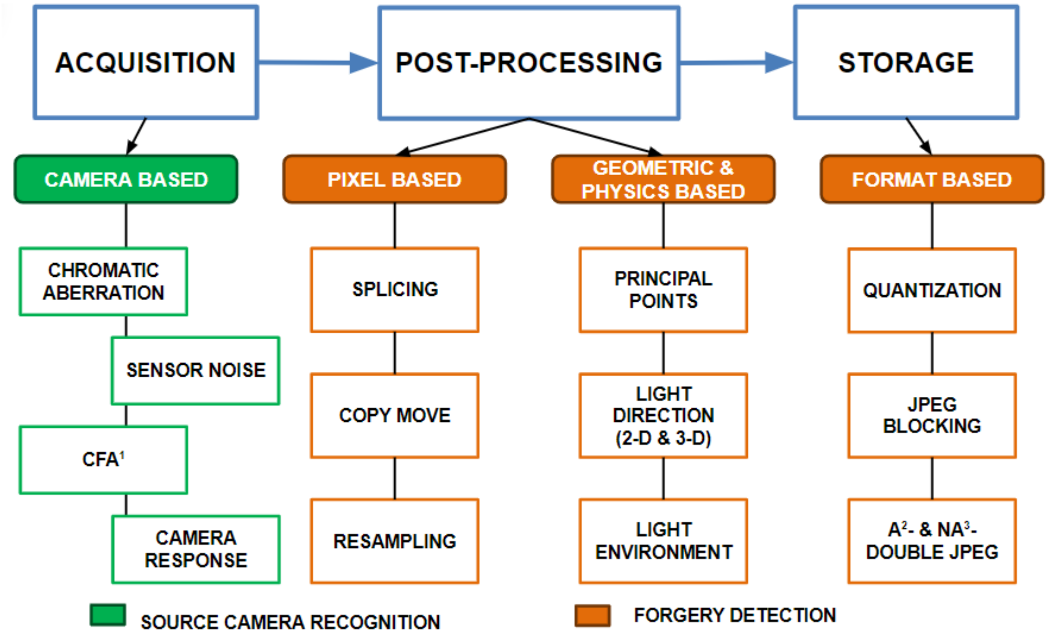


Figure 2: Traces of the digital image creation pipeline, with their dedicated application. [1]Color Filter Array (CFA), [2]aligned and [3]non-aligned.

These artifacts are caused either by the camera that captured the picture or by modifications of the image by users. The modifications can be harmless with the sole purpose of improving the image and its visual quality. However, these alterations can also be malicious, which leads to the analysis of artifacts to ensure the authenticity of the images. These artifacts can be grouped into four classes.

Camera-based artifacts

Camera-based artifacts are introduced during the acquisition step by the camera lens (Johnson and Farid 2006; K. S. Choi et al. 2006b, 2006a; Van et al. 2007), the sensor with the Photo-Response Non-Uniformity (PRNU) (Lukas et al. 2006; M. Chen et al. 2008) or the Color Filter Array (CFA). (Bayram et al. 2005; Popescu and Farid 2005b; Ferrara et al. 2012; Gallagher and Chen 2008; Mahdian and Saic 2008). Each of these elements plays a key role in digital image acquisition. The camera lens transmits the light from the scene to the sensor in a single point and transforms it into pixels. The camera assigns a specific color to each pixel, first through a mosaic of colors, then through an interpolated representation. This thorough process produces some artifacts due to imperfections in the camera model that captures the image. Therefore, these artifacts are mainly exploited for the recognition of the source camera, they can also detect falsifications applied to the content of an image.

Pixel-based artifacts

Pixel-based artifacts result from image modifications at the pixel level (Kirchner and Fridrich 2010; H.-J. Lin et al. 2009). The main manipulations of DIF are splicing, which merges a part of an image A into an image B ; cloning also called copy-move (Bayram et al. 2009), which copies a part of an image on itself. Resampling (Popescu and Farid 2005a) is also frequently used to match (resize, rotate, etc.) a falsified region to an image. These falsifications create artifacts that can be used to detect altered images. However, these artifacts can be hidden by applying specific methods, either to mask the falsifications or to enhance the image. This is why some detection methods focus on manipulations (Geradts et al. 2001).

Geometry and physics-based artifacts

Geometry and physics-based artifacts correspond to the inconsistencies that could be created in a forgery. When an image is modified, by adding or removing an element, the real features are rarely preserved. Indeed, light (*2-D*, *3-D*, environmental) or geometric parameters are usually neglected when falsifying an image, which can be exploited to detect the falsification (Johnson and Farid 2005, 2007). In addition, lighting inconsistencies can help to detect falsified images (Asati and Pardhi 2014).

Format-based artifacts

Format-based artifacts represent information from a specific compression approach. For example, during the quantization step, pixel blocks are converted to frequency space by the Discrete Cosine Transform (DCT). Anomalies can be introduced in the DCT coefficients (Z. Lin et al. 2009) and also in the Joint Photographic Experts Group (JPEG) block (Ye et al. 2007; Farid 2009a; Krawetz 2007). After a falsification, the image is usually recompressed and some inconsistencies can appear (e.g., in the JPEG block or the quantization matrix) (F. Huang et al. 2010; Luo et al. 2007; Kirchner and Gloe 2009).

1.1.2 Application Tasks

These artifacts are essential to the analysis of the images, as they reveal useful statistics about the images. Indeed, these statistics can help to determine the origin of the images or to detect falsifications. Source camera recognition and forgery detection are the main goals of DIF, even if the literature has addressed other tasks. Thus, in our manuscript, we focus on forgery detection and source camera recognition. However, we also address source social network recognition.

Forgery Detection

Altering an image can change its semantic meaning, and thus lead to potential repercussions if it is shared on social media. Falsifications can be applied in different ways by hiding (camouflage), deleting (removal) or adding information inside the images. There are two methods to perform these alterations: the *copy-move* (Popescu and Farid 2004) method, which is mainly used to remove or duplicate an object (e.g., a bird); and *splicing*, which is applied for camouflage or to include new materials in the image. These falsifications can be detected by artifacts that are essentially created or left behind during the post-processing or storage phases (see Fig. 2). This task is called falsification detection, and with the evolution of State-Of-The-Art (SOTA) methods, another specificity has been taken into account: falsification localization.

Source Camera Recognition

Society is increasingly confronted with images. Thus, their role has evolved. They can be seen as evidence, whether in a police investigation or a trial. It is therefore interesting to clarify the provenance of the images. Camera source recognition exploits the artifacts created by the camera during acquisition to provide information about the origin of the images. The literature shows different ways to recognize cameras (see chapter 3 for more details). However, the literature has few problems that we highlight and discuss deeper in our manuscript.

Social Network Recognition

Another task of DIF is dedicated to the clarification of the source of the images, based on their provenance from Social Network (SN)s. We also focus on SN recognition, which is close to source camera recognition. The context is quite similar: images can be identified as evidence of cybercrime, such as cyberbullying or instigating crimes. Thus, recognition SN is an interesting task that should be considered in our analysis of DIF applications.

Digital Image Forensics (DIF) is a fairly recent field that emerged to authenticate images in the growing context of image editors. The methods are conducted by analyzing the artifacts that are created during the digital image creation pipeline, which consists of three stages: acquisition, post-processing and storage. Artifacts can be grouped into four groups: 1) camera-based; 2) pixel-based; 3) geometry and physics-based; and 4) format-based. There are two main tasks, which have their dedicated artifacts, to authenticate images: forgery detection and camera recognition. Social Network (SN) recognition is a task that we also consider in our manuscript.

1.2 Deep Learning

The first methods of DIF were based on mathematical statistics and the analysis of dedicated artifacts. Then, with the emergence of architectures from Deep Learning (DL), as for many other fields of image processing such as object detection, the methods turned to Deep Neural Network (DNN). In particular, these methods were based on Convolutional Neural Network (CNN), which are well suited to images. In the following subsections, we present the background of CNN, and detail the evolution from classical methods to DL through survey analysis.

1.2.1 Background

In the specific domain of image processing, DL-based methods are mainly based on CNN. *Yann LeCun* proposed LeNet (Lecun et al. 1998), which is the oldest model. Other models, such as AlexNet (Krizhevsky et al. 2012) and GoogLeNet (Szegedy et al. 2015), are also well-known. These types of models are data-driven, meaning that they are trained with a labeled dataset (i.e., categorized images), which is usually divided into training, validation and test subsets (N.B. usually $80 : 10 : 10$).

Network architecture

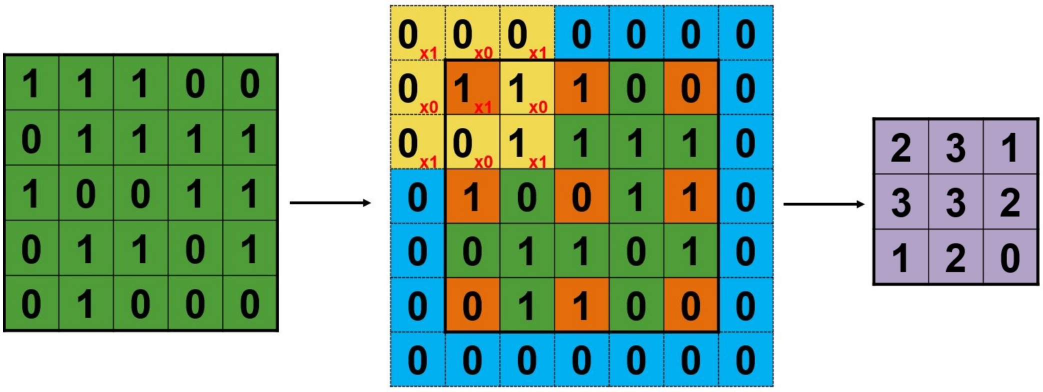
Deep Architectures and particularly CNN are built in a three-stage architecture:

- the input, which is usually a raw image (i.e. unprocessed).
- feature learning, which reduces the images to facilitate the process, without losing essential information to obtain appropriate predictions. This part consists of convolutional and pooling layers that, respectively, extract high-level features (e.g., edges) and reduce the size of convolutional features. The number of these layers depends mainly on the trade-off between computation

time and accuracy. Convolutional layers are defined with additional parameters that need to be defined, for each layer i, such as kernel K_i, padding P_i or stride S_i (see Eq.1, where N_i is the size of layer input) as well as activation and normalization functions.

$$N_{i+1} = 1 + (N_i + 2P_i - K_i)/S_i \tag{1}$$

The kernel represents the filter (size 3 × 3, see Fig. 3) that browses the whole input image during the convolution. The padding consists of adding zero pixels around the image (set to 1 in Fig. 3), and the stride is the step between each convolution (set to 2 in Fig. 3).



$$\begin{bmatrix} 1&1&1&0&0\\ 0&1&1&1&1\\ 1&0&0&1&1\\ 0&1&1&0&1\\ 0&1&0&0&0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0&0&0&0&0&0&0\\ 0&1&1&1&0&0&0\\ 0&0&1&1&1&1&0\\ 0&1&0&0&1&1&0\\ 0&0&1&1&0&1&0\\ 0&0&1&1&0&0&0\\ 0&0&0&0&0&0&0 \end{bmatrix} \longrightarrow \begin{bmatrix} 2&3&1\\ 3&3&2\\ 1&2&0 \end{bmatrix}$$

Figure 3: Application diagram of a convolution to an image. (left) input image 5×5 (center) kernel window (yellow) apply on image with the stride (orange) and the padding (blue). (right) output image

The activation (e.g., Rectified Linear Unit (ReLU), Hyperbolic Tangent (TanH), etc.) and normalization (batch or local-response) functions work in pairs: the former must have nonlinear output and the latter controls unbounded values. For pooling layers, there are also kernel and stride parameters, but the most important is the function used. Indeed, this kind layer reduces the size of convolutional features through a filter, like convolutions. But, in this case, the whole window is replaced by a value defined by the pooling function (max/min/average).

- The classification, which gives the label of the input image, i.e. the class to which it belongs (e.g. *cat* in the case of animal classification or *male* in the case of gender classification). The final output is obtained by a Fully-Connected (FC) layer, which connects all neurons together. First, the feature maps are flattened (i.e., the *3-D* input from the feature training part is transformed into *1-D* output). Then, the vector is reduced to a size N, where

N defines the number of classes, by sequential FC layers. This sequence is often combined with a dropout function, which corresponds to the random and temporal deletion of neurons (values are set to 0) in order to avoid overfitting during the training step. Finally, an activation function called Soft max is applied to set the best value of the group to 1 and the others to 0.

Training phase

After the creation of the architecture, the weights of the neurons are updated and controlled during the learning phase. This process is called the backpropagation (Fig. 4). This part is performed with a labeled dataset, and the goal is to update the weights of the neurons to obtain a result as close as possible to the expected one. Two elements must be defined: 1) the loss function (e.g. Cross Entropy, Mean Squared Error (MSE), etc.) which defines how incorrect the results are; 2) the optimizer (e.g. Stochastic Gradient Descent (SGD), Nesterov-Accelerated Adaptive Moment Estimation (Nadam), etc.) which performs the changes on the weights.

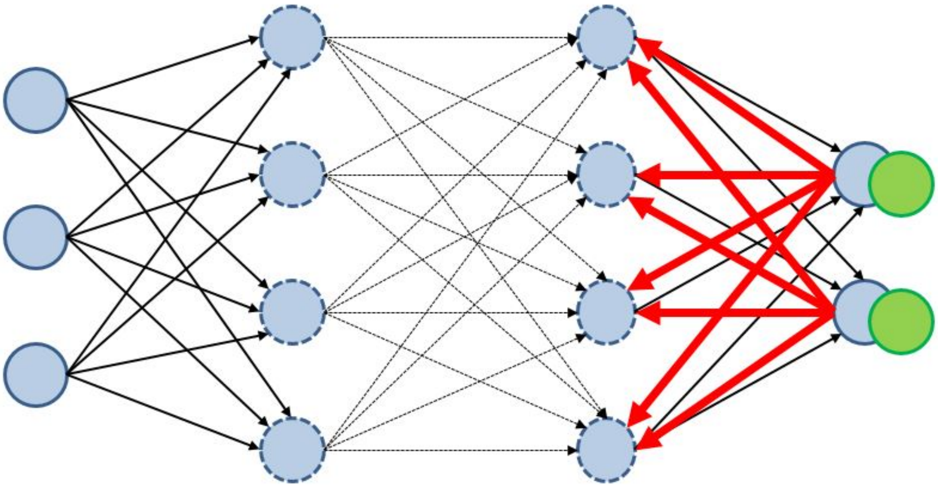


Figure 4: Diagram of a back-propagation (red), with the connections between neurons (circle), the expected output (green).

In addition, the training dataset is usually divided into different batches to feed the network in order to avoid excess memory (too large dataset). This process is conducted over several iterations to accomplish an epoch (i.e., the dataset is passed back and forth in the model). For example, if there are $2,000$ images, they can be divided into batches of 100 images to be used in the model for 20 iterations

$(20 * 100$ to complete the dataset) to complete 1 epoch. Moreover, to avoid over-fitting (learning the dataset too accurately) or under-fitting (lack of learning), we have to find the right compromise to obtain a high accuracy. Indeed, if the dataset is small, the number of layers should be reduced, whereas it is the opposite for wide datasets. Then, the model is ready to be evaluated with a labeled dataset to obtain the performance results.

1.2.2 Existing Surveys

Various reviews have been published in the literature on classical and Deep Learning (DL)-based methods of DIF. Most of them are addressing classical methods and are dealing about specific or complete techniques (artifacts or forgery). However, surveys on Deep Learning (DL)-based reviews have been released more recently, after the emergence of such architectures in DIF (see Tab. 1).

Table 1: Surveys dealing with digital image forensics. Order by year of release.

Survey	**Approach**	**Objective**	**Contribution**
A Comprehensive Review of Deep-Learning-Based Methods for Image Forensics (Castillo Camacho and Wang 2021) - 12 cit.	DL	DIF and deepfakes	Present a wide review of DL methods for DIF.
Media forensics and deepfakes: an overview (Verdoliva 2020) - 189 cit.	Classical and DL	DIF and deepfakes	Provides a comprehensive overview on methods and databases for DIF and deepfakes. In the case of DIF, classical and DL techniques for forgery detection are detailed.
Review of Imaging Device Identification Based on Machine Learning (J. Wu et al. 2020) - 6 cit.	Classical and DL	Camera Recognition	Present the different methods that have been used for source camera recognition.
A survey of DL-based source image forensics (Yang et al. 2020) - 35 cit.	DL	Source image forensics	Details the methods of various domains from source image forensics as the detection of recaptured images and the recognition of source cameras. Another part is dedicated to counter-forensic methods.
A survey on digital image copy-move techniques (Tan et al. 2019) - 5 cit.	Classical and DL	Copy-move	Provides an overview of copy move detection approaches using passive methods classified into three types: block-based methods, key point-based methods and DL.
Image Forgery Detection: Survey and Future Directions (Meena and Tyagi 2019) - 28 cit.	Classical	Forgery Detection	Depicts a comparison of 58 pixel-based methods. A comparative study is made according to 4 types of tampering: image splicing, copy-move, resampling and retouching.
Recent advances in passive digital image security forensics: A brief review (X. Lin et al. 2018) - 56 cit.	Classical	Artifacts	Details of recent techniques according to the artifacts left in the digital image during these pipeline steps: acquisition, storage and editing.

Survey	Approach	Objective	Contribution
Digital image integrity – a survey of protection and verification techniques (Korus 2017) - 88 cit.	Classical	Active protection and forgery detection	Summarizes various techniques of image authentication domains and for DIF the methods are classified according to the traces exploited (noise, sensor, etc.).
Large-scale evaluation of splicing localization algorithms for web images (Zampoglou et al. 2016) - 145 cit.	Classical	Splicing	Presents classical methods for splicing localization and conducts a comparative study based on various datasets.
A bibliography of pixel-based blind image forgery detection techniques (Ali Qureshi and Deriche 2015) - 172 cit.	Classical	Artifacts and forgery detection	Provides a comprehensive overview of artifacts-based techniques as well as forgery detection methods.
Pixel-based image forgery detection: A review (Ansari et al. 2014) - 130 cit.	Classical	Pixel-based techniques	Gives an overview of pixel-based techniques for image forgery detection.
A review on copy move image forgery detection techniques (Ali Qureshi and Deriche 2014) - 59 cit.	Classical	Copy-move	Gathers pixel-based techniques and especially summarizes copy-move detection methods.
Passive detection of copy-move forgery in digital images: State-of-the-art (Al-Qershi and Khoo 2013) - 205 cit.	Classical	Copy-move	Focuses on copy-move detection algorithms based on various artifacts (DCT, key-points, etc.).
An evaluation of popular copy-move forgery detection approaches (Christlein et al. 2012) - 799 cit.	Classical	Copy-move	Presents copy-move detection algorithms based on block-based and key-point features.
Digital Image Forensics: A booklet for beginners (Redi et al. 2011) - 377 cit.	Classical	DIF	Provides a complete overview on algorithms for DIF. Methods are gathered by their topics: camera identification, forgery detection and counter-forensics.
A bibliography on blind methods for identifying image forgery (Mahdian and Saic 2010) - 264 cit.	Classical	Forgery detection	Summarizes classical methods according to 15 various artifacts (splicing, recompression, local noise, etc.).
Digital image forensics (Fridrich 2009) - 365 cit.	Classical	Sensor-based techniques	Gives a focused overview on sensor-based artifacts used for the task of forgery detection and camera identification.
Image forgery detection (Farid 2009b) - 795 cit.	Classical	Artifacts	Provides a complete description of the techniques exploited for image forgery detection. The methods are presented according to the type of artifacts.
Overview of state-of-the-art in digital image forensics (Sencar and Memon 2008) - 195 cit.	Classical	DIF	Gives an overview of methods for discrimination of synthetic images and of various artifacts used for camera recognition and forgery detection.
A survey on digital camera image forensic methods (Van Lanh et al. 2007) - 175 cit.	Classical	DIF	Describes various techniques for authenticating camera and detecting forgery. The techniques mentioned are classified according to the artifacts exploited.

Classical Reviews

All these techniques based on artifact analysis have been classified and analyzed in several review articles. These studies focus on the classical methods used to authenticate a camera or detect tampering. Some of them are specific to a single artifact, such as sensor-based techniques (Fridrich 2009) or a particular manipulation, such as splicing (Zampoglou et al. 2016). (Ansari et al. 2014) address pixel-based techniques, while more articles are covering copy-move methods (Ali Qureshi and Deriche 2014; Al-Qershi and Khoo 2013; Christlein et al. 2012).

Other surveys describe techniques for an entire objective, such as forgery detection (Mahdian and Saic 2010; Korus 2017; Meena and Tyagi 2019). For example, (X. Lin et al. 2018; Farid 2009b) detail all types of artifacts that can be exploited for DIF. Some of these articles address different topics to be more complete as (Ali Qureshi and Deriche 2015), which present both artifacts and falsification detection are studied.

Some surveys are also analyzing DIF in its entirety (Van Lanh et al. 2007; Redi et al. 2011; Sencar and Memon 2008). In these reviews, each artifact, forgery and classical method are tackled in respective sections.

DL-based Reviews

DL-based methods have also been discussed recently in a few surveys. (Tan et al. 2019) is restricted to copy move operations, and DL is limited to a single subsection. (Yang et al. 2020) present subdomains of source image forensics in general. For DIF, only recaptured image detection, source camera and SN recognition are discussed. Part of this review also covers anti- and counter-forensic methods. (J. Wu et al. 2020) also discuss the DL-based methods for source camera recognition in a section. Therefore, DL-based techniques presented in these articles are specific to the particular purpose of DIF.

(Verdoliva 2020) deal globally with DIF based on DL and also with deepfakes. This survey is comprehensive as it presents methods and databases for both purpose. Despite its undeniable contribution, this overview is not fully dedicated to DIF and does not provide performance comparison. (Castillo Camacho and Wang 2021) is also addressing deepfakes and DIF. The review presents most of DL-based methods that are used in DIF applications.

From classical to DL

The increase in DL studies is recent, as this domain emerged with the availability of fast, parallel computing devices (e.g., Graphics Processing Unit (GPU)), and large, high-quality labeled data sets. The methods have been applied to several domains such as natural language processing, image classification or facial recognition. In particular, the performances obtained for classification challenges such as Image Large-Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al. 2015) have proven the effectiveness of DL-based methods (in 2012 and 2015).

Since 2015, the application of Deep Neural Network (DNN) has emerged in the field of DIF. As shown in Fig.5, articles using DNN have eclipsed traditional DIF publications in 2017. In the chapter 2, we examine DL-based methods for DIF with other perspectives by analyzing preprocessing modules, detailing architectures and databases. In addition, we also highlight some issues and opportunities from the literature that have pushed our research in certain directions.

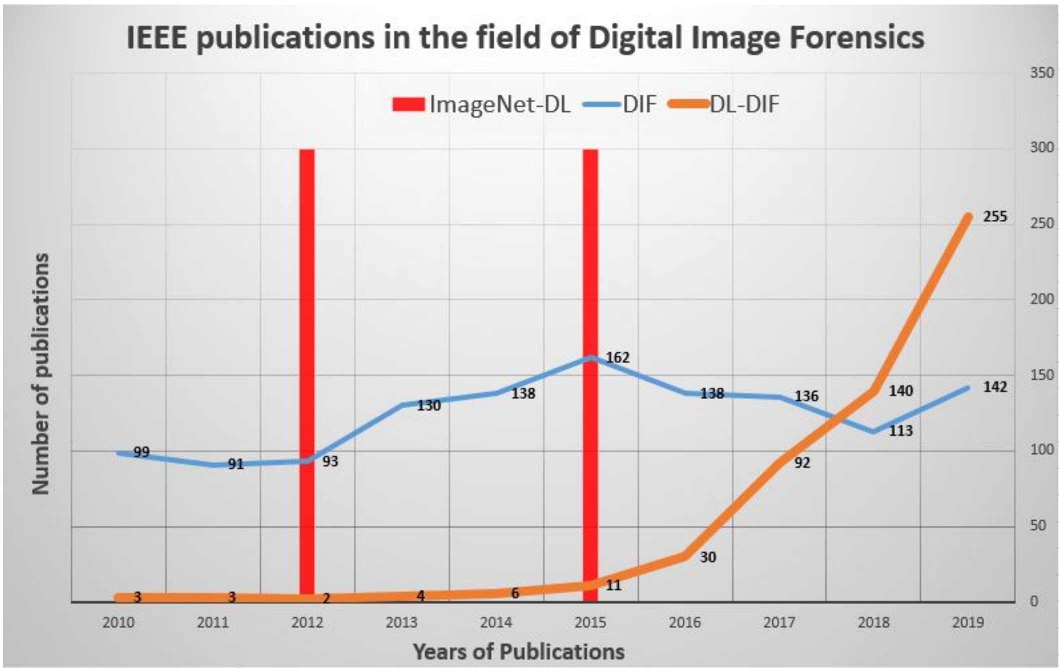


Figure 5: Number of publications in the field of DIF over the last 10 years. Data were retrieved from the IEEE explore website, with the query *Digital Image Forensics* for DIF and the queries *Camera Identification Deep* and *Forgery Deep* for DNN-DIF. The red lines represent two years of achievement from DL methods in ILSVRC (Russakovsky et al. 2015).

Deep Learning (DL) has emerged in Digital Image Forensics (DIF) as for other image processing areas. Notably, Convolutional Neural Network (CNN) are efficient for images and their architecture is built in three parts: 1) the input; 2) the features learning; 3) the classification. Existing surveys testify of the recent interest, as only few of them are addressing DL. However, articles have already switched from classical to DL-based methods since 2017.

2 State-of-the-art Review

2.1 Pre-processing Modules

In recent years, many image processing tasks have used methods based on Deep Learning (DL), especially Deep Neural Network (DNN). Some areas of image processing such as face recognition or steganalysis have adopted these neural networks. However, there are differences between these two examples. For face recognition, digital images are fed directly into the network, which learns the features of the pixels. In the case of steganalysis, a pre-processing step based on the filters of the steganalysis rich model (also called Spatial Rich Model (SRM)) is applied (Chaumont 2019). Digital Image Forensics (DIF) is closer to this domain than to face recognition, as the pre-processing modules must extract dedicated artifacts before the network learns the features. In the following subsections, we explain the reasons for the pre-processing step in the DL based methods and present the main artifacts. There are two ways to apply these modules: before the network or in the first layer of the network.

2.1.1 Artifacts Extraction

In a preliminary study, (J. Chen et al. 2015) attempted to authenticate raw images directly with a DL-based method. Then, (Tuama et al. 2016) proposed to use a High-Pass Filter (HPF) before the network to highlight relevant artefacts. In fact, DNN does not learn key statistical properties relevant to forensic image analysis. This means that forgery detection or any other type of image analysis cannot be performed with the usual methods. In fact, traditional methods still rely on relevant artifacts to detect fakes. Therefore, the equivalent process for DNN is the extraction of traces left by forgeries. Without this crucial step, the network only learns the features of the image content, resulting in disappointing performance in detecting forgeries. To properly authenticate images, it must learn about hidden artifacts that are masked by the image content. Pre-processing is therefore a mandatory step in the methods. In the literature, two main artifacts are exploited to perform an analysis : noise residues and the histograms of the Discrete Cosine Transform (DCT) coefficients.

Noise Residues

Noise residues are one of the main artifacts extracted during the pre-processing phase. Each image has a specific noise due to the camera that captured it and the operations applied to the content. Indeed, an alteration frequently produces noise, either to hide a previous operation or to modify the image. These residues can therefore be used as elements for any task (e.g., tampering detection, camera recognition). However, these traces are often overshadowed by the image content. Whatever the type of denoising filter, the technique to obtain the residues is similar. It consists in subtracting the denoised image *F(I)* from the original image I to obtain the desired artifact $\tilde{I}$ (see Eq. 2).

$$\tilde{I}_n = I_n - F(I_n) \tag{2}$$

Histograms of DCT coefficients

After falsifications, the tampered image is usually stored again by applying another compression with a different Quality Factor (QF). This affects the distribution of the histograms of the DCT coefficients. In the case of single compression, the histograms follow approximately a generalized Gaussian distribution (regardless of the QF) whereas the histograms show some anomalies for double compression. These anomalies depend on the QF used for the two steps: i) if QF1 (1^{st} compression) is larger than QF2 (2^{nd} one), there will be peaks and valleys in the histogram; ii) there will be missing values for the inverse (i.e., QF2 superior to QF1). These differences illustrate the application of recompression and the consequence of possible alteration. Thus, the histograms of the DCT coefficients are mainly used to detect recompressed images.

2.1.2 Classical techniques

The pre-processing module can be applied before the DNN, resulting in a two-step model: first the artifact extraction, then the network. This way of applying the pre-processing is close to the classical methods (i.e., without DL). Indeed, for these methods, features are first extracted and then used to detect forgeries. In the literature on DL for DIF, various techniques have been used, including residues (noise or Median Filter (MF)) and histograms of DCT coefficients.

Noise Residues

(J. Chen et al. 2015) implemented the first pre-processing approach for DIF with DL. This technique is also based on filtering (Median Filtering Residual (MFR)). It aims to remove interference from irrelevant information, which are the edges and textures of the image. The process is almost the same as for the application of a denoising filter. The residuals $d(i, j)$ are the results of the difference between the output $y(i, j)$, obtained by applying a $w \times w$ MF window on the image, and the image $x(i, j)$ (see Eq. 3).

$$d(i,j) = med_w(x(i,j)) - x(i,j) = y(i,j) - x(i,j) \tag{3}$$

The application of a High-Pass Filter (HPF) (Qian et al. 2015) represents another way to isolate noise from the image. Different HPF have been used in the State-Of-The-Art (SOTA) literature. (Tuama et al. 2016) used the HPF from (Qian et al. 2015) prior to the network and they compared it to a wavelet-based filter (Fridrich 2009).(Pengpeng et al. 2017) applied a Laplacian filter (3 × 3), usually used for edge detection, on small patches (64 × 64) to detect recaptured images. Indeed, the Laplacian filter is dedicated to the identification of regions with fast intensity changes. Therefore, this technique is sensitive to noise because it causes a wave effect on the image.

The SRM (Fridrich and Kodovsky 2012) is based on 30 basic HPFs, with non-linear operations, from seven groups (1^{st}, 2^{nd} and 3^{rd} orders, $EDGE$ and $SQUARE$ with kernels of size 3 and 5). They are used in the calculation of residual maps, and their results can be considered as a local noise descriptor.

$$F_{sqrt5} = \frac{1}{12} * \begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & \mathbf{-12} & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix} \tag{4}$$

(Kim and Lee 2017) use one ($SQUARE$ with a kernel of size 5, see Eq. 4) because the input is on a single channel. In comparison, (Zhou et al. 2018) use three, because their input is in RGB (2^{nd} order, $SQUARE$ with kernel size 3 and 5). In both cases, they use as few filters as possible to reduce the computation time.

Histograms of DCT coefficients

(Wang and Zhang 2016) specify a fixed interval that solves the problem of variable size for histograms of DCT coefficients, and reduces the computation with negligible loss of information. (Amerini, Uricchio, Ballan, et al. 2017) also exploit this technique, but with a different interval size (vectors of size 909 × 1 against 9 × 11 for the first).

(Barni, Bondi, et al. 2017) detail a CNN with "embedded histograms of DCT coefficients". The network includes a first step devoted to pre-processing with the calculation of histograms of coefficients. The network is fed with small raw patches (64×64), which are then handled in the pre-processing step. This first part extracts the self-learning artifacts without accessing the pixel values. Even if the network can be considered as end-to-end, the pre-processing is somehow equivalent to a classical technique because it is not impacted by the learning phase. It is therefore an intermediate step between the techniques performed upstream of the network and those integrated into it.

2.1.3 Network incorporated techniques

The second type of techniques is implemented in the network and is subject to the learning phase. Contrary to the previous ones, these methods automatically extract the artifacts within the model. The extraction is applied in the first convolutional layer by changing its weights. The first layer is then considered as a preprocessing layer. In fact, the network is forced to learn the particular artifacts that cause the difference with the standard models.

Noise Residues

(Cozzolino et al. 2017) explain how to recast a residual-based local descriptor into a CNN. First, the processing chain of a residual-based local descriptor consists of several steps. Only the first one is dedicated to the extraction of the noise residues. This phase is usually performed with a HPF to highlight the relevant artifacts. The conversion of this model into a CNN is done in two phases: 1) from local features to a feature bag paradigm; 2) then, to a CNN. However, only the extraction of artifacts is related to pre-processing. Indeed, the noise residues R are obtained by a group of shifted filters. This corresponds to a bank of N filters in the case of Bag-of-Features, then to a convolutional layer in the case of CNN. Finally, they replaced these filters by a convolutional layer that computes the residuals with the same filter coefficients.

(Rao and Ni 2016) describe another technique, influenced by steganalysis, applying SRM to the weights of the first convolutional layer of the network. (Zhou et al. 2018) have also used these filters in their classical pre-processing module, for the computation of residual maps. Instead of only three, this technique exploits 30 basic HPFs, from seven c_i classes, to initialize the weight of the convolutional layer. The output of this first layer is a set of 30 feature maps. In the case of a RGB input (three color channels), the outputs are obtained with three SRM filters. For optimal results, the three filters used for the feature maps should be of the same class category, but not identical (i.e. $SQUARE$ 3, 5 and 7). Therefore, the initialization strategy is to associate a set of three filters with each feature map. Applying the SRM filters, even for weight initialization, highlights sharp edges that are introduced by alteration operations such as splicing. In addition, (Rao and Ni 2016) state that this initialization speeds up the convergence of the network.

Constrained Convolutional Layer

Finally, the last preprocessing technique introduced in the literature is also a modification of the first convolutional layer. (Bayar and Stamm 2016) propose an innovative approach with the constrained convolutional layer. The first layer of the

CNN is forced to extract features for manipulation detection. The key to this process is still the same as previously presented with other pre-processing techniques: isolate artifacts that are overshadowed by the image content. Therefore, the task of this constrained convolutional layer is to remove irrelevant information, which is accomplished by updating the weights. The precise artifacts retained by this pre-processing layer are prediction error filters that provide the value of the center pixel of the filter window. The weights w of each filter K are forced as follows: the center value is set to -1 while the sum of the remaining pixels is set to 1 (Eq. 5).

$$\begin{cases} w_k(0,0) = -1 \\ \sum_{l,m\neq 0} w_k(l,m) = 1 \end{cases} \tag{5}$$

This constraint is applied during the learning part of the network with a particular sequence. The weights of each filter are randomly initialized, as is usually the case. Then, a two-step iterative process is started with the back-propagation until the value of the loss function is reached: the weights are first forced by the constraint, and then updated according to the Stochastic Gradient Descent (SGD). As a consequence, to this first article on the constrained convolutional network, a series of papers have been published. (Bayar and Stamm 2017a) represent an innovation of the preprocessing layer. It exploits a dual-stream filtering layer to capture prediction error filters, as before, but also non-linear artifacts. The input image of size $256 \times 256 \times 2$ is processed separately by each pre-processing module. It passes through the constrained convolutional layer and is processed in parallel, first with a Median Filtering Residual (MFR), then with an identity convolutional layer. The two outputs are then merged with a concatenated layer. As a result, this preprocessing technique provides more artifacts and thus more accurate feature extraction.

2.1.4 DIF Specificity

Unlike other areas of image processing, such as face recognition for example, DIF is only at the beginning of the adoption of methods based on DL. Some pre-processing techniques are inspired by steganalysis, such as the use of SRM filters. Indeed, the objective is to highlight the artifacts dedicated to DL as it is done with classical methods. Therefore, typical pre-processing techniques are based on noise residues or compression traces. This section shows that pre-processing is an essential step for camera recognition or tampering detection, whether it is applied upstream of the network or integrated into it.

However, the techniques used inside the network benefit from the computational power of the Graphics Processing Unit (GPU) that are used with the DL-based methods. The impact of each pre-processing is analyzed in the section 2.3

with the comparison of some State-Of-The-Art (SOTA) methods. The architectures used in these methods are also another element that affects performance, and we detail their variety in the section 2.2.

Unlike other domains, images must be handled by pre-processing modules before being fed into the network. There are two ways to apply it: before the network or integrated to it. Noise residues and histograms of DCT coefficients are the main artifacts extracted. With the DNNs, the extraction is performed in the first layer, and the constrained convolutional layer is one of the most efficient techniques.

2.2 Architectures

The main objective of Digital Image Forensics (DIF) is image authentication, which can be achieved mainly by two means: camera recognition and forgery detection. Initially, only classical methods were present to perform these tasks. Section 1.2 detailed the impact of Deep Learning (DL) in the literature with a radical change in research. This change has led to the proposal of different architectures, which have their own specificities from one to another. Since 2015, several papers describing DL approaches for DIF have been published.

Regardless of their purpose, their architectures can be broadly similar or completely different and innovative. In this section, we detail the architectures used for DIF. Even if the CNN represents the main network, as in almost all image-based domains using DL, other Deep Neural Networks (DNNs) are used. They can be based on innovative techniques or use the Convolutional Neural Network (CNN) as a baseline. Thus, the description is divided into four parts, each dealing with a type of architecture: 1) CNNs; 2) networks with improved features; 3) Two-Stream Networks (TSNs); 4) other approaches.

2.2.1 CNNs

Convolutional Neural Network (CNN) was one of the first networks to be used in the field of image processing. Convolutional layers are particularly effective at learning relevant features from images. The relevant forensics features require artifact extraction (see section 2.1). Along with pre-processing modules, CNNs also perform well, and are thus quite exploited in the literature. This subsection presents the essential parameters of these CNNs, which vary from one paper to another.

Layers Number

A CNN is composed of several layers that are divided into two parts: feature extraction and classification. One of the differences between these networks is the number of layers for both parts. In fact, the more layers, the better. But on the other hand, training the model takes more time and the risk of over-fitting is higher. Usually, there are between 3 and 5 convolutional layers and 2 or 3 Fully-Connected (FC) layers. However, the most recent methods for image processing tasks are using wide deep architectures (e.g. EfficientNet (Tan and Le 2019) or ConvNet (Liu et al. 2022)). But in some cases, especially for small datasets, increasing the feature learning layers and decreasing them in the classification part offers a solution to avoid over-fitting. For example, (Rao and Ni 2016) use respectively 8 layers and one for feature extraction and classification parts. (Y. Wu et al. 2019) propose a method built with 12 convolutional layers, which is higher than usual, and with a particular classification part. They proposed a local anomaly detection network that extracts the anomalous features based on the output of the extraction part. Therefore, one way to customize CNN for a particular application is to change the classification part.

Classification

Notably, instead of using a FC layer, a machine learning classifier is an alternative technique. (Bondi, Baroffio, et al. 2017) use a Support Vector Machine (SVM) as a classifier, while (Bayar and Stamm 2017c, 2017a) propose an Extremely randomized Trees (ET) and a FC layer in parallel. In both cases, the classifier is applied after the second FC layer: the SVM uses the vector output to obtain the final result and a deep feature strategy (Donahue et al. 2013) exploits the output and passes it to the ET. However, most of the methods use FC layers for the classification part.

Input Size

The input of the network is also an aspect that differs for each DNN. Mainly, images are divided into patches (of sizes 64, 128, 256 or 512) because the computational cost of a large image is too high. The smaller the patch, the more difficult it is to classify. On the other hand, the use of smaller patches offers more *accuracy* in locating falsifications. (Marra et al. 2020) propose a method that exploits a complete image as input to an end-to-end CNN.

In summary, there are three essential aspects to analyze for the architecture of the CNN. Some of these differences are described in Tab. 2. The convolutional layer represents a baseline for images that is exploited in other architectures, regardless of the objective (i.e., camera recognition or forgery detection).

Reference	Input	Classification	Database
Camera Recognition			
(Tuama et al. 2016)	256 × 256	FC layers	*Dresden*
(Bondi, Baroffio, et al. 2017)	64 × 64	SVM	*Dresden*
(Bayar and Stamm 2017b)	256 × 256	FC layers	IEEE IFS-TC
(Bayar and Stamm 2017a)	256 × 256	FC layers /ET	*Dresden*
(Bayar and Stamm 2018b)	256 × 256	FC layers	*Dresden*
(Junior et al. 2019)	40 × 40	▨	*Dresden, Vision*
(Kirchner and Johnson 2020)	64 × 64	Various (SVM, ET, etc.)	Private datasets
Forgery Detection			
(J. Chen et al. 2015)	64 × 64	FC layers	*BOSSbase, UCID, Dresden*
(Bayar and Stamm 2016)	256 × 256	FC layers	Private dataset
(Wang and Zhang 2016)	64 × 64	FC layers	*UCID, Dresden*
(Rao and Ni 2016)	128 × 128	FC layers	*Cavias, Columbia gray*
(Pengpeng et al. 2017)	512 × 512	FC layers	Private dataset
(Cozzolino et al. 2017)	128 × 128	FC layers	Private dataset
(Kim and Lee 2017)	256 × 256	FC layers	*BOSSbase*
(Bunk et al. 2017)	64 × 64	FC layers	*NC16*
(H. Choi et al. 2017)	64 × 64	FC layers	*BOSSbase, Dresden*
(Barni, Bondi, et al. 2017)	64 × 64	FC layers	Raise
(Bayar and Stamm 2017c)	256 × 256	FC layers /ET	Private dataset
(Bayar and Stamm 2018a)	256 × 256	FC layers	*Dresden*
(Y. Wu et al. 2019)	256 × 256	Anomaly Detection	*Dresden, Kaggle*
(Marra et al. 2020)	Image	FC layers	*Vision, UCID*

Table 2: Architecture details of CNN methods. Information on input size, classification part and databases used.

2.2.2 Enhanced-features Networks

Pre-processing modules were used to highlight relevant artifacts that are often overshadowed by the image content. In the same way, forensic features can be emphasized inside the network. In fact, DNN can be customized to exploit other features. CNNs are known to systematically go deeper into the extracted features. This means that in the first layers, the output maps are low-level features, while the last convolutional layers of the extraction part provide higher-level features. On the other hand, features are set aside through each layer as they move from low- to high-level. However, these low-level features may appear to contribute to DIF.

Shorcut-based Networks

DenseNet Neural Network (DenseNet) and Residual Neural Network (ResNet) are known in DL to preserve low-level features while computing convolutions to obtain high-level features. The process is to skip i connections ($i \geq 1$) between layers. The output of the N layer will be added to the output of the $N{+}i$ layer, without additional parameters or convolutional filters. Thus, the network learns features while including the skipped parts (called residuals). Dense connectivity consists of connecting each layer directly to successive layers. For example, the output of the layers N and *N+1* will be respectively connected to the layers $N+i$ and $N+i+1$ ($i \geq 1$) and so on. Therefore, ResNet (Kuzin et al. 2018; Y. Chen et al. 2017) and DenseNet (Ding et al. 2019) have been used for DIF applications. (Y. Wu et al. 2018b) present RemNet, composed of Remnant blocks, which contain 3 successive convolutional layers in parallel with shortcut connections. (M. Zhao et al. 2020) present a method that combines residual and regular convolutional layers to extract features. Combining networks is also an approach discussed in the literature.

Combination Networks

(Tang et al. 2017) also exploit this technique of shortcut connection, combining the extraction of various features to extend the image analysis. Indeed, they applied multiscale convolutional layers and *mlpconv* layers (M. Lin et al. 2014). The multiscale convolution consists in applying different kernels (1, 3 and 5) to obtain correlations between adjacent pixels of different sizes. The *mplconv* connects the input to the output features vector, with a MultiLayer Perceptron (MLP) that consists of multiple FC layers with non-linear activation functions. (Zhong and Pun 2020) address the combination of various features extraction and shortened connections, with a pyramidal feature extractor. This block consists of four Dense-InceptionNet layers and dense connectivity. The five outputs (input and output of four layers) are then concatenated. The shortcut connections are also exploited after each extractor.

Innovations

In comparison to classical CNNs, the enhanced-feature architectures present different ways to extract features, but also provide innovative classification parts (see Tab. 3). A correlation matching module is used to improve the network output. There is a triple classification based on FC layers, which comes directly from the shortcut connections. The features, which are extracted at each step, are also used for classification.

(Salloum et al. 2018) detail a Multi-task Fully Connected Network (MFCN) that provides two mask outputs: one for edges and another for the surface. (Wei et al. 2018) also propose an estimation of the mask for the classification. They employed two CNNs accordingly: i) the first to roughly distinguish the altered regions; 2) the second to refine the detection. These two networks are used successively, but some methods use both at the same time. This type of architecture

Reference	**Input**	**Features extraction**	**Classification**	**Database**
Camera Recognition				
(Y. Chen et al. 2017)	256 × 256	ResNet	FC layers	*Dresden*, private dataset
(Kuzin et al. 2018)	480 × 480	DenseNet	FC layers	*Kaggle*
(Ding et al. 2019)	48 × 48	ResNet	Triple Classification	*Dresden*
(M. Zhao et al. 2020)	48 × 48	ResNet + CNN	Triple Classification	*Dresden*
(Rafi et al. 2019)	64 × 64	Remnant	Convolutional layers	*Dresden*
Forgery Detection				
(Tang et al. 2017)	64 × 64	Multi-scale conv. and *mlpconv*	FC layers	*BOSSbase, UCID*
(Wei et al. 2018)	Image	C2R: 2 CNNs	Mask Estimation	*Casia v2*
(Salloum et al. 2018)	Image	Multi-task fully-connected network	Mask Estimation	*Casias, Columbia color, NC16*
(Y. Wu et al. 2018b)	256 × 256	CNN + Pointwise feature extractor	Mask Decoder	*Casias*, private dataset
(Zhong and Pun 2020)	Image	Pyramid features extractor	Correlation matching	*MICCs, GRIP, SUN, Coverage, CMH*

Table 3: Architecture details of methods using enhanced convolutional network. Information on input size, extraction and classification part and databases used.

is considered as TSNs. The following subsection details their use in the context of DIF.

2.2.3 Two-Stream Networks

CNN represents a baseline for image processing applications based on DL. It is usually used directly, but can also help to build a more creative architecture. TSNs are built with two models that process in parallel. In general, CNN are used as feature extractors in each stream, and then the classification part is applied to both outputs. The feature extraction is relatively similar to the previous networks, while the classification is really different. There are two types of TSNs, depending on the goal: either to compare two inputs (with the same labels or not), or to increase the diversity and the number of features of an image.

SNNs

Siamese Neural Networks (SNNs) are also built in two parts: extraction and similarity comparison. The extraction part is made of similar models (same layers, shared weights), called subnetworks, to obtain the same feature shape from input images. Their outputs are then compared for a binary classification. (Mazumdar et al. 2018) use a distance layer to compare the features and classify the images into identical or different pairs. (Mandelli et al. 2020) propose the similarity network, another method to classify two images. (Rao et al. 2020) is quite particular because it uses a SNN as a local descriptor, which is used in parallel with a classical CNN for feature extraction. This method is similar to the second type of TSN: the two-domain network.

Two-domain Networks

Two-domain networks are more flexible because both subnetworks used for feature extraction can be different. Also, there is usually only one input image, which is expressed in two different ways (one per subnetwork). The process is almost the same as for SNNs: analyze two inputs and merge their features to predict the final result. (Zhou et al. 2018) propose a two-way analysis method based on noisy and original images. (Amerini, Uricchio, Ballan, et al. 2017) follow this approach with two extractors that extract features from spatial and frequency domain respectively. (H.-G. Kim et al. 2018) also use this architecture with a regular CNN and a Markov network that processes the DCT coefficients. (Bondi, Lameri, et al. 2017) propose a method based on two streams: 1) a regular CNN, which provides a label for the input image; 2) another that gives a confidence score (i.e., how much we can trust the result). A mask is estimated from both outputs. For this type of double-stream architecture, the role of the classification part is to merge the feature maps, with bilinear pooling or with FC layers.

Reference	**Input**	**Features extraction**	**Classification**	**Database**
Camera Recognition				
(Mayer and Stamm 2018)	256×256	SNN	Similarity	*Dresden*, private dataset
(Mandelli et al. 2020)	80×80	SNN	Similarity	*Dresden, Vision*
(Mayer and Stamm 2020)	256×256	SNN	Similarity	*Dresden*, private dataset
Forgery Detection				
(Amerini, Uricchio, Ballan, et al. 2017)	64×64	Spatial and frequency CNN	FC layers	*UCID*
(Bondi, Lameri, et al. 2017)	64×64	CNN and Confidence network	Mask Estimation	*Dresden*
(Zhou et al. 2018)	Image	RGB and Noise CNNs	Bilinear pooling	*Casias, Columbia, Coverage, NC16*
(Y. Wu et al. 2018a)	256×256	Similarity and Manipulation	Fusion	*Casias, CoFoMoD*
(Rao et al. 2020)	128×128	SNN and CNN	Fusion + SVM	*Casias, Carvalho, Columbia gray*
(H.-G. Kim et al. 2018)	64×64	CNN and Markov	FC layers	*BOSSbase, Dresden, Raise*
(Mazumdar et al. 2018)	64×64	SNN	Distance layer	*Dresden*
(Cozzolino and Verdoliva 2018)	48×48	SNN	Distance Layer	*Dresden, SOCRatES, Vision*
(Cozzolino and Verdoliva 2020)	48×48	SNN	Distance Layer	*Dresden, SOCRatES, Vision*

Table 4: Architecture details of TSNs. Information on input size, extraction and classification part and databases used.

Whatever the technique used, this type of architecture is more complex than the CNN, but also more complete (flexible network) and can be very robust in some cases (SNN). Indeed, it either predicts the label of an image with a better accuracy (more features), or it classifies a pair of images according to their label (see Tab. 4). Moreover, in the case of the SNN, since it does not learn specific features (i.e., it is not dedicated to the detection of a particular forgery), it is more robust to any CF method. (Bappy et al. 2019) presents a TSN consisting of an Long-Short Term Memory (LSTM) and an encoder, followed by a fusion layer and a decoder. These particular networks are also used in DIF, although this is rarely the case.

2.2.4 Other Approaches

Auto-encoder

The auto-encoder is also a technique used in the fields of image processing. These networks are particularly dedicated to image compression. With the emergence of DL, new solutions have appeared for this particular task. Most of them are based on auto-encoders. This architecture consists of three parts: 1) the encoder that reduces the size of the input image; 2) the bottleneck that contains the compressed features; 3) and the decoder that reconstructs the image. However, these networks are also employed for DIF tasks. (Cozzolino and Verdoliva 2016) detail a method that first extracts features and then uses an auto-encoder to produce forgery detection maps. (Zhang et al. 2016) present a stacked auto-encoder to detect regions of tampering.

Transfer Learning

Other techniques are employed in DIF to obtain successful architectures. One of them is transfer learning, a subfield of machine learning. The principle is to build a network by copying some layers of a pre-trained model and randomly initializing the remaining layers of the target model. (Zhan et al. 2017) explain that there are two transfer methods: 1) one for the task (the most common transfer), which copies only the first few layers; 2) another for the database, which uses the deep layers. There is also a difference depending on the database used for the transfer. In the task transfer, the database comes from another domain, whereas the database is for the same application in the case of the database transfer.

In both cases, a reference model *M1* is used to transfer the weights ((Xu et al. 2016), here from steganalysis and a database *B1*). Fine-tuning of the target model *M2* (for the database *B2*) is necessary to adapt it to its new task. There are three alternative strategies, depending on the size and similarity of the two databases (*B1* of the pre-trained model and *B2* of the target model): 1) Full network training (large *B2* and different datasets); 2) Partial network training: A few layers (large *B2* and similar datasets) or several layers (small *B2* and different datasets); 3) Training only the classification part of the network (small *B2* and similar datasets).

(Al Banna et al. 2019) propose a method that applies full training to a CNN pre-trained on *ImageNet* (from ILSVRC). The main advantage of this technique is the speed of learning the network. Thanks to the already learned features, the network only needs to be refined and, thus, the training phase is really faster.

Architectures play a key role for DL-based methods to innovate and be more creative. However, behind this first aspect, the goal is to improve the performance for DIF tasks. For DNN, performances are established with the help of metrics and conducted through evaluations using dedicated databases. Moreover, it allows comparing methods, and the section 2.3 addresses these points.

Various architectures have been used by DL-based methods for DIF. CNNs are particularly effective with images, and have been widely used in other areas of image processing. As pre-processing modules are used to extract essential artifacts, enhanced features networks are also particularly exploited. Their purpose is to extend feature maps, using techniques such as shortcuts or dense connections. Similarly, TSNs aim to extend features by analyzing two different domains (i.e., one per stream), but can also provide robustness with SNNs. Other networks such as auto-encoder or transfer learning approaches can be applied.

2.3 Evaluation and Comparison

In many other areas of image processing, Deep Learning (DL)-based methods have outperformed classical ones. The main reason for this is the large amount of data (e.g., images in Digital Image Forensics (DIF)) that are used to train networks. Above all, the key point of DL-based methods for DIF tasks lies in performance improvement. In this section, we detail the databases that have been used in the literature. We also provide the main metrics to evaluate these different methods, and we even report some comparisons of their performances.

2.3.1 Databases

The database is an important element, especially in domains that deal with images, and even more for DL-based methods. In particular, DNNs are based on learning features from images. Their performances are improved according to the consistency between the desired and the real results. The outputs correspond to the labels that are associated with each image in the dataset. The classification is binary (e.g., forged or not) or multiple in some cases (e.g., cameras 1, 2, etc.). All of these images used for DL-based methods often come from public databases and sometimes from private datasets. Most of the databases presented in this subsection are dedicated to DIF. Therefore, the images are already labeled with the purpose of the application (i.e., cameras, manipulations, etc.). However, some datasets come from other domains such as steganalysis and are employed with modifications to be exploited. In this subsection, we detail the specifics and purpose of each database.

Camera Recognition Databases

The *Dresden Image Database* (Gloe and Böhme 2010) is perhaps the most popular in the field of DIF. It is mainly dedicated to camera recognition, but it is also used for forgery detection (with application of upstream manipulations). It is composed of more than 14, 000 images of various indoor and outdoor scenes that were captured by 73 cameras (27 different models) in order to perfectly establish their characteristics. *SOCRatES* (Galdi et al. 2019) is another database dedicated to the recognition of cameras, especially smartphones. It consists of 9, 700 images captured by 101 smartphones (62 different models). The images were captured by different people in order to obtain a heterogeneous database that represents real life scenarios well. The *Forchheim Image Database* (Hadwiger and Riess 2020) is quite similar to *SOCRatES* as it is dedicated to smartphones, with over 23, 000 images from 143 scenes by 27 devices (25 different models). Each image is provided in 6 qualities (original and from 5 social networks).

(Society 2018) hosted a challenge for camera recognition (with the *Kaggle* database). The training dataset consists of 2, 750 images of arbitrary scenes from 10 cameras. The test dataset contains original images, as well as recompressed images with random JPEG quality, rescaling or gamma correction. *Vision* (Shullani et al. 2017) was first released for video authentication, but can also be used for camera recognition. It contains 34, 427 images and 1, 914 videos from 35 portable devices. For these databases, the images are mainly in JPEG format (see Tab. 5), in contrast to the databases dedicated to forgery detection. Indeed, whether for copy-move or splicing, these databases contain uncompressed (i.e., Tagged Image File Format (TIFF), Nikon Electronic File (NEF), BitMaP (BMP), etc.), lossless (i.e., Portable Network Graphics (PNG)) or lossy (i.e., JPEG) compressed images.

Camera Recognition				
Database	**Reference**	**Size**	**Format**	**Composition**
Dresden	(Gloe and Böhme 2010)	3039 × 2014 - 3900 × 2616	JPEG, NEF	14K
Vision	(Shullani et al. 2017)	960 × 720 - 5248 × 3696	JPEG	34K
Kaggle Camera	(Society 2018)	1520 × 2688 - 4160 × 3120	JPEG	2750
SOCRatES	(Galdi et al. 2019)	640 × 680 - 5344 × 3006	JPEG	9,700
Forchheim	(Hadwiger and Riess 2020)	960 × 540 - 4608 × 3456	JPEG	23K

Table 5: Databases for camera recognition, sorted according to their released date. Information of size, format and composition are given.

Copy-Move Databases

The detection of falsifications is the other main task of DIF. Thus, it also exists several databases to this purpose and most of them are especially dedicated to copy-move (see Tab. 6): *Coverage* (Wen et al. 2016), *FAU* (Christlein et al. 2012), *GRIP* (Cozzolino et al. 2015), *CMH* (Silva et al. 2015) or *CoMoFoD* (Tralic et al. 2013). Some of them like *CoMoFoD* are more complete because several post-processing operations (JPEG compression, Additive White Gaussian Noise (AWGN), Gaussian Blurring (GB), etc.) are applied to the images. Applying manipulations on a falsified image is double-edged because it can mask the falsification, but also add other artifacts to the image. A series of databases dedicated to copy-move, named *MICC* have been proposed with a progressive number of images: *MICC-F220*, *MICC-F600* (Amerini et al. 2013) and *MICC-F2000*, which has the largest number of falsified images (i.e., 1300). For more information, (Amerini et al. 2011) presents a review of these databases dedicated to copy-move.

Copy-move					
Database	**Reference**	**Size**	**Format**	**Composition**	**Mask**
MICC-F220	(Amerini et al. 2011)	722 × 480 - 800 × 600	JPEG	110/110	No
MICC-F2000	(Amerini et al. 2011)	2048 × 1536	JPEG	1,300/700	No
FAU	(Christlein et al. 2012)	2362 × 1581 - 3888 × 2592	PNG, JPG	68/69	No
MICC-F600	(Amerini et al. 2013)	800 × 532 - 3888 × 2592	JPEG	448/152	No
CoMoFoD	(Tralic et al. 2013)	512 × 512 - 3000 × 2000	PNG	260/260	Yes
GRIP	(Cozzolino et al. 2015)	768 × 1024	PNG	80/80	No
CMH	(Silva et al. 2015)	845 × 634 - 1296 × 972	PNG	108/108	Yes
Coverage	(Wen et al. 2016)	400 × 486	TIFF	100/100	Yes

Table 6: Databases specialized for copy-move, with information about ground-truth masks.

Splicing Databases

Splicing is the other main falsification that has dedicated databases. However, there are fewer databases than for copy-move (see Tab. 7).

Database	**Reference**	**Size**	**Format**	**Composition**	**Mask**
Splicing					
Columbia Gray	(Ng and Chang 2004)	128 × 128	BMP	933/912	Yes
Columbia Color	(Hsu and Chang 2006)	757 × 568 - 1152 × 768	TIFF	183/180	Yes
Carvalho	(de Carvalho et al. 2013)	2048 × 1536	PNG	100/100	Yes
Casia v1	(Dong et al. 2013)	384 × 256	JPEG	800/925	Yes

Table 7: Databases specialized for splicing, with information about ground-truth masks.

There are two versions of the *Columbia database*: *color* (Hsu and Chang 2006) and *gray* (Ng and Chang 2004). The gray dataset contains 912 falsified patches (128 × 128) while the color dataset is composed of 183 images of different sizes. The falsifications are quite old and were done in a rough way. Thus, they can be easily visualized and detected by humans. *Carvalho* (de Carvalho et al. 2013) is a newer and more realistic database that also deals with splicing, with 100 of high resolution images (size 2048 × 1536). Another database, more realistic and with more images, has been released: the *Casia v1* (Dong et al. 2013), which includes indoor and outdoor scenes of everyday life. The first version contains 925 of images of size 384 × 256 in JPEG format.

Multi-Forgeries Databases

Casia v2 (Dong et al. 2013) contains 5, 123 forged, copy-move and splicing, of various sizes and formats. This database is quite realistic, and some post-processing operations have been applied to cover the traces of forgeries. The *Realistic Tampered Dataset (RTD)* is also dedicated to realistic tampering, with 220 splicing and copy-move.

Other datasets have been created to meet specific needs, such as *VIPP* (Bianchi and Piva 2012b). This dataset contains splicing to evaluate the detection of Double JPEG-Compression (DJPEG-C). On the other hand, *Wild Web* (Zampoglou et al. 2015) is devoted to real cases with a collection of images from the Internet. There is no guaranteed information on forgeries, but different versions of the images are provided to get the basic truths. Other databases are also comprehensive, as they deal with various forgeries and have many images (see the tab 8).

Database	Reference	Size	Format	Composition	Mask
Multi-Forgeries					
VIPP	(Bianchi and Piva 2012b)	300 × 300 - 3456 × 5184	TIFF	68/69	Yes
Casia v2	(Dong et al. 2013)	240 × 160 - 900 × 600	Raw, JPEG	7,491/5,123	Yes
WildWeb	(Zampoglou et al. 2015)	72 × 45 - 3000 × 2222	BMP, PNG, JPEG	90/9,657	No
RTD	(Korus and Huang 2017)	1920 × 1080	TIFF	220/220	Yes
NIST NC16	(Guan et al. 2019)	500 × 500 - 5616 × 3744	JPEG	1,124/564	No
NIST NC17	(Guan et al. 2019)	60 × 120 - 8000 × 5320	Raw, PNG, JPEG	4,077/1,410	No
NIST MFC18	(Guan et al. 2019)	128 × 104 - 7952 × 5304	Raw, PNG, JPEG	14,156/3,265	No
NIST MFC19	(NIST 2019)	160 × 120 - 2624 × 1968	Raw, PNG, JPEG	10,279/5,750	No
DEFACTO	(Mahfoudi et al. 2019)	240 × 320 - 640 × 640	TIFF	-/229K	Yes
NIST MFC20	(Fiscus et al. 2020)		Raw, PNG, JPEG	17K	No

Table 8: Databases dedicated to multi-forgeries, sorted according to released date. Information on size, format, composition and ground-truth mask are given.

The National Institute of Standards and Technology (NIST) organized an annual challenge called Media Forensics Challenge (MFC) that used different databases each year. First, they created a pilot dataset, *NIST16* (Guan et al. 2019), which contains 564 of forged images. Each image is spliced with four separate operations (low/high qualities of JPEG compression and with/without post-processing on falsifications) to evaluate their impact on performance. This process was abandoned for the following challenges (*NC17*, *MFC18-20* (NIST 2019), (Fiscus et al. 2020)). However, they have been built for multiple forensic applications, such as forgery, manipulation or deepfake detection. As a result, these datasets consist of various types of manipulation (blurring, intensity normalization, etc.) and falsification (copying, splicing, etc.). In addition, falsified images may contain multiple falsifications, making them difficult to detect. In summary, NIST has provided several databases that contain various manipulations and falsifications.

DEFACTO (Mahfoudi et al. 2019) is another database dedicated to various falsifications. It includes more than 200, 000 images, which are indoor and outdoor scenes or also faces, with a set of information about the location of the falsification (annotations) and the alteration process. These images were generated by applying four categories of tampering on the *MS COCO database* (T.-Y. Lin et al. 2014): 1) splicing $(105,000)$; 2) face morphing $(80,000)$; 3) object removal $(25,000)$; 4) copy-move $(19,000)$. The databases presented so far are dedicated to DIF tasks, but the way *DEFACTO* has been built shows that other types of datasets can be used if falsifications are applied to them.

Other Tasks Databases

The *MS COCO database* deals with object detection and was used to create *DEFACTO*. *MS COCO* was created from complex images of everyday scenes in their natural context. Each image is labeled with item segmentation to facilitate object location applications. It contains 91 features with a total of 2.5 million labeled instances in $328,000$ images. *Scene UNderstanding (SUN)* (Xiao et al. 2010) is also dedicated to object detection by categorizing scenes, and has been exploited to produce forgeries for DIF applications. It consists of 899 categories from $130,519$ images. Usually, these databases are exploited to create a synthetic dataset of forged images through the accurate segmentation of their components.

Steganalysis, which is another important topic in forensics, has provided useful databases for DIF (see Tab. 9). *Bossbase* (Sedighi et al. 2016), frequently used in steganography, contains $10K$ of images (512×512) captured by 7 cameras in the format Portable GrayMap (PGM). There are very smooth, unclear and very dark images, which shows the diversity of this dataset and thus its availability as a standardized source. *Alaska#2* (Cogranne et al. 2019) represents another steganalysis challenge to provide a large and diverse dataset. Indeed, it is composed of $80K$ of images from more than 40 cameras (smartphones, cameras, tablets, etc.) that have been processed realistically. Moreover, this dataset is mainly designed

Database	**Reference**	**Size**	**Format**	**Composition**
Object Detection				
SUN	(Xiao et al. 2010)	200 × 200	JPEG	130K
MS COCO	(T.-Y. Lin et al. 2014)	-	JPEG	330K
Steganalysis				
BossBase	(Sedighi et al. 2016)	512 × 512	PGM	10K
Alaska#2	(Cogranne et al. 2019)	256 × 256 and 512 × 512	RAW	80K
Real World Images				
NRCS	(Macdonald 2004)	1500 × 2100	TIFF, JPEG	11K
UCID	(Schaefer and Stich 2004)	512 × 384 and 384 × 512	TIFF	1338
Raise	(Dang-Nguyen et al. 2015)	4288 × 2848	TIFF, NEF	8,156

Table 9: Databases from other domains that are used in DIF.

for DL-based applications with the huge amount of data that can be exploited and the heterogeneity of images that is a challenge.

There are other databases more focused on real-world scenes and certain raw image formats. The *Raise* (Dang-Nguyen et al. 2015) database contains both TIFF and NEF images. It consists of $8,156$ uncompressed and unprocessed images, captured by 3 different cameras. These images represent various indoor and outdoor scenes in more than 80 locations in Europe. TIFF is also in *Uncompressed Color Image Database (UCID)* (Schaefer and Stich 2004), which was built for content-based image search. As the name implies, this database consists of $1,338$ uncompressed images of outdoor and indoor scenes of different locations (e.g., natural places, objects, etc.). Natural Resources Conservation Service (NRCS) (Macdonald 2004) is also dedicated to TIFF images.

Finally, the mentioned databases have been exploited in AI-based methods, and in particular *Dresden* and *Casia* have been widely used. A database is essential to compare the models with each other, especially since 2015, as they are almost all based on DL. Thus, the methods can be compared based on their architecture and databases. To perform this comparison, methods must be evaluated using dedicated metrics.

2.3.2 Evaluation metrics

Evaluation is mandatory to assess the effectiveness of a method, especially those based on DL. Databases are used to train the methods, but are also necessary to perform the evaluation. Therefore, the databases are divided into three data sets: training - validation (during training) and testing. Experimental tests are calculated using metrics to provide the performance of a method. These metrics are the baseline for judging the quality of the results provided by a method.

Confusion Matrix

There are several scoring metrics, all of which are calculated from the confusion matrix (see Tab. 10), which counts and classifies the different results. Indeed, the results, which are obtained by DL-based methods, are expressed in terms of four values (for example, in the case of forgery detection): 1) the True Positive (TP)

		Predicted Class	
		Forged	Original
Actual	Forged	TP	FN
Class	Original	FP	TN

Table 10: Confusion matrix for a binary example in detection of manipulation

for images correctly detected as forged; 2) the False Negative (FN) for images incorrectly detected as original; 3) the False Positive (FP) for images incorrectly classified as forged; 4) and finally the True Negative (TN) for images correctly classified as original. Several elements, which are exploited for metrics, can be defined from this confusion matrix.

The first elements, defined as True Positive Rate (TPR) and False Positive Rate (FPR), give an indication of the number of positive predictions (here forged) with respect to the set of results. They correspond to the probability of predictions ($P \in [0,1]$) classified as forged with respect to the set of data labeled as original (for FP and TN) and as forged (for TP and FN) (see Eq. 6 and 7).

$$TPR = \frac{TP}{TP + FN} \quad (6) \qquad \text{and} \qquad FPR = \frac{FP}{FP + TN} \quad (7)$$

Three other elements also derive from the confusion matrix: 1) *sensitivity* (also called *recall*), which is identical to TPR (see Eq. 6); 2) *specificity*, which is the rate of well-predicted negative results (see Eq. 8); 3) *precision*, which is the probability of well-classified predictions as positive based on all positive results (see Eq. 9). In fact, *specificity* is inversely proportional to *sensitivity* (i.e., if one increases, the other decreases) while TPR and FPR are proportional (i.e., if one increases, the other increases).

$$specificity = \frac{TN}{TN + FP} \quad (8) \qquad \text{and} \qquad precision = \frac{TP}{TP + FP} \quad (9)$$

Common Metrics

Some measures are based on the elements of the confusion matrix. The best known evaluation measure is the *accuracy* ($Acc \in [0,1]$), which is used in almost all camera recognition literature and evaluates the probability of a good model prediction (see Eq. 10).

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

The F_1 *score* is more robust than the *accuracy* because it takes into account both parts of the confusion matrix. It is computed from *recall* and *precision* (Eq. 11), which are already elements computed from the elements of the confusion matrix.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{11}$$

Thus, F_1 *score* is more precision than *accuracy.* Indeed, F_1 *score* represents the probability of positive elements according to the prediction and the labeled sets.

Precision indicates the proportion of predicted positive elements that are relevant (i.e. well-classified) while *recall* (TPR) gives the number of labeled positive elements that are well ranked.

The other metrics are based on the rates resulting from the confusion matrix. The Receiver Operating Characteristic (ROC) (Huang and Ling 2005) can be calculated from TPR and FPR. This is a probability curve generally used in classification problems to measure the model performance. The graph is obtained by plotting the TPR against the FPR (see Fig. 6). The *Area under the Curve (AuC)* is a scoring metric associated with the ROC that indicates how well the model is able to distinguish classes. Thus, the higher the *AuC*, the better the performance of the model.

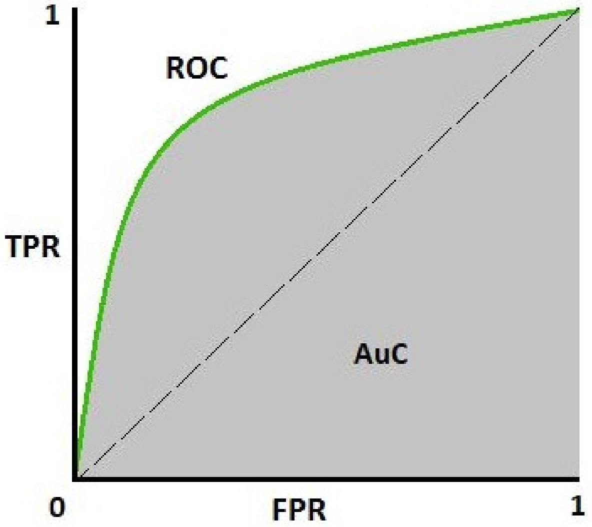


Figure 6: Graphic of the ROC. (green line) ROC (gray area) AuC

Finally, there are several ways to evaluate the performance of a method, and all the measures presented here have been used in SOTA publications. However, the main techniques for comparing methods are the *accuracy* and the F_1 *score*. The F_1 *score* is widely used for forgery detection, as it evaluates how well the output mask matches the ground truth. In this case, *precision* and *recall* are computed on a pixel basis.

2.3.3 Performance Assessment

Performance evaluation is an essential task, especially for making comparisons. In this subsection, we present the results of some SOTA methods, applying a criterion that allows a fair analysis. For camera recognition, we chose the database (i.e., *Dresden*) as the common element. For forgery detection, we chose methods dedicated to manipulations (e.g., compression JPEG, resampling, etc.).

Camera Recognition

Camera recognition has been widely discussed in the literature. With the growth of articles, new ideas and ways to recognize cameras have emerged. To date, camera recognition has several ways of being applied, which prevents from comparing methods on an equal footing. The first problem that makes it difficult to compare methods is the diversity of protocols. On the one hand, this diversity is a valuable aspect, as it contributes to the improvement and enrichment of camera model identification. But, on the other hand, this variety of protocols reduces the possibilities to compare methods and to evaluate their performance on an equal footing. Several approaches exist for camera model identification, and three main applications can be particularly identified in the literature: basic classification, triple classification, and open-set classification. Among all these articles, we decided to focus our analysis only on the methods that work on *Dresden*. Indeed, *Dresden* is the most used database and is dedicated to camera recognition.

The basic scenario involves identifying the camera based on a label (brand, model, or device) and is the most specific and discussed application. Most methods for this scenario classify camera models (Bondi, Baroffio, et al. 2017). However, even among these methods that use the same evaluation protocol, there is still a problem with the number of cameras used in the experiments. (Tuama et al. 2016) conduct a series of experiments to highlight the data dependency problem by increasing the number of camera models in the dataset. The evaluation focuses on the classification of a baseline model in three experiments with an increasing number of camera models (12 - 14 - 33).

The triple classification is close to the basic scenario, as the goal is to identify the camera according to a label. But, it classifies cameras according to the brand, the model and the device. The triple classification aims to identify cameras based on all labels and is more global (Y. Chen et al. 2017; M. Zhao et al. 2020).

Open-set classification is particular and addresses the robustness problem. The objective of this approach is to evaluate the generalization of a method by classifying *unknown* cameras. In this case, the network has never learned their characteristics, which makes classification difficult. (Bondi, Baroffio, et al. 2017) approaches the *unknown* camera problem with an additional experiment. (Mayer and Stamm 2018) fully addresses the problem of the open-set scenario, with a method to classify image pairs as either *known* or *unknown* cameras. For this application, the evaluation was performed with a series of experiments based on three subsets according to an image pair: 1) only *known* camera models; 2) one *known* and one *unknown* model; 3) only *unknown* camera models. The evaluation protocol was performed with 65 camera models.

The aspect that stands out in the SOTA publications is the disparity of applications. Thus, the comparison is hard to index (see Tab. 11). Performance is significantly correlated with the number of cameras: the larger the database, the lower the performance. In addition, it is more difficult to authenticate cameras that belong to the same model, because their fingerprints are more similar and thus more difficult to identify. This phenomenon is well illustrated in papers dealing with triple classification. The performance is decreasing from one label to another, and is extremely low (less than 55%) for the devices. (Mandelli et al. 2020) present a method based on SNN, which obtains a higher score (88%) as it predicts whether two images come from the same device $(1 - to - 1)$. Additional experiments can be performed for camera recognition, such as the evaluation of robustness to manipulations (Ding et al. 2019; Mayer and Stamm 2020). Manipulations are also a full-blown application, mostly tackled in forgery detection.

References	**Brand**		**Model**		**Device**	
	Nbr. cameras	Accuracy	Nbr. cameras	Accuracy	Nbr. cameras	Accuracy
Basic Classification						
(Tuama et al. 2016)			12 14 33	98% 97.09% 91.9%		
(Bondi, Baroffio, et al. 2017)			18	92.83%		
(Bayar and Stamm 2017a)			26	98.58%		
(Rafi et al. 2019)			27	97.03%		
(Mandelli et al. 2020)					87[1]	88.0%
Triple Classification						
(Y. Chen et al. 2017)	13 19[1]	99.12% 97.73%	27	94.73%	74	45.81%
(Ding et al. 2019)	13	99.6%	27	97.2%	74	52.4%
(M. Zhao et al. 2020)	13	99.4%	27	96.1%	74	47.5%
Open-set Classification						
(Bayar and Stamm 2018b)			10 25	99.06% 97.74%		
(Mayer and Stamm 2018)			10	91.1%		
(Mayer and Stamm 2020)			25	94.0%		

Table 11: Performance of SOTA methods on *Dresden* for different applications of camera recognition. [1] indicates the use of another dataset in addition.

Forgery Detection

The goal of falsification detection is to identify whether an image has been modified by tampering. In particular, finding the location of modified areas is the task that has been most discussed in the literature. In fact, locating tampering in images is more difficult because it involves detection at the pixel-level. Several metrics are used for this type of classification, such as the F_1 *score* or *AuC* (see subsection

2.3.2). In addition, there are many databases that deal with forgeries, such as *Casias* or *UCID* (see subsection 2.3.1). This makes it quite difficult to make a comparison with fair criteria.

However, many manipulations can be employed after falsifications, either to conceal them or to enhance the image. These manipulations are applied for various purposes: i) to degrade the forensic traces with MF used as a denoising filter, Additive White Gaussian Noise (AWGN) or Gaussian Blurring (GB); ii) to improve the quality with Contrast Enhancement (CE); iii) to match the falsified area to the original image with resampling; iv) to store the falsified image with JPEG compression. We therefore based our analysis on the detection of these manipulations, as the performance of the SOTA methods is comparable.

All of these techniques required the adjustment of parameters such as the kernel size for MF or the standard deviation for GB. Most DL-based methods deal with parameters that are easier to classify. For example, MF is easier to detect with a kernel of size 5 than with a kernel of size 3, as it leaves more obvious artifacts in the image. (Cozzolino et al. 2017) address this particular classification problem, examining various parameters for each manipulation. The article highlights the correlation between difficult variables and decreased performance. However, it is a binary classification, which is not comparable to multiple classification as it provides better results.

Some methods from the literature deal with only one manipulation (binary classification) while others tackle the multi-classification problem. Multi-classification consists in predicting the label associated to an image (here the manipulation) according to a set of labels (different manipulations). Thus, this application is more difficult than the binary classification. (Bayar and Stamm 2018a) report performances for difficult parameters in the context of multi-classification. Moreover, this method is globally more efficient than the other techniques on easy parameters (see Tab. 12).

References	**Original**	**MF**	**Resampling**	**GB**	**AWGN**	**JPEG**	**CE**
(Tang et al. 2017)	92.43%	97.90%	96.34%	97.39%[1]	▨	97.93%	89.09%[1]
(Kim and Lee 2017)	90.92%	99.45%	95.98%	97.5%	99.48%	▨	▨
(Bayar and Stamm 2018a)	**99.49%**	**99.77%**	**99.51%**	99.46%	**99.98%**	**99.79%**	▨
(Mazumdar et al. 2018)	99.35%	99.64%	99.26%	**99.51%**	96.61%	▨	95.24%
(H.-G. Kim et al. 2018)	▨	98.10%	99.23%	91.8%	88.34%	▨	▨
(Cozzolino et al. 2017)	▨	99.75%	99.78%	96.56%	99.66%	94.54%	▨

Table 12: *Accuracy* of forgery detection for various manipulations. [1] outlines the use of challenging parameters.

(Bayar and Stamm 2018a) tackle a more realistic problem. Indeed, instead of classifying a single manipulation, they have developed a database with images that have been modified with a combination of two successive manipulations (i.e., MF-GB). The classification of such images is interesting because an attacker would have to modify an image with a forgery followed by various manipulations. Such operations can be applied to improve the quality of the image, store it or concealing artifacts. Among all these techniques, some are used to hide or degrade forensic traces that are crucial for forgery detection. The application of manipulations for this purpose corresponds to Counter-Forensics (CF), which is detailed in the following section 2.4.

> Evaluation of DL-based methods can be performed using metrics and databases. The main metrics used in the literature are the *accuracy* and the F_1 *score*, which are often associated with camera recognition and false document detection, respectively. The SOTA methods have also exploited various databases, from dedicated databases such as *Dresden* (camera recognition) or *Casia v2* (forgery detection) to those from other image processing domains. We also report the performance of methods for camera recognition on *Dresden*, and for manipulation detection, which is a subtask of forgery detection.

2.4 Anti- and Counter-Forensics

Counter-Forensics (CF) is a topic that emerged in parallel to Digital Image Forensics (DIF), aiming to improve the reliability of forensic methods by testing the robustness of models through attacks. These methods were designed in the context of classical methods to hide forgeries and deceive forgery detectors. Anti-Counter-Forensics (ACF) is the opposite domain, as it attempts to address the problem of attackers. The goal is to detect images that have been modified by CF methods. Since the advent of DNNs, the anti- and- counter-subdomains have also evolved for DL-based methods. (Barni, Stamm, et al. 2018) specifically address these subdomains in a general review. There are also dedicated analyses, such as the study of attacks against image tampering detectors (Gragnaniello et al. 2018) or against camera recognition (Marra et al. 2018). This section details the classical and DL approaches to these subdomains.

2.4.1 Classical Approaches

Compression traces

The main methods of CF are related to compression. Indeed, after falsification, the image is usually restored, leaving artifacts such as inconsistencies in the Discrete Cosine Transform (DCT) coefficients. For this reason, many CF measures aim to hide the traces of any compression. Various methods are used to achieve this by removing or masking different artifacts. (Stamm et al. 2010) attacks JPEG blocking artifacts by applying a MF and adding Additive White Gaussian Noise

(AWGN). (Sheng and Su 2014) propose to remove the traces left by the DCT quantization. These approaches are effective in fooling compression detectors, but they also leave other artifacts in the image. Indeed, Anti-Counter-Forensics (ACF) have studied the impact of such modifications on the intra- and inter-blocks of the JPEG compression. (Li et al. 2012) detail the detection of CF based on the correlations between the JPEG blocks, while (Singh and Singh 2019) analyze the co-occurrence matrix of these blocks. Other approaches are based on spatial noise (Valenzise et al. 2011), high frequency AC (Alternative Current) coefficients (Lai and Böhme 2011) or distortion caused by CF modifications (Valenzise et al. 2013). (Wang et al. 2014) discuss wavelet compression with a study of the Discrete Wavelet Transform (DWT) coefficients based on the Hough transform to detect inconsistencies related to compression. To be more robust to these detections, the CF methods have been improved by adding a denoising step (Valenzise et al. 2014) or by minimizing the distortion (Barni et al. 2016). Training a model with adversarial images is a solution to these improvements of CF methods (Barni, Nowroozi, et al. 2017). This method is very close to the basic ACF method for DNNs (see subsection 2.4.2).

Camera traces

Camera recognition is one of the main objectives of DIF along with forgery detection. Since compression traces are concerned by tampering, the other artifacts studied for the methods are the camera fingerprints. (Karaküçük and Dirik 2015; Dirik and Karaküçük 2014) detail a method to attack camera recognition through the Photo-Response Non-Uniformity (PRNU). The objective is to remove the camera fingerprints with a factor that depends on the images and their content. This factor is therefore accurately estimated by an iterative search based on a denoising filter (Wiener or wavelet). (Sameer et al. 2019) present an ACF method to manage this adaptive search based on a denoising filter by using the local binary pattern to authenticate a camera.

(Raj and Sankar 2019) present an attack that focuses on the PRNU, which is specific to each camera and allows recognizing them. The method replaces the fingerprints of the camera with the PRNU of another camera. In fact, the PRNU contained in an image is removed by DWT. Then, a new fingerprint is calculated from a set of images coming from another camera, and added to an image considered as false. (Goljan et al. 2011) propose a method ACF to treat this problem: the triangle test. Let I be the image to be tested, J the images supposed to have been exploited for falsification and K the sure images. A conclusion can be drawn by calculating the PRNU of each and examining the correlation between them: if the correlation is higher between I and J than between I and K, then image I has been used for falsification. (Marra et al. 2014) attack this method by proposing a compromise between the triangle test and the camera recognition to fool both the correlation comparison and the camera recognition.

Others

Many other artifacts are attacked to hide deception from detectors. There is a race between ACF and CF, as when one method of attack is proposed, another to detect it is developed. (Kirchner and Bohme 2008) detail a double attack based on a MF and geometrical distortions respectively for the low- and high-frequency components. The objective is to hide the resampling traces, which are usually characterized by periodic dependencies. To cope with this method CF, a periodicity-based detection is first applied to discriminate the images that have been attacked (Peng et al. 2015). This process is based on the analysis of the partial auto-correlation with a threshold to decide whether an image is suspicious or not.

MF is also an enhancement tool often applied after falsification, as it is a noise removal filter. Therefore, hiding the application of the MF is a powerful subject. (Z. Wu et al. 2013) explain that the distributions of pixel differences are not the same between the forged and the original images. They proposed a method to falsify the distribution of the forged images in order to reproduce the original images by including noise. The detection of CF methods based on MF has been addressed by two approaches. In the first one, (Zeng et al. 2014) explain that the histogram of the horizontal difference of the pixels of the different rows has a certain periodicity in the case of a falsified image. Therefore, they applied a peak detection method to distinguish images that have been exposed to a CF method. The second method (Kang et al. 2015) exposes artifacts in the histogram distribution of Median Filtering Residual (MFR) for textured areas. In the case of an original image, the central bin is much higher than its neighbors, which is not the case for forged images.

(Costanzo et al. 2014) detail an attack and an ACF method on the analysis of key-point suppression and injection Scale-Invariant Feature Transform (SIFT). They proposed two key-point suppression detectors: 1) one based on the ratio of key-points to corners; 2) another based on histograms of key-points in image blocks of different variance levels. The injection of key-points in the image is used to fill this weakness and completed according to the distribution of the key-points. Other methods of ACF are detailed, such as Contrast Enhancement (CE) detection based on the gray level co-occurrence matrix (De Rosa et al. 2015) or SVM detector attack for global manipulations (Z. Chen et al. 2017).

2.4.2 Deep Learning Methods

As far as classical methods are concerned, CF in DL addresses camera recognition and other forensic features (see Tab. 13). However, one particular aspect is related to DNN. There are two ways to apply an attack, known as the white- and black-box cases. The major difference comes from the knowledge of the detector to be deceived. In the white-box case, the network being fooled is known by the attackers (weights, layers, etc.) and they exploit its information to compute the method. The black-box case is more complicated because the attackers have no clue about the method, even though they can use it in most cases (give an input to the network and collect the output). On the other hand, the white-box scenario is easier for attackers to handle and therefore more difficult for defenders. The black-box case represents the exact opposite.

Counter-forensic			
References	**Purpose**	**Scenario**	**Attack**
(C. Chen et al. 2018)	Camera recognition	White-box	GAN
(C. Chen et al. 2019)	Camera recognition	Black-box	GAN
(Güera et al. 2017)	Camera recognition	White-box	FGSM / JSMA
(D. Kim et al. 2018)	MF	Black-box	GAN
(Mehrish et al. 2019)	CE	Black-box	Adaptive CE
(W. Zhao et al. 2019)	Recaptured images	Black-box	Cycle-GAN
(Barni, Kallas, et al. 2018)	Adversarial transfer	White-box	FGSM / JSMA

Anti-counter-forensic		
References	**Purpose**	**Method**
(Tariang et al. 2019)	MF	MFR + residual dense network
(Carrara et al. 2017)	Adversarial images (FGSM/ L-BFGS)	CNN + k-NN
(Carrara et al. 2019)	Adversarial images (FGSM/ L-BFGS)	CNN + k-NN
(Schöttle et al. 2018)	Adversarial images (PGD)	Linear filter

Table 13: Details of counter and anti-forensic methods. Scenario and attack are given for anti-forensic, whereas the method is outlined for counter-anti-forensic.

White-box attacks

White-box attacks concern all DIF tasks, and in particular the recognition of cameras. One attack is mainly exploited by the CF methods: the misclassification. The goal of such an attack is to direct the network towards a wrong label.

(Güera et al. 2017) present two distinct methods that mislabel an image in the context of attacks against camera recognition. The first is the Fast Gradient Sign Method (FGSM) (Goodfellow et al. 2015), which generates adversarial examples based on the loss function to emphasize their classification. As a result, these images are systematically misclassified by the network. The second method is the Jacobian based Salience Map Attack (JSMA) (Papernot et al. 2015), which aims to modify the input features that most significantly change the output of the network. Both methods require the information of the model to be applied, as they exploit the loss function and the input characteristics.

(Barni, Kallas, et al. 2018) also address both approaches in a study of transfer for adversarial examples. They proposed two tasks: 1) the database transfer, where they keep the same network with the same attack but with a different database;2) the network transfer, where they keep the same database with the same attack but with a different network. Their analysis proves that the adversarial examples cannot be used in another context (different attack, different database, etc.).

(Carrara et al. 2017) present two methods (with a more elaborate version (Carrara et al. 2019)) for detecting adversarial images in the context of a white-box scenario. Their method combines a regular Convolutional Neural Network (CNN) with a k-Nearest Neighbors (k-NN) that respectively predicts a class and classifies the image based on the deep features (of the CNN). They used two white-box approaches to generate the adversarial images: FGSM and box-constrained L-BFGS (Szegedy et al. 2014). (Schöttle et al. 2018) detail another ACF method dedicated to adversarial images. They used the method Projected Gradient Descent (PGD) (Madry et al. 2019), a variant of FGSM, to generate attack images. The detection method is based on a threshold. Indeed, the pixels of an original image can be estimated from the values of its neighbors. Based on this assumption, they apply a linear filter, compute the average of the differences between the pixels and compare it to a predefined threshold.

(C. Chen et al. 2018) propose a Generative Adversarial Networks (GAN) (Goodfellow et al. 2014) that attacks CFA artifacts to fool camera recognition. They use the attacked CNN as a discriminator to update the generator through classification loss. GANs are widely used for black box attacks because network information is not needed.

Black-box attacks

The main difference with the white-box comes from the discriminator, because the attackers have no knowledge of the detector. Therefore, the generator cannot be updated with the classification loss. (C. Chen et al. 2019) propose to include a substitution network, to manage it by taking as inputs the original image and the output label of the attacked CNN (obtained by a query). (D. Kim et al. 2018) detail a GAN that restores MF images as original images. The generator produces restored images from the MF image and the discriminator gives it its label (i.e., original or restored) and updates the generator to improve the forgery.

(Tariang et al. 2019) detail an ACF method for MF. They proposed a dense residual network with a MFR layer to extract residual forensic features. (Mehrish et al. 2019) present a method based on spatial and DCT domains to fool detectors of CE. Their method is based on various distributions to generate adaptive CE images. (W. Zhao et al. 2019) detail a cyclic framework consisting of two generators and two discriminators to attack a recaptured image detector. The two GANs are processed in parallel, with one processing original and false-recaptured images, while the other processes the recaptured and false-original images. The main advantage of this cycle-GAN architecture comes from the loss function, which contains the adversarial loss (like each GAN) but also the cycle consistency loss (thanks to the cycle architecture).

Counter-Forensics (CF) is a sub-theme of DIF, which aims to attack DIF methods. Classical attacks are mainly based on compression and camera traces, while DL-based methods exploit misclassification techniques or GAN. Anti-Counter-Forensics (ACF) is the adversary, as the goal is to detect these attacks.

2.5 Perspectives and Issues

The analysis of the literature shows the difference with other fields of image processing, as well as its evolution from classical toDeep Learning (DL)-based methods. We have detailed their variety in the last sections through their architectures, their preprocessing modules and the databases used. In conclusion to this study, this section highlights questions raised by the literature. In particular, we present the problem of camera recognition protocols and the perspective with the future Artificial Intelligence (AI)-based compression standard: JPEG-AI.

2.5.1 Issue of Camera Recognition Protocols

In the case of camera recognition, the study of the literature highlights the difficulty of comparison, despite the numerous methods proposed. Taking the *Dresden* database as a comparison criterion, the analysis is mainly limited to one task (application, label and recognition type) and the same number of cameras. On the one hand, task diversity is important because it broadens the scope of camera recognition. On the other hand, it makes it impossible to compare methods that have been used for different tasks. For the same task and with the use of the same database (i.e. *Dresden*), the number of cameras varies too much to make a fair comparison.

The first aspect is associated with the variety of camera recognition. However, there is also another problem endemic to this topic: the variety of databases. In fact, among all DL-based methods, few papers use more than one database (e.g., *Dresden*) and the second is often a private dataset. This evaluation is in contrast to other areas of image processing, which have conducted their evaluation on different databases.

2.5.2 Perspective of AI-based Compression

It is interesting to note that some architectures take the whole image as input, whereas usually the DNN is trained with patches to avoid resizing, which removes forensic traces. Removing forensic traces is the goal of Counter-Forensics (CF) methods, which are particularly dedicated to compression and camera traces. Notably, traces of double compression are exploited for forgery detection. Recently, the JPEG organization launched an analysis of AI-based solutions for compression. This study is part of a new compression standard that will be released in the next few years: JPEG-AI.

As explained in the subsection 2.3.3, manipulations can affect the traces used in DIF. These manipulations are exploited to assess the robustness of DL-based methods. In particular, JPEG is one of the main manipulations studied. AI-based compression should also be taken into account as an attack to remove or hide forensic artifacts. Moreover, when images are compressed with DL-based methods, especially auto-encoder or GAN, the traces could be degraded until decreasing detection performances.

These two issues are handled in this manuscript with dedicated chapters. In particular, we propose progressive protocols to address camera recognition problems in the chapter 3. In the chapter 4, we introduce AI-based compression and present its impact on two DIF tasks: forgery detection and Social Network (SN) recognition.

The literature review highlighted two main problems that we address in the following chapters: 1) the difficulty of comparison for camera recognition methods; 2) manipulations can hide forensic traces, and AI-based compression, with fairly recent solutions, must be considered. At the same time, video forensics has started to develop recently and could be the next perspectives for DIF.

3 Progressive Protocols to Address Issues of Source Camera Recognition

3.1 Source Camera Recognition

At the beginning of the century, with the development of digital devices (cameras, cell phones, etc.), access to images and videos increased to the point of becoming an important communication channel. At the same time, the modification of digital images became easier thanks to free and accessible image editors. These modifications can be applied to improve the quality of an image, but they can sometimes be malicious. In some applications such as court cases or police investigations, images are crucial evidence and their authenticity must be proven. Thus, digital image retouching is a key issue, especially for proving the authenticity of an image. At the same time, the recognition of the source camera, a field of Digital Image Forensics (DIF) (Redi et al. 2011), has proven to be a solution for the detection of such falsifications.

3.1.1 Introduction

Camera Fingerprints

Camera recognition was first approached with classical methods based on the artifacts of the digital image creation pipeline. The set of these artifacts is often called the camera fingerprint, similar to the human fingerprint, which is used to identify a person. The camera fingerprint is composed of different elements, such as the features created by the Color Filter Array (CFA) (Long and Huang 2006; Celiktutan et al. 2006), or the chromatic aberration, due to the imperfections of the lens (K. S. Choi et al. 2006b; Van et al. 2007). Other important components of the camera fingerprint are the Sensor Pattern Noise (SPN) (Li 2010; Mahdian and Saic 2009) and the Photo-Response Non-Uniformity (PRNU) (Filler et al. 2008; Lukas et al. 2006; M. Chen et al. 2008). Notably, the PRNU is due to the imperfections of the silicon wafer during the manufacture of the sensor. These imperfections result in different pixel sensitivity to light, generating a distinctive pattern unique to each camera, refered as the digital camera fingerprint. Finally, the traces resulting from image enhancement (Tsai and Wu 2006; Kharrazi et al. 2004) or JPEG quantization (Farid 2006), are also used for camera recognition.

Classification Levels

In particular, the SPN or the PRNU first allowed to establish the camera fingerprint. Then, with the democratization of DL, the performances were improved notably thanks to Deep Neural Network (DNN). This architecture is made of three steps (see subsection 1.2). First, the CNN takes as input an image. Then, the image is analyzed by the feature extractor, which generates an output map. Finally, these output maps are used in the classification part. The purpose is to assign a class to the studied image. There are especially different levels of image classification depending on the camera. In this domain, cameras are defined by three characteristics: the brand (e.g., *Nikon*), the model (e.g., *Nikon D70*), and the camera itself, called the device (e.g., *Nikon D70 by Bob*). Thus, the recognition of the source camera can be done according to three levels of classification: the brand, the model and the device. To this end, several techniques have been developed, using the artifacts left during the acquisition of a digital image (Farid 2009b). Moreover, there are diverse recognition approaches.

3.1.2 Recognition Approaches

Among these DL methods, there are three different applications of recognition: i) basic - this task is the most common, as it is also the first to be performed. The goal is to recognize the source camera according to a classification level. Most of the work in the literature focuses on the camera model. ii) triple - as the name implies, the goal is to provide a prediction for all three classification levels. iii) open-set - with recognition of both *known* and *unknown* cameras. For this task, the cameras are distinguished between those used during the training of the network, called *known*, and the others called *unknown* which are only used for testing. The objective is to evaluate the ability of a model to generalize its performance to other cameras (i.e. other databases) and to prove its robustness.

In image processing domains that involve recognition, such as face recognition, there are two main approaches: identification ($1-to-N$) and verification ($1-to-1$). The principle of the first is to identify, among a group of N cameras, the one associated with the studied image. For the verification, two images are confronted together, in order to check if they come from the same camera or not.

3.1.3 Literature Issues

The analysis of the SOTA described in the chapter 2 has highlighted some problems. Indeed, the most obvious one is the difficulty of comparison between source camera recognition methods. This problem comes from the lack of databases to evaluate the performances of these methods, but also from the diversity of protocols and applications. Moreover, recognition is rarely performed on a per-verification basis, let alone on a per-device basis, which is the most difficult level of classification. Therefore, our work in the area of camera recognition has focused on establishing protocols to solve these problems.

In a first time, we try to compare DL architectures and their robustness with a protocol based on transfer learning. The purpose is to evaluate the ability of networks to generalize their performance for various cameras. Thus, we focus our protocol on the open-set scenario. Based on the promising results, we expand our study to multiple databases, as it is usually done in other image processing fields. For this second protocol, we focus on the basic classification. Finally, to address the most difficult task, which is the recognition of camera according to the device, we propose a more reliable protocol based on verification. The following subsections are detailing these protocols and the associated experiments.

> Camera recognition is performed based on three levels: brand, model and device. Then, there are three main tasks: basic classification (one level), triple classification (three levels) and open-set classification (*unknown* cameras). Finally, recognition can be applied by identification $(1-to-N)$ or verification $(1-to-1)$. We propose three successive protocols that address the problems in the literature regarding camera recognition: 1) the robustness comparison with transfer learning; 2) the lack of databases; 3) the most difficult label: camera recognition with a verification protocol.

3.2 Robustness Study via Transfer Learning

Source camera recognition can be done according to the brand, the model or the digital device itself. The identification of the camera model is the most studied application in the literature. The principle is to recognize, among a group of camera models, the one associated with the studied image. Several techniques have been developed, using the artifacts left during the acquisition of a digital image. Then, with the democratization of DL, the performances were improved, in particular thanks to the CNN. These networks first analyze the artifacts and reduce them via a feature extractor, while identification is performed by classification layers.

However, even though these methods have proven to be effective in identifying camera models, there is still a strong dependence on the data. In fact, each digital camera has its own artifacts: the camera fingerprint. The camera fingerprint depends notably on the PRNU, which is linked to physical inconsistencies. Thus, cameras of the same brand have close digital fingerprints. The uniqueness of the artifacts and the similarity of the fingerprints raise the question of performance robustness. In this section, we exploit transfer learning to conduct a comparative study on the robustness of DL-based methods for camera model identification. Furthermore, this study is conducted with three well-known CNN architectures as well as with the two main applications of source camera recognition: basic and open-set classifications.

This section 3.2 presents the protocol based on transfer learning through different aspects. The following subsections explain the motivations of our comparative study (subsection 3.2.1) and detail the architectures used as well as our protocol (subsection 3.2.2).

3.2.1 Problem of Performance Generalization

The analysis of the State-Of-The-Art (SOTA) has raised some questions. In particular, the problem of performance generalization, which is addressed by open-set classification. In this case, the evaluation is performed with *unknown* cameras, which were not used during training, in order to generalize the performance. This objective is critical due to the problem of similar digital fingerprints, and we propose to use transfer learning to evaluate the generalizability of DL-based methods.

Similar Digital Fingerprints

The identification of camera models is the most discussed topic in the literature. Among these papers, the problem of similar digital fingerprints has been highlighted. (Tuama et al. 2016) tackle this issue through a series of three experiments. The method is based on a CNN with a High-Pass Filter (HPF) used as a pre-processing module. 1) The network was first evaluated with 12 cameras from the *Dresden Image Database*. 2) Then, with 2 additional cameras of the same brand to highlight the digital fingerprint similarity problem. 3) Finally, with all cameras (33 models: 27 from Dresden + 6 private) to generalize this phenomenon. (Bondi, Baroffio, et al. 2017) also address this similarity problem for camera models coming from the same brand. The method is based on a CNN for feature extraction and a Support Vector Machine (SVM) for classification. In addition, open-set classification is addressed in an experiment. The objective is to identify the *known* and *unknown* cameras, which are used for training and only for testing, respectively. This problem is an important topic in the literature, which focuses on the robustness of DL-based methods. (Bayar and Stamm 2018b) conduct their experiments on the open-set scenario with a method based on a CNN, whose first layer is used as a pre-processing module. The objective of this approach is to classify the images as coming from a *known* or *unknown* camera models.

Transfer Learning

The uniqueness of the digital fingerprints raises the question of the robustness for the performance of SOTA methods. For example, a method with high performance during an evaluation on a database *B1* (used for training) could undergo a drop in performance on a new database *B2*. Indeed, if the cameras in the database *B2* are *unknown* to the network, the latter would not be able to classify the camera models correctly. In recent years, transfer learning, a field of DL, has been used to develop new networks in a faster way without losing efficiency. Notably, it exists three ways of fine-tuning networks (see 2.2.4): 1) Full network training; 2) Partial network training; 3) Partial network training.

3.2.2 Robustness Protocol

To tackle this problem of performance generalization, associated with similar digital fingerprints, we propose a robustness protocol. In this subsection, we present the architectures used for our assessment and the different experiments of our protocol.

Architectures

With the emergence of the DL during the last decade, several challenges for image processing have appeared, leading to the implementation of new architectures. Some of them have become standards for image processing applications, especially because of their performance. We decided to use three architectures with different aspects: VGG19 (Simonyan and Zisserman 2014), which is a classical CNN; ResNet50 (He et al. 2016), which is a CNN using shortcut connection layers to widen the domain of studied features; and DenseNet201 (G. Huang et al. 2016), which is a CNN connecting each layer with the following ones to obtain more complete and diverse features. For each architecture, we replaced the classification part with a flattening layer, two dense layers (of size 1028 and 512), two dropout layers (set to 0.5) and an output of size N (the number of cameras, depending on the experiment).

Protocol

The important aspect to consider for the evaluation protocol is the number of camera models used for feature learning and for classification. Let j be the cameras used for learning, referred as *known* cameras, to train the features extraction part of the network. Let k be the *unknown* cameras, which have not been analyzed by the network and should assess its robustness. Finally, let l be the number of cameras (both *known* and *unknown*) for evaluation respectively. We addressed two well-known topics in the literature: fingerprint uniqueness and *unknown* cameras. We first obtained reference networks (one per architecture) from a re-training of pre-trained networks on *ImageNet* (from ILSVRC). Our protocol consists of a series of three experiments (see Tab. 14) using fine-tuning approaches.

1) First, we perform a simple evaluation using transfer learning with full training of the networks. 2) Then, we tackle the *unknown* camera problem. 3) Finally, the last evaluation takes into account the two aspects studied: *unknown* cameras

Protocol	**Pre-training**	**Scenario**	**Evaluation**	
			Unknown	**Total**
Experiment 1	j=8	Only known	k = 0	l = 8
Experiment 2		Only unknown	k = 8	l = 8
Experiment 3		Known + Unknown	k = 19	l = 27

Table 14: Distribution of number of camera models.

and fingerprint uniqueness. The protocol is based on transfer learning, including the three possibilities of fine-tuning the network (see subsection 3.2.1). Network fine-tuning is an essential approach to transfer learning, and we evaluate all three approaches for their impact on performance. We conduct our protocol on the three presented architectures. The results and database of this study are detailed in section 3.2.3.

3.2.3 Experimental Evaluation

Databases

In order to make a fair comparison, all reference networks were trained with the *Dresden Image Database* (Gloe and Böhme 2010). It contains a total of 27 camera models for over $14,000$ images, from which we extracted patches of size 128×128 to fit the network inputs. Thus, the final dataset consists of 2.6 million patches, which we divided images from *known* cameras into three subsets: training (80%), validation (10%) and testing (10%). We defined *unknown* and *known* cameras according to their model, in order to avoid having patches from *unknown* cameras for the training. We used early stopping as recall (end of training if no improvement) and Stochastic Gradient Descent (SGD) as optimizer.

Preliminary Study

Before applying our protocol, a preliminary study is conducted to show the recognition problem related to *unknown* models and similar fingerprints. The objective is to compare the performance of each network (i.e., VGG19, ResNet50 and Denset201) in three experiments. The first experiment is equivalent to a baseline classification (8 *known* camera models) while the other two are considered as open scenarios (8 and 19 *unknown* camera models). The training of the networks was performed with transfer learning and fine-tuning of the classification part.

The results obtained (see Tab. 15) show a performance loss of about 10% of *accuracy* between the first and the second experiment, with a similar number of cameras (8 models). We therefore concluded that this loss of performance was due to the *unknown* models used in the second experiment. This phenomenon is even more accentuated for the third experiment (19 *unknown* camera models) with a decrease in *accuracy* of about 17% confirming our assumed problem.

Architectures	*Known*	*Unknown*	*Both*
VGG 19 (Simonyan and Zisserman 2014)	98.47 %	88.52 %	82.66 %
ResNet50 (He et al. 2016)	99.46 %	87.78 %	81.68 %
DenseNet201 (G. Huang et al. 2016)	99.49 %	90.82 %	81.82 %
Mean	99.14%	89.04%	82.05%

Table 15: *Accuracy* results from the three evaluation protocol experiments for a preliminary study.

Final Study

The final study is conducted with the same protocol to show the impact of different transfer learning approaches on performance. Moreover, the goal is to observe the behavior of the architectures when faced with the *unknown* camera problem. For the three experiments, we included in the results the learning time of one iteration as well as the *accuracy* of the networks in order to obtain a more complete comparison. The results obtained show that each approach to fine-tuning the networks after transfer learning has strengths and weaknesses (see Tab. 16).

Transfer	**Complete**		**Partial**		**Fine tuning**	
	Min.	Acc.	Min.	Acc.	Min.	Acc.
Unknown scenario						
VGG19	21.3	98.02 %	17.7	**97.69 %**	16.6	88.52 %
ResNet50	21.2	97.65 %	17.7	93.84 %	14.3	87.78 %
DenseNet201	**20.5**	**98.85 %**	**12**	96.97 %	**9.5**	**90.82 %**
Mean	21	98.11%	15.8	96.17%	13.5	89.04%
Known - Unknown scenario						
VGG 19	104	**93.48 %**	74.3	**91.22 %**	70.7	**82.66 %**
ResNet50	**88.2**	91.02 %	80.8	87.08 %	68.8	81.68 %
DenseNet201	93	92.58 %	**59.3**	90.05 %	**57.8**	81.82 %
Mean	95.1	92.36%	71.5	89.45%	65.8	82.05%

Table 16: Results (*accuracy*) for the selected architectures and transfer learning approaches in training time per iteration.

The results follow the phenomenon of transfer learning already shown by the literature: full training gives better results, but necessarily requires a longer training time, whereas it is the opposite for the fine-tuning of the classification part, which is faster, but also less accurate. Moreover, the partial fine-tuning offers a compromise between these two approaches. The *accuracy*-duration difference for full transfer versus partial fine-tuning is more advantageous for the unknown scenario (2% better for 6 minutes longer, see 16 - top rows) than for the known-unknown one (3% better for 25 minutes longer, see 16 - bottom rows), prompting a preference for partial fine-tuning for larger databases. In terms of robustness, VGG19 is more accurate, but takes longer to train against the DenseNet201 architecture. The phenomenon is similar to that of the fine-tuning approaches: the difference between *accuracy* and time for VGG19 versus DenseNet201 is more advantageous for the second experiment (0.7% better for 6 more minutes) than for the third (1.2% better for 15 more minutes).

Finally, the results show that partial fine-tuning (last block of the feature extractor and classification layers) is to be preferred over full training for large databases in order to benefit from efficient understanding. The same observation can be made for DL architectures favoring DenseNet201 over VGG19. Our preliminary study verifies the problem of *unknown* camera models and of similar digital fingerprints. These elements are at the root of the difficulty in generalizing the performance. However, this first protocol is mainly aimed at highlighting and confirming these problems rather than proposing a solution. The next section addresses the generalization problem with a multi-database protocol.

Source camera recognition is an important area of DIF and camera model classification is the most discussed application. The study of SOTA shows a difficulty in generalizing the performance. This aspect comes from the uniqueness of the artifacts for each camera. Moreover, camera models of the same brand show similarities in their digital fingerprints. Therefore, we propose a protocol to study the robustness of three different architectures: VGG19, Denset201 and ResNet50. This protocol is realized via transfer learning, which offers an interesting alternative to this problem. In particular, we exploit fine-tuning approaches to fully evaluate the impact. The protocol consists of three experiments, which merge *unknown* cameras and an increasing number of cameras to be classified. The preliminary study confirms the performance generalization problem, while the final evaluation shows that partial fine-tuning and DenseNet201 are better suited for large databases.

3.3 Comparative Study with Multiple Databases

In the literature, most of the work focuses on the identification of camera models. With the democratization of DL, approaches improved their results thanks to CNN. Regardless of the approach, these DL-based methods need databases composed of camera or smartphone images to learn and be effective. Despite the availability of such databases, the *Dresden Image Database* (Gloe and Böhme 2010) is often the only one used in most publications (see fig. 7). In some cases, private datasets are used as a second dataset, but they cannot be reused because they are not publicly available. This evaluation process contrasts with other areas of image processing, such as face recognition, where methods are typically trained on a specific database and evaluated on other databases.

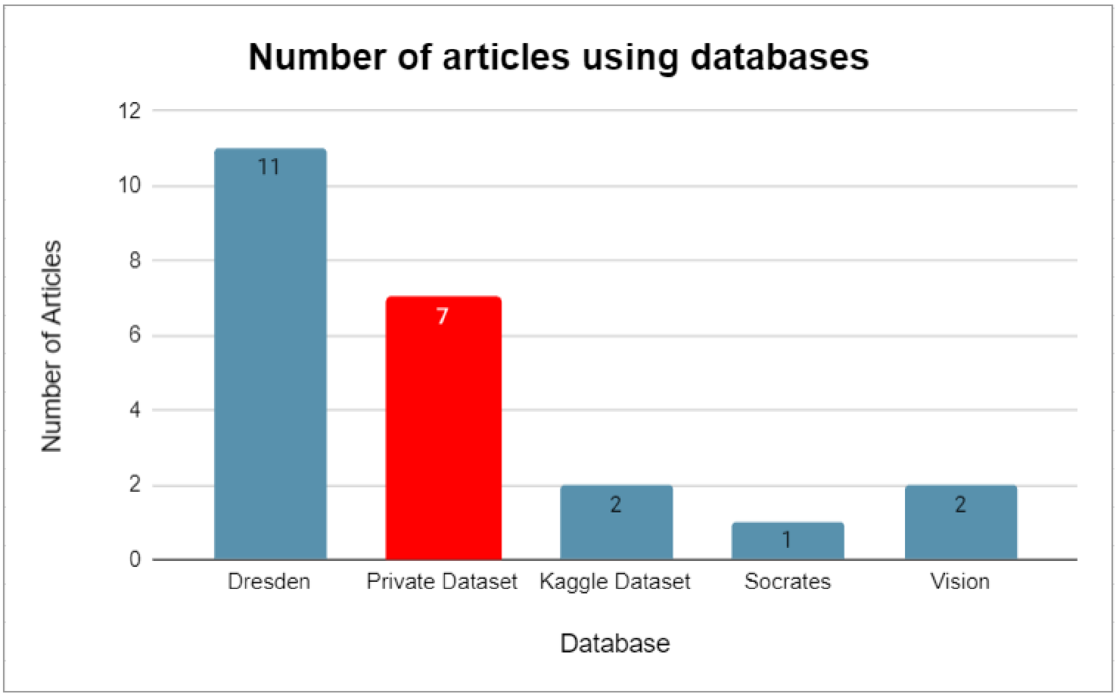


Figure 7: Statistics made among 13 articles linked to cameras: identification, extraction of pattern, etc. (blue) publicly available database (red) private datasets (not publicly available)

In this section, we provide a comprehensive comparative study to evaluate State-Of-The-Art (SOTA) methods for camera model identification, which includes two difficulties. The following subsection 3.3.1 introduces the issues raised by our literature review (see chapter 2, subsections 2.3.3 and 2.5.1) and the elements of our comparison protocol. The problem of database dependency is further detailed in the subsection 3.3.2. The State-Of-The-Art (SOTA) methods that we replicated for our performance evaluation are presented in the subsection 3.3.3. Finally, the subsection 3.3.4 presents the databases and properly explains the protocol of our evaluation with the results.

3.3.1 Introduction

Despite the abundance of articles concerning the identification of camera models, the literature presents some problems. The main problem remains the difficulty of comparing methods.

Classification Diversity

The first is the diversity of classifications among methods: i) scenario (basic, triple or open); ii) label (brand, model or device); iii) type (identification or verification). This disparity prevents comparison, even when the same database is used. To conduct our comparative study independently of this problem, we decided to focus on the identification of the basic camera model, without considering additional elements (brand, *unknown*, etc.).

Lack of Database

Most of the methods are evaluated on *Dresden* (see Fig. 7). Thus, the second problem is the lack of database diversity prevalent in the literature. Evaluating DL-based methods with a single database (i.e., *Dresden*) is not consistent with the usual evaluation process in image processing and computer vision. The results of such an evaluation are not reliable with respect to the diversity of the databases and their different characteristics. This problem arises from the digital fingerprint of the cameras, which is used for identification and is unique to each device. Therefore, classifying cameras from a database not used to train the network is more difficult. This is because the network learns to extract a fingerprint from an image, but cannot classify fingerprints that it has not analyzed. Thus, while DL methods have proven effective for source camera recognition, there is a data dependency problem specific to this topic.

Multi-Databases Protocol

Therefore, we conducted our comparative study on three complementary and publicly available databases: *SOCRatES* (Galdi et al. 2019), *Dresden* (Gloe and Böhme 2010) and *Forchheim* (Hadwiger and Riess 2020). They have as many characteristics as possible: all types of cameras, a diverse distribution and a different number of cameras. Our evaluation will focus on the basic classification of camera models. The evaluation protocol is based on transfer learning, and peculiarly on fine-tuning the network to evaluate the methods on each database. We applied our evaluation on four methods from the literature, based on various architectures and performed for various applications. Essentially, this protocol provides identification *accuracy* for these methods on an equal footing for various databases. This lack of diversity in the databases leads to unreliable and incomplete assessments. The diversity in classification also makes comparison difficult. In the following subsection, we illustrate the issue of database dependency.

3.3.2 Database dependency

The open-set classification is particularly interesting, as it addresses the problem of robustness. The evaluation of this aspect is difficult because each camera has its own fingerprint, which corresponds to the combination of artifacts generated during the creation of a digital image. Thus, the methods in the literature are often dedicated to the cameras in the database used for their training.

Overuse of *Dresden*

This database dependency, illustrated by the use of *Dresden* in nearly 50% of the literature, represents the second problem to overcome in comparing SOTA methods. In fact, most methods use the *Dresden Image Database* (Gloe and Böhme 2010), which is a standard in terms of camera identification, containing 27 camera models. This overuse represents an advantage for performance evaluation because the methods can be compared on a comparable basis. However, except a few papers using an equivalent number of camera models (i.e., 27) (Y. Chen et al. 2017; M. Zhao et al. 2020) for triple classification, most methods classify disparate numbers of camera models (see Tab. 11).

Robustness Problem

In addition, using a single database for performance evaluation poses another problem related to robustness. In another area of image processing, the standard method for evaluating methods requires the use of multiple databases. So far, the performance evaluation has almost always been performed on the same database as the training part and most often on a single database (N.B. *Dresden*). This is understandable because the identification of the camera is done by its fingerprint, which is unique to each camera. Therefore, evaluating a method on another database than the one used for training is too difficult and leads to a decrease in performance. However, this allows to prove the robustness of a method from one database to another.

Based on these peculiarities of the camera recognition literature, we decided to propose our own comparative study to overcome the problems raised above. The first aspect of our proposal is to provide a standard protocol for evaluating the performance of methods for camera model identification. Thus, we replicated four SOTA methods, which have their own peculiarities.

3.3.3 Camera Model Identification Methods

We have selected four methods that have proven to be efficient in terms of performance. There are several protocols discussed in the literature that we detailed earlier. In particular, to cover the full diversity of camera model identification, we took at least one method per scenario. Moreover, we have opted for different architectures in order to perform a complete evaluation. To be more specific, we chose to focus on the constrained convolutional network (Bayar and Stamm 2016) first introduced for tamper detection. This network is relatively new and has been an important improvement for pre-processing modules in DL methods. (Berthet and J.-L. Dugelay 2020). We also discussed Siamese Neural Network (SNN), which is currently used in the literature, especially to overcome the robustness problem. The other architectures are respectively based on an improved pre-processing module and a particular layer block inspired by residual neural networks (He et al. 2016). Therefore, in this subsection, we describe the four SOTA methods that we selected for our performance evaluation.

First Method

(Bayar and Stamm 2018b) present a constrained convolutional network for *unknown* camera identification, which they first introduced for tamper detection. In this architecture, the first layer is transformed from a convolutional layer into a pre-processing module. The goal is to extract the desired artifacts from within the network to achieve an end-to-end architecture. In fact, in many approaches, a pre-processing module is included to isolate artifacts that are overshadowed by the image content. In this network, artifacts are thus extracted by constraining the weights of the first convolutional layer. First, these weights are randomly initialized and then forced during back-propagation of the network to learn the prediction error filters. The weights w of each filter K are forced as follows: the center value is set to -1 while the sum of the remaining pixels is set to 1 (see Eq. 12).

$$\begin{cases} w_k(0,0) = -1 \\ \sum_{l,m\neq 0} w_k(l,m) = 1 \end{cases} \tag{12}$$

The identification of *unknown* cameras was done with *Dresden* and a personal data set. They defined the set T with two types of cameras: *known* for training and *unknown* to classify. The set T consists of 10 different *known* camera models from *Dresden* and 15 *unknown* camera models from their personal dataset. They also defined a set *T'*, which represents the cameras to be classified that are not in the base set T. The set *T'* consists of 15 new *unknown* camera models from their personal dataset. The evaluation was conducted using a protocol of three experiments: i) They used the T set for the first evaluation. ii) They used the combination of the two sets T and *T'*. iii) The final experiment is a subset consisting of the *known* cameras from the T set and the *unknown* cameras from the *T'* set. The architecture is divided into three parts: 1) the pre-processing module with the constrained convolutional layer (Bayar and Stamm 2016); 2) the feature extractor with four convolutional blocks consisting of a convolutional layer, batch normalization, activation Hyperbolic Tangent (TanH), and pooling (max and average for the last one); 3) the classification with 3 FC layers with activation TanH and soft max for the last one. Their results for each experiment are: 1) 99.38%; 2) 98.57%; 3) 97.74%.

Second Method

(Bayar and Stamm 2017a) also proposed a network based on this constrained convolution network. The only change in this architecture, compared to the previous one, is the pre-processing module. They decided to augment the feature maps of their model by combining two artifact extraction layers: the constrained convolutional layer and a Median Filtering Residual (MFR). The goal was to evaluate the robustness of this artifact fusion for camera model classification. They conducted a series of experiments to compare their approach with other methods such as the High-Pass Filter (HPF) based on CNN and the classical constrained convolutional network. They evaluated the robustness of these methods with resampling (120%, 90%, 50%), resampling+JPEG (QF=90) and normal images. In most cases, the proposed method showed the most efficient performance and, except one test (50% resampling + QF=90), the *accuracy* was always above 90%.

Third Method

(Mayer and Stamm 2020) address the generalization of camera model identification with a new approach. The objective is to propose a method to determine whether two images come from the same camera without knowing their forensic traces. Their method is based on the SNN: a feature extractor used twice in parallel to produce deep features from two image patches and a similarity network to compare them. For the feature extractor, they used only the constrained convolutional network (Bayar and Stamm 2018a). It is mostly the same as before, except that the classification consists of only 2 FC layers. The similarity network consists of three parts: 1) a first FC layer for each branch; 2) then each branch as well as their product passes through an artificial layer f_{inter} (Eq. 13); 3) a concatenation layer, a FC layer and a sigmoid activation to obtain a similarity score.

$$f_{inter}(X) = \phi(\sum_{i=1}^{N} w_{k,i} \quad f_i(X) + b_k) \qquad (13)$$

For their experiments, they also treated aspects of the *known* and *unknown* camera models to clearly evaluate the generalizability of their method. *Accuracy* was calculated for three distinct cases: with the *known* camera models only (95.93%), with the *unknown* and *known* camera models (93.72%), and with the *unknown* camera models only (92.41%). They also evaluated the best configuration (patch size, architecture, etc.) and the effects of other parameters (recompression, etc.) for their method.

Fourth Method

(Ding et al. 2019) addresses multi-classification with the camera identification according to the triple classification (brand, model and device). The method relies on a pre-processing module based on domain knowledge and ResNet blocks. The ResNet is known for its shortened layers that improve diversity by allowing more exploration of the feature space. A ResNet block is defined as the merging of features obtained in parallel by two consecutive 3 × 3 convolutional layers. The architecture is developed in four parts: the pre-processing module and three successive sections for each classification (brand, then model and device). Each section consists of three consecutive ResNet blocks followed by a classification part composed of a global average pooling, a FC layer and a soft max layer. The pre-processing module is composed of three parts: 1) a multiscale HPF to obtain three different residuals; 2) a convolutional layer and a ResNet block applied on the 4 elements (HPF and input); 3) a concatenation layer to obtain the final output. They obtained an *accuracy* of 97.1% for camera model classification.

Methods	**Preprocessing**	**Features extractor**	**Classification**
(Bayar and Stamm 2017a)	Const. conv + MFR	4 conv. blocks	3 FC
(Bayar and Stamm 2018b)	Const. conv.	4 conv. blocks	3 FC
(Mayer and Stamm 2020)	Const. conv	4 conv. blocks + 2 FC	Similary network
(Ding et al. 2019)	multi-scale HPF	3 ResNet blocks	3 class. blocks

Table 17: Details of method architecture, according to the preprocessing, the features extractor and the classification.

The table 17 completely summarizes the purpose of our choice. On the one hand, we can observe four completely different methods, although three of them use the constrained convolutional layer for pre-processing. However, it is difficult to estimate which one is the most robust and efficient. Therefore, we performed a comprehensive evaluation of our protocol based on three databases, which we detail in the next section 3.3.4.

3.3.4 Multi-Databases Protocol

To date, the methods from the literature have demonstrated a lack of diversity for databases. Most methods have only used the *Dresden Image Database*. In contrast, we decide to exploit three databases, dedicated to camera recognition, for our protocol. We use *Dresden* (Gloe and Böhme 2010) because it is a standard in the literature. In addition, we exploited *SOCRatES* (Galdi et al. 2019) and the *Forchheim Image Database* (Hadwiger and Riess 2020) which are dedicated to smartphone cameras. Our protocol is a two-stage evaluation based on transfer learning and especially fine-tuning. In this subsection, we detail the contents of each database and the steps of our multi-database protocol.

Databases

The *Dresden Image Database* (Gloe and Böhme 2010) is perhaps the most popular database in the field of DIF. This database is mainly dedicated to the identification of cameras, but it is also used for the detection of falsifications (with application of manipulations to the images). It is composed of more than $14,000$ images of various indoor and outdoor scenes that were captured by 74 cameras (14 different brands and 27 models) in order to perfectly establish their characteristics. As detailed in the paper, the development of forensic methods for camera model identification requires many images per camera of the same scene to perform a complete comparison of their forensic traces.

SOurce Camera REcognition on Smartphones (*SOCRatES*) (Galdi et al. 2019) is a database of images and videos. It is specially designed for the recognition of the source camera on mobile devices. *SOCRatES* is currently one of the databases with the largest number of different devices. It consists of approximately $9,700$ images and $1,000$ videos captured with 103 different smartphones (2 unclassified) from 15 different brands and 62 different models. The acquisition was performed under uncontrolled conditions. To collect the database, multiple individuals were involved and asked to use their personal smartphones to collect a set of images. Instructions were given to the participants, and they collected the set of pictures independently. The reason for this choice is, on the one hand, to collect a heterogeneous image database and to maximize the number of devices employed, and, on the other hand, to carefully reproduce the actual scenario of applying techniques that will use this database as a reference.

The *Forchheim Image Database* (Hadwiger and Riess 2020) is also a database dedicated to source camera recognition from smartphones. *Forchheim* is one of the largest databases available for source camera identification, with over $23,000$ images from 143 scenes captured by 27 different smartphone cameras (9 brands and 25 models). All images were captured in or near the city of Forchheim, Germany (hence the name) and each camera shows one image per scene. In addition, each image is provided in 6 different qualities to assert a disparate quality: the original version from the camera and 5 copies from Social Networks (SNs).

We chose these databases for their particularities in terms of variety - cameras (*Dresden*) and smartphones (*SOCRatES* and *Forchheim*) - and composition with different ratios of models per camera: i) 36.48% for 74 cameras (*Dresden*); ii) 61.39% for 101 cameras (*SOCRatES*); ii) 92.59% for 27 cameras (*Forchheim*). With three databases, we want to clearly observe the robustness of the methods with our evaluation.

Protocol

The fine-tuning of the networks facilitates the evaluation of the methods from one database to another (N.B. *Dresden*, *SOCRatES* and *Forchheim*). The protocol is divided into two parts: 1) creation of a reference network for each database; 2) transfer of the architecture and weights of the reference networks to obtain the transferred networks.

The first step in our protocol is to create a reference network for each possible combination of method and database. Our study is based on three databases (*Dresden*, *SOCRatES* and *Forchheim*) and four SOTA methods. So, we obtained a total of 12 networks for the first step. On the one hand, this part gives indications on the efficiency of a method depending on the database, but also on the most difficult database. On the other hand, this first step allows us to obtain the reference networks necessary for the second step. The second step consists in refining (i.e. re-training) the transferred network, with transfer learning and notably the fine-tuning of the classification part, as we see in the previous section 3.2 that it is the most challenging choice.

A transferred network is created for each database from the reference network. We thus obtained 3 networks per method to be evaluated on each database (see table 18). The goal of this second step is to analyze the robustness of each method, but also to evaluate which database is the most efficient to create the reference network.

3.3.5 Experimental Results

Training

The experimental study is conducted on the SOTA methods presented with the previously detailed databases. To evaluate these methods with our protocol, we

Reference Database (B1)	**Transferred Database (B2)**	**Total**
Dresden	Dresden - SOCRatES (D-S)	**3 networks**
	Dresden - Forchheim (D-F)	
SOCRatES	SOCRatES - Dresden (S-D)	**3 networks**
	SOCRatES - Forchheim (S-F)	
Forchheim	Forchheim - SOCRatES (F-S)	**3 networks**
	Forchheim-Dresden (F-D)	

Table 18: Scheme of the steps of the protocol with reference and transferred databases.

divided each database into three datasets for training, validation and testing (80 : 10 : 10). Then, from each image, we extracted patches of size 128 × 128 and applied numerous pre-processing operations to fit them to the input of the networks. In particular, we applied successive Gaussian filters on the patches to obtain 4 different inputs, as in the article (Ding et al. 2019). For the constrained convolutional network (Bayar and Stamm 2018b), the pre-processing is included in the first layer. However, we had to apply a Median Filter (MF) to the patches to create the second entry of the enhanced constrained convolutional network (Bayar and Stamm 2017a). We created pairs of patches for the SNN, which requires two inputs (Mayer and Stamm 2020).

Finally, we obtained 2.5, $1.16M$ and $0.59M$ of patches from *Dresden*, *Forchheim* and *SOCRatES*, respectively (same split). For the training, we selected the same number of patches (based on the smallest set). Once the datasets were defined, we trained the selected methods according to our protocol, i.e., in two parts: the creation of reference networks and the transferred networks. For the creation of reference networks, we trained the models for 30 epochs with the hyper-parameters specific to each method. Then, we trained the transferred networks for 10 epochs to limit the adaptation to the new database, with the hyper-parameters specific to each method too.

Results

The results obtained with the *accuracy* give a complete overview of the camera model identification (see table 19). Unlike the table 11, the methods can be compared with each other independently of their approach. Therefore, comparative study is possible, and some conclusions or discussions can be made. On the one hand, (Ding et al. 2019) seems to propose the best method when changing the database between training and testing (6 of the 9 best results). On the other hand, (Mayer and Stamm 2020) obtain the best *accuracy* when the training and the test are performed on the same database. (Bayar and Stamm 2017a, 2018b) show stability in performance regardless of the training database, but their results are less convincing than those of Tab. 11. On the other hand, *Dresden* appears to be the easiest database to handle (overall *accuracy* of 90.69%). These results confirm that the evaluation of methods on a single database is not relevant and that other databases are required.

Network	(Bayar and Stamm 2018b)	(Bayar and Stamm 2017a)	(Mayer and Stamm 2020)	(Ding et al. 2019)
		Evaluation on Dresden		
Dresden	91.78%	92.11%	**96.17%**	94.66%
S-D	90.20%	92.19%	86.01%	**96.68%**
F-D	89.08%	91.68%	80.42%	**97.30%**
		Evaluation on SOCRatES		
SOCRatES	75.20%	75.05%	**92.31%**	88.08%
D-S	79.43%	**80.98%**	75.37%	74.95%
F-S	73.97%	77.21%	81.82%	**87.45%**
		Evaluation on Forchheim		
Forchheim	56.14%	57.65%	80.18%	**82.89%**
D-F	58.44%	60.63%	66.67%	**69.08%**
S-F	57.01%	58.56%	79.90%	**81.14%**

Table 19: Results of each network according to the database of evaluation and the method used.

Thus, we propose the first protocol to evaluate methods on several publicly available databases: *Dresden*, *SOCRatES* and *Forchheim*. This evaluation takes into account two crucial problems in the literature: the use of a single database and the disparity of approaches between methods. Thus, comparative study is possible across various databases and across method approaches. Furthermore, the results show that the triple classification-based method outperforms the others in most cases, while (Mayer and Stamm 2020) also performs well.

Camera recognition is particularly dedicated to the identification of camera models. However, DL-based methods address several scenarios: basic, triple or open-set. Unlike other areas of image processing such as face recognition, most of these methods are only evaluated on a single database (*Dresden*) while a few others are publicly available. The available databases have a diversity in terms of content and camera distribution that is unique to each of them and makes the use of a single database questionable. Therefore, we performed extensive tests with different public databases (*Dresden*, *SOCRatES*, and *Forchheim*) that combine enough features to make a viable comparison. In addition, the different scenarios pose a disparity problem preventing comparisons. Therefore, we decided to focus only on the identification of the base camera model. Our protocol is the first multi-database evaluation for camera recognition in the literature.

3.4 More reliable and reproducible protocol

Most DL-based approaches recognize the camera according to its model - a task called *camera model recognition* in the literature. However, this task is not sufficient in most scenarios where the set of cameras considered contains at least 2 cameras of the same model. In this case, the recognition of the source camera must be based on the specific features associated with the device - what we will call *camera device recognition*. The literature on camera recognition shows the increasing difficulty of classifying the camera according to the labels: brand, model, and device - where brand is the easiest and device the hardest to classify. This problem stems from camera fingerprints, which are more likely to be close to each other for cameras of the same brand and model. In the literature, DL-based methods have largely addressed camera model recognition, while camera device recognition is still under-studied. The few papers that deal with camera device recognition, however, do not fully address the problem of close camera fingerprints.

In addition, the evaluation protocol adopted by these methods is that of identification ($1-to-N$) and the database most used in their experiments is *Dresden* (Gloe and Böhme 2010) (see section 3.3). The following problems were identified in this regard: i) the $1-to-1$ verification protocol might be more appropriate in some cases. When we want to know whether an illegal image was captured by a certain device, we will compare it to the fingerprint of that device; ii) the distribution of cameras in the database (e.g., the number of cameras for each model) is not controlled. Therefore, the different levels of difficulty of classification are not highlighted as they depend on the distribution of cameras; iii) using a single database for testing means having always the same exact composition of cameras, which is not representative of real life since. For example, more than 1.6 billion capture devices were sold in 2020 (cameras[1] and smartphones[2]).

Based on this, we decided to focus our work on the verification protocol ($1-to-1$) and on a controlled selection of cameras, so that the distribution does not depend on the distribution of the chosen database. We propose a reliable and reproducible protocol to fully evaluate the SOTA methods. This protocol consists of three levels of difficulty, namely *basic*, *intermediate*, and *difficult*. They respectively correspond to the selection of cameras according to three camera characteristics: brand, model, and device. To our knowledge, this was the first verification protocol to comprehensively evaluate DL-based methods for camera recognition. The subsection 3.4.1 presents the relevant methods of SOTA dealing with camera recognition. In subsection 3.4.3, we explain our motivations and detail our proposed protocol. The experimental evaluation is described in the subsection 3.4.5 with a special metric specifically designed to evaluate the impact of difficulty levels.

3.4.1 Related Work

Regarding traditional source camera recognition approaches (i.e., not based on DL), the most widely used and effective are those based on the analysis of Sensor Pattern Noise (SPN), first introduced in 2006 (Lukas et al. 2006), and improved by several works in the following years. This method is based on the analysis of noise residuals. The challenge today is therefore to further improve the recognition performance of cameras using DL, which has greatly improved the performance of many image processing tasks so far. In the following, we present the literature on camera device recognition with DL, analyzing their architecture and evaluation protocol based on identification.

First Method

(Y. Chen et al. 2017) address multiple classification in three experiments to provide performance for each label: brand, model, and device. Their method is based on the Residual Neural Network (ResNet) (He et al. 2016), which is a network that incorporates shortcut connections in its layers. The idea is to keep the low-level features, while the convolutional layers process the images to obtain high-level features. By combining both, the final result is more complete and includes more information to recognize camera fingerprints. They achieved identification *accuracy* for brands (99.12%), models (94.73%) and devices (45.81%).

[1] https://www.statista.com/statistics/1172711/forecast-of-digital-camera-sales-volume/

[2] https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/

Second Method

In the following two works, multiple classification is also addressed with a very similar protocol, as they produce predictions for all three labels (brand, model and device) with a single experiment. (Ding et al. 2019) use a pre-processing module, which exploits a concatenation of 3 HPFs and the original image to obtain a greater diversity in the features. The network is composed of three parts that are built with three ResNet blocks followed by a classification layer to identify a single label: first the brand, then the model, and finally the device. The ResNet blocks consist of two consecutive convolutional layers in parallel with a single convolutional layer, for high and low level feature extraction. They achieved identification *accuracy* for brand (99.6%), model (97.1%) and device (52.4%).

Third Method

(M. Zhao et al. 2020) propose a method based on the combination of a ResNet in parallel with a set of convolution layers, which extract respectively the camera attributes and the relevant information from the image neighborhoods. They use a recursive method with a cascading classification: predictions are given with consecutive sub-classifiers (first the brand, then the model, and finally the device). The sub-classifier can influence the parent-classifier to drop some features that are valuable for the sub-classification. They achieved identification *accuracy* for brand (99.4%), model (96.1%), and device (47.5%).

Model vs. Device

In these SOTA methods, evaluations were performed by identification $(1-to-N)$. They showed that recognition is increasingly difficult for devices sharing the same brand and model. Thus, camera device recognition is the most difficult task (note the drop in performance by up to half when classifying brands or models against devices).

Regarding the recognition of devices sharing the same brand, the difficulty of classifying them is confirmed. Indeed, Tab. 20 reports part of the confusion matrices from SOTA methods for some camera models. The performance of camera model recognition is lower for cameras of the same brand. The tab. 21 shows the results of the same SOTA methods as before, but this time used for device classification. The table shows the average *accuracy* for selected camera models, which was calculated on 3 devices per model. The table also shows the overall *accuracy* of the classification, which is much lower than for the model classification (see Tab. 20 overall *accuracy* for comparison).

The drop in performance between the two tasks (model and device identification) is surely due in part to the number of classes on which to classify cameras,

Method	(Y. Chen et al. 2017)		(Ding et al. 2019)		(M. Zhao et al. 2020)	
Camera model	*CI55*	*CI70*	*CI55*	*CI70*	*CI55*	*CI70*
Canon Ixus 55	**56%**	38%	**76.5%**	23.5%	90%	9%
Canon Ixus 70	6%	87%	0.6%	99.4%	4%	96%
Camera model	*ND70*	*ND70s*	*ND70*	*ND70s*	*ND70*	*ND70s*
Nikon D70	**58%**	39%	**69.6%**	29.5%	**64%**	35%
Nikon D70s	42%	**56%**	53.2%	**44.1%**	41%	**58%**
Overall Accuracy	94.73%		97.1%		96.1%	

Table 20: Confusion matrix for camera identification according to their model. Performances in the original papers are assessed over 27 models. Here, only *Canon Ixus 55/70* and *Nikon D70/D70s* are reported.

Camera model	(Y. Chen et al. 2017)	(Ding et al. 2019)	(M. Zhao et al. 2020)
FujiFilm FinePixJ50	48.14%	49%	-
Olympus Mju-1050SW	-	**43.33%**	-
Sony DSC-T77	-	**77.67%**	**64%**
Samsung NV15	-	-	47%
Casio EX-Z150	-	-	**35%**
Overall Accuracy	45.81%	52.4%	47.5%

Table 21: Confusion matrix for camera device identification. Performances in the original papers are assessed over 74 devices. Here, only some devices are selected. *Accuracy* is averaged over three devices per model. Bold font indicates performance values that are larger or smaller than the overall accuracy.

which is generally higher for the device than for the model (i.e., in a dataset, there are generally more different camera devices than different camera models). It is known that in DL, the *accuracy* and the number of classes are inversely correlated. This decrease is also due to the fact that cameras of the same brand and model have close fingerprints, which is further analyzed in the next subsection 3.4.2.

3.4.2 Close Camera Fingerprints

The literature has shown that camera recognition is becoming increasingly difficult as cameras of the same brand or model have similar digital features. (Ding et al. 2019) illustrates the problem of similar camera fingerprints well with their feature visualization graph extracted with t-Distributed Stochastic Neighbour Embedding (t-SNE) (Fig. 8). This visualization highlights the similarity of camera features based on their brand and model. For example, cameras *Olympus mju 1050SW* are quite difficult to group together. This graph also shows that cameras of the same model can still be differentiated, such as the *Sony DSC-T77*, whose features can be grouped for each camera of that model.

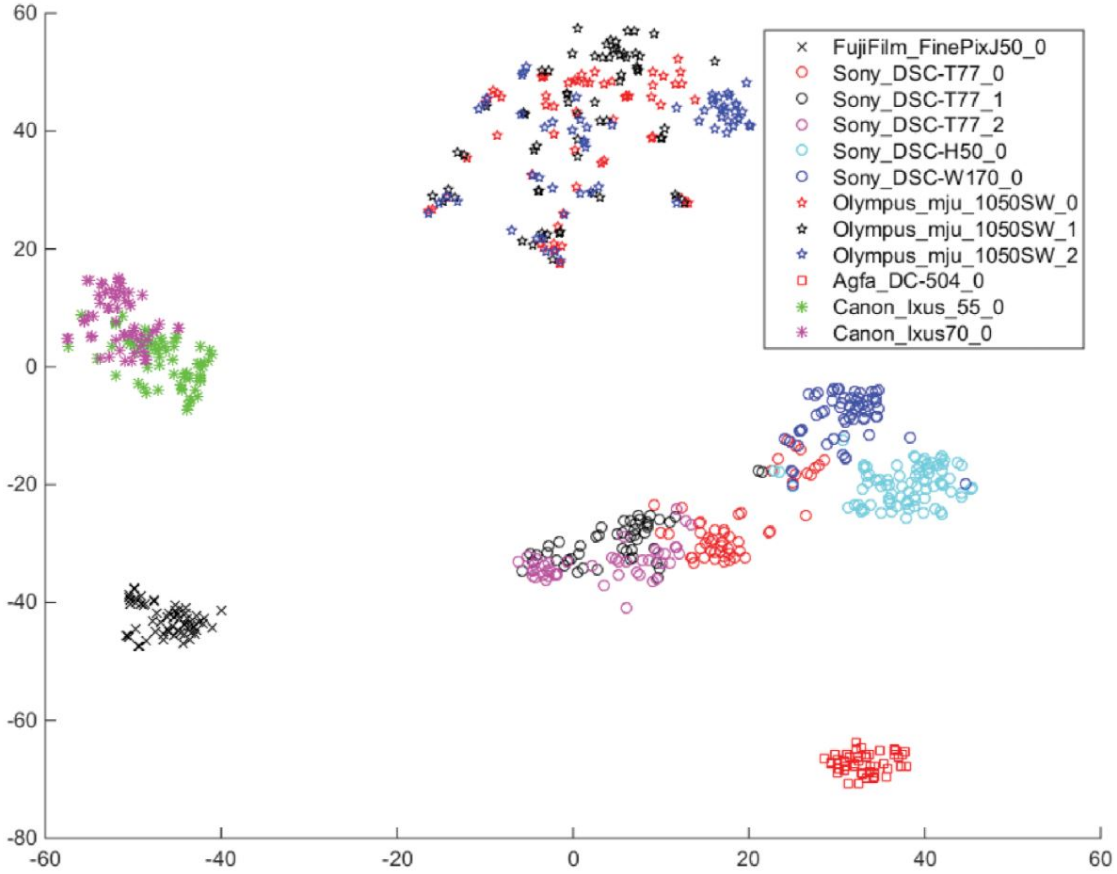


Figure 8: Visualization of the similarity of different cameras in the feature space t-SNE (Ding et al. 2019). (stars) Olympus; (circles) Sony; (asterisk) Canon; (cross) Fuji; (square) Agfa.

The issue of close camera fingerprints has been mentioned in the literature of DL-based methods for camera recognition, especially via confusion matrix analysis. However, it has never been fully addressed. In particular, SOTA methods are always evaluated by identification $(1 - to - N)$ on an entire database, which does not highlight the challenge of the similarity of camera fingerprints. Indeed, in this type of evaluation, the distribution of cameras is often, if not always, uncontrolled. Cameras of the same model (or brand) are mixed with many other models (or brands). Thus, the difficulty of classification can be different from one database to another. To overcome this problem, we propose to adopt a protocol that uses camera selection to create sets with controlled camera distribution. Camera selection controls the presence of cameras with close digital fingerprints, and thus controls the difficulty of classification. In addition, the protocol we propose in the following subsections is based on $1 - to - 1$ verification, as we believe that verification is more likely to be used in future applications (e.g., in police investigations). In other words, the goal is to distinguish iPhone 11 of Bob from iPhone 11 of Alice rather than recognizing it in a random group of smartphones.

3.4.3 Verification Protocol

Verification has already been adopted in some camera recognition work. For example, (Mandelli et al. 2020) propose an Siamese Neural Network (SNN) for device recognition, by evaluating the similarity of the camera fingerprint between pairs of images. SNN is an architecture that has been used quite a bit for camera recognition and particularly in model classification. The network is composed of two twin subnetworks whose weights are updated identically. They trained one part of the network with coherent pairs and the other with non-coherent pairs. The residual noise of an image associated with a device d_i is combined with the PRNUs of the same device d_i and a dissimilar device d_j to create coherent and non-coherent pairs, respectively. The PRNU is obtained from a large set of images of each device to obtain a more robust and reliable model. The idea is to extract and then compare the PRNU using the SNN, each of which outputs an encoding of the input image (e.g., a vector of size 1024). The network works in tandem on two different input images to compute comparable output vectors.

Instead of asking "which class does the image come from?", SNNs answer the question "do the two images belong to the same class?". We can draw a parallel with biometric recognition by saying that single-stream networks perform a $1-to-N$ comparison, and thus identification, while Two-Stream Network (TSN), such as SNNs, perform a $1 - to - 1$ comparison, and thus verification. In fact, camera recognition is sometimes even called *hardwaremetry* (Galdi et al. 2015). A major advantage of using SNNs is that, once trained, they are able to establish whether two images are from the same class, even for unseen classes. The goal is to determine whether two images are from the same camera. One of the SOTA methods is based on SNN, and thus naturally involves evaluation by verification. In addition, we propose to evaluate single-stream SOTA methods with a verification protocol. To do so, the encoding of an image computed by the neural network is extracted before the classification part, and compared to other encodings in a $1 - to - 1$ comparison by Euclidean distance.

3.4.4 Proposed Cameras Selection

Traditionally, the evaluation of camera recognition methods is performed on the entire database without any particular camera selection strategy. However, using the entire database as is does not take into account the problem of close camera fingerprints. Ideally, databases for camera recognition should contain a large and balanced number of cameras of the same model, otherwise it would be unclear whether a method actually classifies the camera based on camera model recognition or camera device recognition. Along with biometric recognition, it would be like having a database of only young women and elderly men, how do you establish whether the model actually recognizes gender rather than age? Thus, we propose a selection of cameras to have a clear control on their distribution, in order to ensure that the recognition is performed for the device.

In practice, the currently available databases have a very limited number of cameras sharing the same model. Our proposed protocol is based on a camera selection that allows us to define subsets of the existing datasets to test SOTA methods at different levels of difficulty. The selection strategy aims to select pairs of cameras for $1 - to - 1$ comparison, and notably for dissimilar pairs (i.e. same camera for similar). We have to ensure that when dissimilar pairs are created, they reflect a level of recognition difficulty associated to a camera label: brand, model or device.

To confirm the problem of increasing difficulty in classifying from brand to camera, we propose to create three levels: i) with only cameras of different brands (*basic*); ii) with only cameras of the same brand and different models (*intermediate*); iii) with only cameras of the same brand and model (*advanced*). Even among these levels of difficulty, some cameras may be easier to classify than others, as shown in the confusion matrices in the subsection 3.4.1. (M. Zhao et al. 2020) propose a method that is able to distinguish well *Canon Ixus 55* from *Canon Ixus 70*, while this was not the case for *Nikon D70* and *Nikon D70s*. Since the verification is performed with pairs of images, these difficulty levels will represent the different dissimilar pairs (see Fig. 9). The problem of database distribution is fixed to the controlled selection of image pairs according to the three difficulty levels.

3.4.5 Experimental results

The protocol with our proposed camera selection is applied to 4 different methods SOTA in order to have a complete analysis of camera recognition that suffer from the problem of classifying devices with close camera fingerprints. Among the previously described camera recognition methods (see subsection 3.4.1), we selected the most efficient (Ding et al. 2019). We also selected two methods for camera model recognition, which were re-trained to perform camera device recognition, to test other architectures. Both methods are based on constrained CNN: the first (Bayar and Stamm 2017a) is the constrained CNN, and the second (Bayar and Stamm 2018b) incorporates enhanced pre-processing. Finally, a method based on SNN (Mayer and Stamm 2020) is also selected in order to test the robustness of such architecture. The study is conducted on two databases, chosen for their different characteristics: *SOCRatES* and the *Dresden Image Database*. In the case of

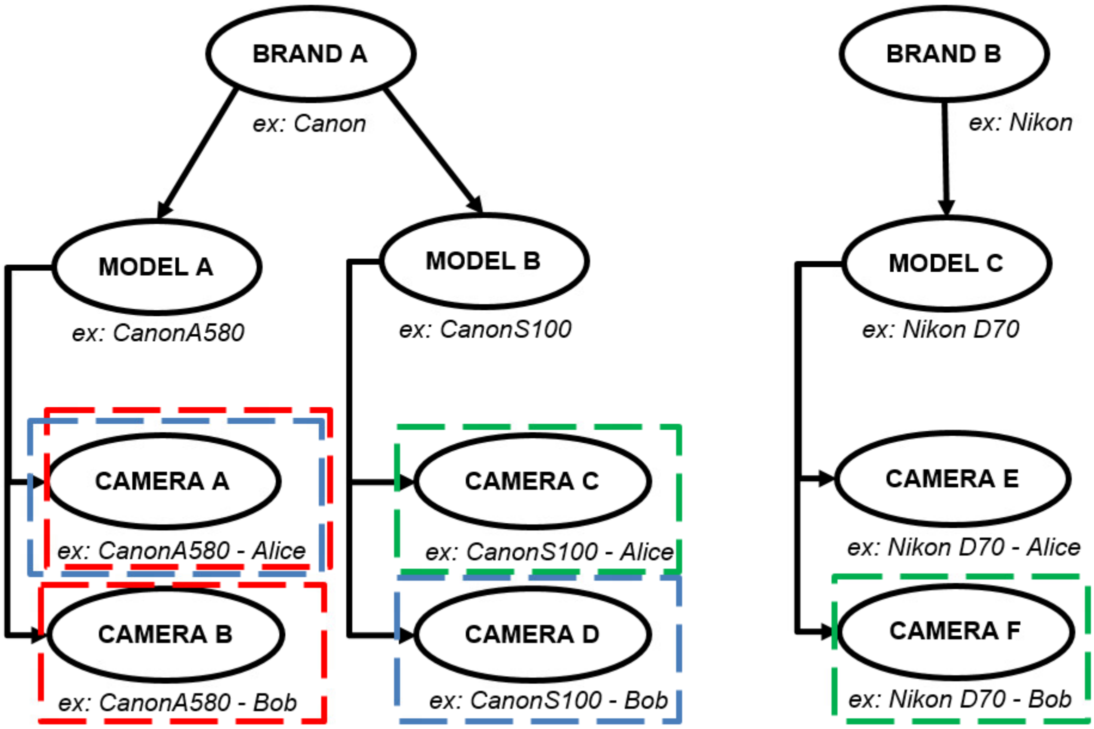


Figure 9: Diagram illustrating difficult and classical dissimilar pairs. (Red) Advanced; (Blue) Intermediate; (Green) Basic.

the methods originally designed for identification, the architecture is adapted for verification by removing the classification layer and comparing the output feature vectors (or coding) with the Euclidean distance.

Databases

SOCRatES (Galdi et al. 2019) is a database of images and videos specifically designed for source camera recognition on smartphone devices. It is composed of about 9,700 images and 1,000 videos captured with 101 different smartphones of 15 different brands and about 62 different models. The acquisition was performed under uncontrolled conditions. The *Dresden Image Database* (Gloe and Böhme 2010) consists of over 14,000 images of various indoor and outdoor scenes that were captured by 74 cameras of 27 different models. Tab. 22 gives an overview of the distribution of the two databases. A difference can already be made in terms of variety of brand: there is an over-presence of Apple and Samsung cameras in *SOCRatES* compared to the other brands, while for *Dresden* the distribution is more uniform. In addition, there is another specificity at the device level: most cameras have only one device in *SOCRatES* while in *Dresden* only a few cameras are represented with one device. Thus, these two databases have really different camera compositions, which highlights the problem of using a single database for evaluation. Furthermore, this compositional specificity will likely affect the results.

Dresden Image Database					
AgfaPhoto		**Canon**		**Nikon**	
DC-504	1	Ixus 55	1	Coolpix S710	5
DC-733s	1	Ixus 70	3	D70/D70s	2/2
DC-830i	1	PS A640	1	D200	2
Sensor 505-X/530s	1/1	**Casio**		**FujiFilm**	
Sony		EX-Z150	5	FinePix J50	3
DSC-H50	2	**Pentax**		**Samsung**	
DSC-T77	4	Optio A40	4	L74wide	3
DSC-W170	2	Optio W60	1	NV15	3
Kodak		**Panasonic**		**Rollei**	
M1063	5	DMC-FZ50	3	RCP-7325XS	3
Ricoh		**Olympus**		**Praktica**	
Capilo GX100	5	1050SW	5	DCZ 5.9	5
Total brand	14	**Total model**	27	**Total device**	74
SOCRatES					
Apple		**Asus**		**HTC**	
iPhone 4s	3	Zenfone 2/3	3/1	One M8	1
iPhone 5/5s	1/2	**Huawei**		**Lenovo**	
iPhone 5c	6	P7/P8 Lite	1	S60	1
iPhone 6/6s/6s plus	8/3/1	**Motorola**		**Acer**	
iPhone 7	3	Moto G/G3	3/2	Liquid E700	1
iPhone SE	1	Moto X-Style	1	**OnePlus**	
iPad Mini 2	1	X Play	1	X/One	1/1
Samsung		**LG**		**Nokia**	
S3/S3 Neo	1/2	G3/G4	4/2	Lumia 635/930	1/1
S4/S4 mini	2/1	Nexus 5X/5	2/1	**Wiko**	
S5/S5 mini	4/1	Spirit LTE	1	Rainbow 4G/Up 4G	1/1
S6/S6 Edge	1/1	K10 4G	1	Highway 4G	1
S7 Edge	2	**Sony**		Birdy 4G	1
Core Max/Prime	1/2	Xperia Z/Z1	1/1	**Vernee**	
Grand Plus/Prime	1/1	Xperia Z3/Z5	3/1	Thor	1
A3/A510	2/1	Xperia T3/E3/M4	1/1/1	**Meizu**	
J7/Note 4	2/1	NEX-VG20	1	M3 Note	1
Total brand	15	**Total model**	62	**Total device**	101

Table 22: Details of the databases: the brand, model and the number of devices; Some devices are on the same line (e.g. S3 and S3 Neo).

Evaluation

Creating the datasets required two steps: establishing a dataset of patches and then the pairs. First, we cropped each image in both databases by a window of size 128×128 pixels. Then, we selected these patches based on their brightness, as dark and saturated areas are not optimal for sensor noise extraction. We selected 2.7 million and $630,000$ patches from the *Dresden* and *SOCRatES* databases, respectively. We divided the two datasets into three subsets $(80 : 10 : 10)$, corresponding to training, validation, and testing, respectively. For training, we thus selected patches according to the smaller set (i.e., *SOCRatES*) to respect balance. The training and validation sets are used to train the SOTA networks following their original protocols, as described in the corresponding papers. The datasets for each difficulty level are created with the test subset based on their respective pair selection. The code used to generate the image patches and the different image pair selections is available online [3] for reproducibility purposes.

Metrics

The performances of the SOTA methods are reported in terms of AuC of the ROC, which plots the true positive rate (see Eq. 14) against the false positive rate (see Eq. 15, with True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN)).

$$TPR = \frac{TP}{TP + FN} \qquad (14) \quad \text{and} \qquad\qquad FPR = \frac{FP}{FP + TN} \qquad (15)$$

An additional metric is used to show the relative drop in performance between the *basic* and the *advanced* levels of difficulty (see Eq. 16). The higher the AuC value, the better the classification capability of the method. The lower the drop value, the more robust the classification method.

$$drop = \frac{(AuC_{BASIC} - AuC_{ADVANCED})}{AuC_{BASIC}} * 100 \tag{16}$$

Since our camera selection strategy involves some randomness, the Monte Carlo method (Kroese et al. 2014) is adopted for random sampling. Therefore, more than 50 repetitions of our protocol are performed, and the average scores are calculated.

Results

The results are presented in Tab. 23. (Ding et al. 2019) show the best results in terms of relative drop, which means that the results are more stable over the three

[3] https://gitlab.eurecom.fr/imagingsecuritypublic/eurecom_difficultdeviceevaluationprotocol

difficulty levels, regardless of the database used for evaluation. The robustness of this method probably comes from its architecture since it is designed to perform a triple classification (i.e., brand, model and device). On the contrary, the other SOTA methods perform better for the *basic* and *intermediate* levels, but the drop for the *advanced* level is more important. Thus, they fail the test to determine if they can actually distinguish between the different devices. (Bayar and Stamm 2018b) shows greater robustness with the enhanced processing compared to (Bayar and Stamm 2017a) (without enhancement). Overall, the results obtained for the *advanced* experiments, especially with *Dresden*, are far from what one should expect. Indeed, for the verification ($1 - to - 1$), a score of 50% corresponds to a random classifier. Our protocol shows that the current SOTA methods are not able to perform the verification for cameras with close fingerprints.

Selection	*Basic*	*Intermediate*	*Advanced*	**Drop (%)**
Methods	**SOCRatES**			
(Ding et al. 2019)	67.5%	66.6%	62.5%	**7.4**
(Bayar and Stamm 2018b)	81.4%	77%	69.5%	**14.62**
(Bayar and Stamm 2017a)	82.4%	78%	68.5%	**16.87**
(Mayer and Stamm 2020)	97.4%	92.5%	76.2%	**22.39**
Methods	**Dresden**			
(Ding et al. 2019)	59.9%	58.9%	50.5%	**15.69**
(Bayar and Stamm 2018b)	87.8%	71.1%	50.3%	**42.71**
(Bayar and Stamm 2017a)	89.9%	74.9%	50.3%	**44.05**
(Mayer and Stamm 2020)	97.8%	75.2%	49.8%	**49.08**

Table 23: Results of camera device verification for four SOTA methods. The reported metric is the AuC of the ROC in percentage: $AuC * 100$.

Overall, the SOTA methods are more robust when camera device verification is performed on *SOCRatES* than on *Dresden*: the relative drop is half as large. This is due to their different characteristics: *SOCRatES* is highly diverse with a camera-to-model ratio of 1.63 while *Dresden* has a ratio of 2.74. Furthermore, the t-SNE feature space for some smartphones from *SOCRatES* (see Fig. 10) shows that clusters can be more easily established than for *Dresden* (see Fig. 8). This may explain the different performance decrease between the two databases. However, even though it is smaller, the performance decrease on *SOCRatES* is also detected through the protocol with our selection of cameras. This selection highlights the fingerprints of close camera fingerprints, providing a more reliable assessment of the source camera verification. In particular, if the performance decreases too much from one difficulty level to another, it means that the method is not able to classify according to the valid feature (e.g., model for *intermediate* and device for *advanced*). Therefore, effective methods must achieve stable performance in each difficulty level. In addition, the higher the performance, the better (N.B. 50% means random classification).

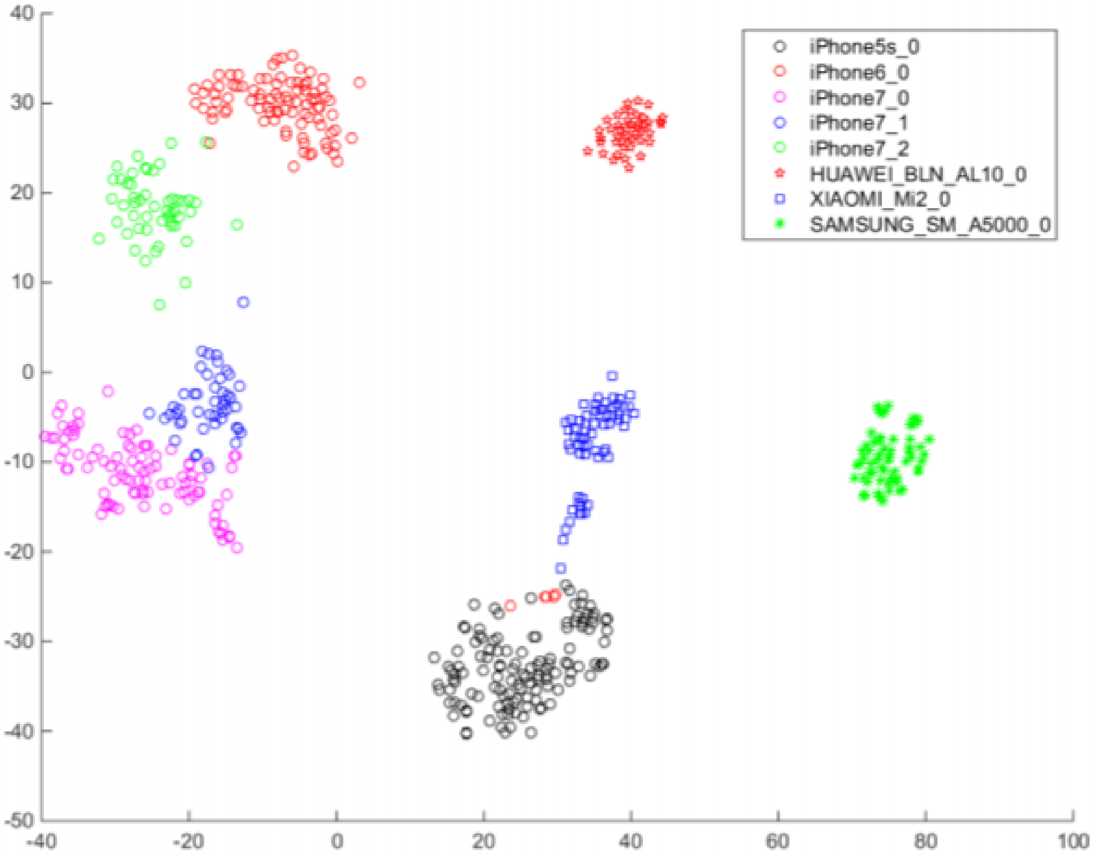


Figure 10: Visualization of the similarity of different cellphones in the feature space t-SNE (Ding et al. 2019). (circle) iPhone; (square) Xiaomi; (asterisk) Samsung; (star) Huawei.

Discussion

The results reveal a decrease in the performance of the tested SOTA methods in the *advanced* scenario (i.e., cameras of the same model), especially on the *Dresden Image Database*. Furthermore, with this protocol, the discrepancy between the *basic* and *advanced* levels confirms the robustness problem of the SOTA methods on different distributions of cameras in the dataset. Therefore, we defined a new, reliable and reproducible evaluation protocol to evaluate the source camera recognition methods. We analyzed and explained the problems with the evaluation protocols used in the literature, and proposed solutions to solve them.

Camera recognition is increasingly difficult as the label become more precise. In the specific case of source camera recognition based on DL, literature has widely addressed recognition of the camera model, while the recognition of the instance of the camera (i.e. device) is currently under-studied. Moreover, we have identified a lack of databases for performance assessment: State-Of-The-Art (SOTA) methods are usually assessed on databases that have specific compositions, such as the *Dresden Image database* (74 cameras of 27 models). However, using only one database for evaluation does not reflect reality. It may be necessary to analyze different sets of devices that are more or less difficult to classify. Also, for some scenarios, verification $(1 - to - 1)$ is better suited to camera recognition than identification $(1 - to - N)$. Based on these elements, we propose a more reliable and reproducible protocol for verification of the camera device. It is made of three different levels (*basic*, *intermediate* and *advanced*) of increasing difficulty, based on camera labels (brand, model and device). SOTA methods are tested with the proposed protocol on the *Dresden Image Database* and *SOCRatES*. The obtained results prove our assumptions, with a relative drop in performance, up to 49.08% between the *basic* and *advanced* difficulty levels. Our protocol is able to assess the robustness of methods for source camera recognition, as it tests whether they are really able to correctly classify cameras in realistic contexts.

4 AI-based Compression: a New Unintended Attack on DIF tasks

4.1 Compression Artifacts-based Tasks

With the rise of social networking and access to new technologies that make it easier to take them, pictures and videos have become commonplace in our daily lives. In parallel to this phenomenon, image editors have developed and are now easy to access and use, leading to potentially malicious modifications. These falsifications can affect different aspects of our society (political, social, etc.). Moreover, they are increasingly difficult to distinguish with the naked eye. Digital Image Forensics (DIF) is a field that provides tools for the blind analysis of images and the localization of certain falsifications. The main manipulations are splicing, where part of an image A is merged with an image B; and copy-move, also called cloning, where part of an image is copied onto itself. The location of these falsified regions is done by analyzing the artifacts that result from the process of creating a digital image. This process consists of three steps: acquisition, post-processing and storage. Notably, the storage stage includes JPEG compression, which creates artifacts in the image. The pixel blocks are converted to frequency space by the Discrete Cosine Transform (DCT) during the quantization step.

4.1.1 Double Compression Artifacts

These artifacts have been particularly used in the literature to detect malicious manipulation. Forgeries are created with image editors that often apply additional JPEG compression when saving the forged image, leading to doubly compressed images. As a result, the authentic and falsified areas do not have the same compression statistics, as shown by (Lukas and Fridrich 2003). In this context, they observed inconsistencies in the histograms of the DCT coefficients, with missing values and peaks, in the case of the double compression (Z. Lin et al. 2009). However, these histograms should follow a normal Gaussian distribution in the case of single compression. Based on these initial analyses, the detection of Double JPEG-Compression (DJPEG-C) has become an important topic of discussion within the forensic imaging community. Most of the methods were based on histogram analysis of the DCT coefficients.

There are different cases of double compression that have been discussed. In fact, the artifacts of double compression change depending on the Quality Factors (QFs) applied. The most likely case is when the 1^{st} (QF1) is different from the 2^{nd} (QF2). In the case of a similar quantization matrix (i.e., QF1 = QF2), no anomalies exist in the histograms, making detection much more difficult. Since this case is particularly difficult, there are a few papers on the identical quantization matrix (X. Huang et al. 2018). Similarly, there are two possibilities to apply double compression depending on the position of the blocks DCT : Non Aligned-Double JPEG-Compression (NA-DJPEG-C) (Bianchi and Piva 2012a) or Aligned-Double JPEG-Compression (A-DJPEG-C). Since these DCT blocks are 8×8, there is only one possibility for the 2^{nd} to be aligned with the 1^{st} (i.e., $\frac{63}{64}$ are not aligned). Of course, the double aligned compression JPEG is also a case to consider, although it is less common.

4.1.2 AI-based Compression

The Counter-Forensics (CF) literature particularly targets double compression artifacts, as they are specific and widely used for DIF tasks. The analysis made in the subsection 2.4 shows that compression traces are often targeted by attackers in order to hide or even remove them (Stamm et al. 2010). Therefore, the emergence of new solutions based on DL architectures for image compression, has opened up some questions about their impact on the tasks DIF. The JPEG organization has been investigating some of these AI-based methods in order to publish a new compression standard, called JPEG-AI, in the coming years. Our main hypothesis regarding the potential influences between AI-based compression and DIF is the impact on performance for certain forensic tasks, such as forgery detection. In fact, methods based on compression artifacts could be particularly affected by these new AI-based solutions.

4.1.3 Social Network Recognition

In addition, we live in a society where SNs images have become an integral part of our daily lives. Billions of images are exchanged every day on the Internet for a variety of purposes, some of which include malicious activity. Cyberbullying, incitement to violence, and psychological harassment are sometimes linked to media files exchanged via SNs such as, for example, WhatsApp, Facebook or Instagram. When a smartphone is confiscated from a suspect, an image can become criminal evidence, and therefore detecting the origin of that image can be really useful to help the investigation. Source SN recognition is an area of DIF, whose goal is to find out from which SN the images were downloaded. However, DL-based methods rely on specific artifacts such as DCT coefficients or the PRNU. Thus, AI-based compression could also affect the performance of these methods.

This chapter is studying the impact of AI-based compression on two particular DIF tasks, whose dedicated methods are using compression artifacts: forgery localization and SN recognition. The following subsections are detailing the elements that we selected to conduct our analysis.

The detection of falsified images is an important topic in the field of DIF. There are two main types of forgery: copy-move and splicing. These forgeries are created with image editors that apply a default JPEG compression when saving the forged images. As a result, the authentic and forged areas have different compression statistics, including histograms of DCT coefficients that show inconsistencies in the case of Double JPEG-Compression (DJPEG-C). There are different artifacts depending on how the double compression is applied: i) aligned/unaligned; ii) same/different Quality Factor (QF). Since the emergence of DL in image processing, AI-based compression methods have been proposed. The JPEG organization has reviewed these solutions within reach of a new compression standard: JPEG-AI. This could affect tasks based on compression artifacts, such as forgery detection or SN recognition.

4.2 Impact on Forgery Detection

The analysis of statistics relating to JPEG compression is therefore an important subject in DIF. Recently, with the rise of the Deep Learning (DL), some AI-based compression methods have emerged. These solutions were mainly based on auto-encoders, which are composed of two parts: 1) the encoder that reduces the input to a bottleneck containing the main features; 2) and the decoder that reconstructs the input from the bottleneck. With the emergence of an innovative compression process, the JPEG organization decided to evaluate these AI-based compression methods to create JPEG-AI[4] as the next image coding standard.

The goal of this new compression standard is to provide better compression for humans and machines. Thus, instead of having a single output (i.e., the reconstructed image), JPEG-AI aims to provide three solutions: the standard reconstruction, an image processing task (e.g., denoising) and a computer vision task (e.g., image classification) (Ascenso et al. 2020). This new compression format is expected to be available in the next few years (estimated in April 2024), as the JPEG-AI proposals will be presented and discussed at the 96^{th} JPEG meeting (July 2022). As a result, the field of DIF could be impacted, and in particular forensic image detectors that are based on JPEG artifacts. This new standard based on DL (a trendy field) could become the next democratized compression method. We are therefore the first to confront the two domains to study the impact of AI-based compression on forensic image detectors that are based on JPEG artifacts.

The purpose is to determine whether AI-based compression can be a potential unintended attack, in anticipation of the future JPEG-AI standardization. In the

[4] https://jpeg.org/jpegai/index.html

following subsection 4.2.1, we have reviewed the state of the art of such detectors in order to determine the most optimized method for our study. In the subsection 4.2.2, we detail the process of our method, as well as the models selected for this purpose. The results of our evaluation are presented in subsection 4.2.3.

4.2.1 Double Compression-based Forgery Detectors

The first methods based on compression artifacts, to locate falsifications, used the detection of DJPEG-C as a solution to find the falsified areas. (Bianchi and Piva 2012b) have taken into account A- and NA-DJPEG-C with a method based on the derivation of a unified statistical model characterizing the DCT coefficients to find the falsified areas. The result is a likelihood map of the images indicating whether the blocks are doubly compressed or not, which allows finding the falsified areas. (Barni et al. 2010) address splicing localization using NA-DJPEG-C detection, in the case of QF2 higher than QF1, with a region by region algorithm. With the development of DL in the last decades, deep architectures have been used for DIF, and thus for forgery detection. In particular, Convolutional Neural Network (CNN) (Krizhevsky et al. 2012) have been widely used for this task, with some pre-processing modules before or inside the network. In fact, DL-based methods for DIF require a pre-processing module to extract relevant artifacts that are overshadowed by the image content. In this subsection, we detail the State-Of-The-Art (SOTA) methods with their different architectures and pre-processing modules.

Aligned and Non-Aligned Double Compression Detection

(Wang and Zhang 2016) propose a method based on histograms of DCT coefficients, which are mainly used to detect DJPEG compressed images. As mentioned in the paper, the artifacts are handcrafted by concatenating the histograms before feeding the network. An interval is set to solve the problem of variable histogram size and reduces the computation with negligible loss of information, resulting in a vector of 99×1 to feed the network. Their architecture is based on a basic CNN with convolutional layers followed by three Fully-Connected (FC) layers for classification. Their model performed well in the case of NA-DJPEG-C, especially when QF2 was superior to QF1 and even for small patches (64×64).

(Barni, Bondi, et al. 2017) present the first method based on the CNN that extracts the artifacts, thanks to a pre-processing module integrated in the network. In fact, three pre-processing techniques are detailed: i) based on the pixel domain with the subtraction of the image mean (handcrafted); ii) based on the noise domain with the residual noise (handcrafted); iii) with the histograms of the DCT coefficients (embedded). The results show that the network based on handcrafted artifacts localizes better when it comes to A-DJPEG-C, while the CNN based on the embedded module is the best on NA-DJPEG-C. In addition, the embedded module-based CNN is able to work even with some basic processing operations.

General Cases

Although previous methods performed well, this was only in specific cases (including NA-DJPEG-C) and for certain quality factors (e.g. QF2 > QF1). (Park et al. 2018) propose a solution to detect DJPEG-C in general cases with mixed quality factors to localize splicing and copy-move. First, a new dataset dedicated to the detection of DJPEG-C is detailed, with the objective of being more realistic. They selected $1,120$ quantization tables (QF between 0 and 100) from JPEG images. These images were extracted from their forensic tool, which guarantees the authenticity of the images and is available on a public website to characterize real-world scenarios. To create their dataset, they applied single and double compressions to raw images by randomly selecting from these $1,120$ quantization tables. The method is based on histograms of DCT coefficients with an embedded module, and quantization tables that are reshaped into vectors and added to the classification part. These quantization tables, contained in the header file, are generally not used for detection because the quality factor is fixed, which is not the case here (QFs mixed). (Verma et al. 2020) follow the same process using the DenseNet architecture, which is fed by histograms of DCT coefficients, whose size has been calculated to be optimal.

The results obtained by the SOTA methods for DJPEG-C detection and the performance for forgery localization are respectively performed on the dataset from (Park et al. 2018) and on RAISE (Dang-Nguyen et al. 2015). All the results are summarized in the Tab. 24.

Methods	**DJPEG-C (Acc.)**	**Copy-move (F_1)**	**Blurring (F_1)**
(Wang and Zhang 2016)	73.05%		
(Barni, Bondi, et al. 2017)	83.47%	0.6323	0.6450
(Park et al. 2018)	92.76%	0.7704	0.7428
(Verma et al. 2020)	94.49%	0.7992	0.7744

Table 24: Performance of the SOTA methods for DJPEG-C detection (*accuracy*) and forgery localization (F_1-*score*).

4.2.2 AI-based Compression and Forgery Detector

Proposed Framework

The objective is to provide a first study on the combination of two areas that have never been confronted: DIF and compression based on AI. In particular, we want to analyze the impact of such recompression on forensic image detectors linked to JPEG. Recompression can degrade the artifacts used in falsification detection. This can occur when distributing images, either on SN or via messaging applications, as they apply compression. Thus, recompression, whether common (i.e., JPEG) or based on AI, is a non-malicious process, which could unfortunately affect forgery detectors. On the other hand, other post-processing operations (e.g., MF, GB, AWGN, etc.) are applied with the intention of degrading their performance. In this section, we want to study the impact of a possible unintended attack on forensic image detectors related to JPEG. Thus, we mainly focus on JPEG and AI-based recompressions, which are considered benign. The objective is to select the best methods in each domain and to confront them with a framework to evaluate whether AI-based compression can be considered as a new unintended attack on forensic image detectors related to JPEG. Our framework is based on three publicly available components: CAT-Net (detector[5]), HiFiC (compressor[6]) and *Casia v2* (database[7]).

Forgery Detector

CAT-Net (Kwon et al. 2021) is a detector capable of locating splicing and copy-move, based on DJPEG-C detection (93.93% *accuracy* on the (Park et al. 2018) dataset). It was evaluated on 6 databases for forgery detection and for robustness to recompression (with 4 QFs) and outperformed several methods in the literature. CAT-Net analyzes the DCT and RGB domains via two streams that respectively process the raw DCT coefficients of the Y-channel with a quantization table and the RGB image. Both streams use the HRNet architecture (Wang et al. 2020), which maintains high resolution representations, and a fusion step is applied to their outputs to obtain a prediction map. The RGB stream is the HRNet itself, while its first step is replaced by a JPEG learning artifact module for the DCT stream. We chose CAT-Net over the SOTA methods for three main aspects: 1) the use of the DCT volume representation, which preserves spatial information (better for localization); 2) the feeding of the network with raw DCT coefficients (instead of histograms); 3) the pre-training of the DCT stream on the DJPEG-C detection.

AI-based Compression

The literature on AI-based compression is quite recent. (Toderici et al. 2017) discuss compression with rational rates based on Recurrent Neural Networks (RNN)

[5] https://github.com/mjkwon2021/CAT-Net
[6] https://github.com/Justin-Tan/high-fidelity-generative-compression
[7] https://github.com/namtpham/casia2groundtruth

(LSTM, Gated Recurrent Unit (GRU), etc.) with a single training. (Ballé et al. 2018) also present an end-to-end network to improve the quality of the compression, especially the distortion rates. However, our choice is the High-Fidelity Compression (HiFiC) (Mentzer et al. 2020), which is the first AI-based compression method using a Generative Adversarial Networks (GAN). They present three aspects of their method that outperform the SOTA: 1) high perceptual fidelity close to the input, with half the bit rate; 2) applicable to high resolution images; 3) optimization of the method with different metrics (Peak Signal to Noise Ratio (PSNR), Multi-Scale Structural Similarity (MS-SSIM), etc.). In addition, they propose three different models with increasing quality: *low*, medium and *high*.

Database

Methods from SOTA (section 4.2.1) have been tested on *RAISE*, which contains only copy-move. However, *Casia v2* is dedicated to forgery detection with both splicing and copy-move. *CASIA v2* database contains $7,200$ authentic images and $5,123$ forged images of various sizes (320×240 to 800×600). As stated in (Dong et al. 2013), the ground-truth masks were available through a third party user (Pham et al. 2019).

4.2.3 Experimental Results

Dataset

Based on these elements, we decided to apply AI-based and common recompressions to images from *Casia v2*. All the original images are in JPEG format, which allows for high detection performance with CAT-Net. In accordance with our framework, we applied different versions of HiFiC, as well as JPEG compression (N.B. QFs from 50 to 80, with a step of 5) to these images. We have also included Additive White Gaussian Noise (AWGN) ($sigma = 5.1$), which affects the quality of the image in the same way as HiFiC-*high*, to give a reference against malicious operations.

Metrics

To evaluate our experiment, we used the same metrics as for CAT-Net (Kwon et al. 2021), based on binary segmentation with True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Thus, we used the *accuracy* (see Eq. 17) for authentic images, while we computed the F_1 *score* (see Eq. 18) which emphasizes the positive class for forged images. Because *accuracy* and F_1 *score* depend on a fixed threshold, they also used *Average Precision (AP)* (area under the *recall-precision* curve), which is a non-threshold performance.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (17) \quad \text{and} \qquad F1 = \frac{2TP}{2TP + FN + FP} \quad (18)$$

The Tab. 25 shows the results of our experiment on *Casia v2*. On the one hand, the *accuracy* is quite high regardless of the compression quality, which means that genuine images are not affected by recompression. On the other hand, according to the F_1 *score*, AI-based compression has a more negative impact than JPEG compression. The AP gives additional insight into each operation, with varying results depending on the parameters chosen. Low QFs have more impact on localization than high ones for JPEG compression, while it is the opposite for AI-based compression.

Operations		**Objective Quality**		**Forged Image**		**Authentic**
		PSNR (dB)	**SSIM**	**AP**	F_1	**Accuracy** (%)
No Compression				0.94	0.79	88.48
JPEG	**50**	**33.7**	**0.915**	0.45	0.35	83.88
	55	**33.9**	**0.92**	0.46	0.33	86.44
	60	34.14	0.926	0.49	0.32	86.67
	65	34.41	0.932	0.58	0.37	90.76
	70	34.78	0.938	0.96	0.40	91.24
	75	35.19	0.945	0.90	0.38	87.98
	80	35.81	0.954	0.84	0.43	91.56
HiFiC	**Low**	31.9	0.787	0.67	0.17	92.11
	High	**33.75**	**0.901**	0.20	0.12	92.25
AWGN	$\sigma = 5.1$	**33.81**	**0.861**	0.3	0.28	80.97

Table 25: Results of forgery localization, with *accuracy* (%), F_1 *score* and AP, according to various operations. Objective quality of processed images is furnished (PSNR, SSIM). original - important drop - the hugest drop.

Overall, when comparing the two compressions with equivalent objective image quality (PSNR, Structural Similarity (SSIM)), localization performance is more affected (at least twice as much) by AI-based compression than by JPEG or even a malicious operation such as AWGN (orange vs. red in the Tab. 25). Moreover, HiFiC has been optimized to reduce the bit rate (half as much as SOTA methods), while preserving a high visual quality of the image (see Fig. 11). Therefore, HiFiC-*high* is able to overcome the detector without compromising the image quality.

Conclusion

This is the first study to jointly address AI-based compression and Digital Image

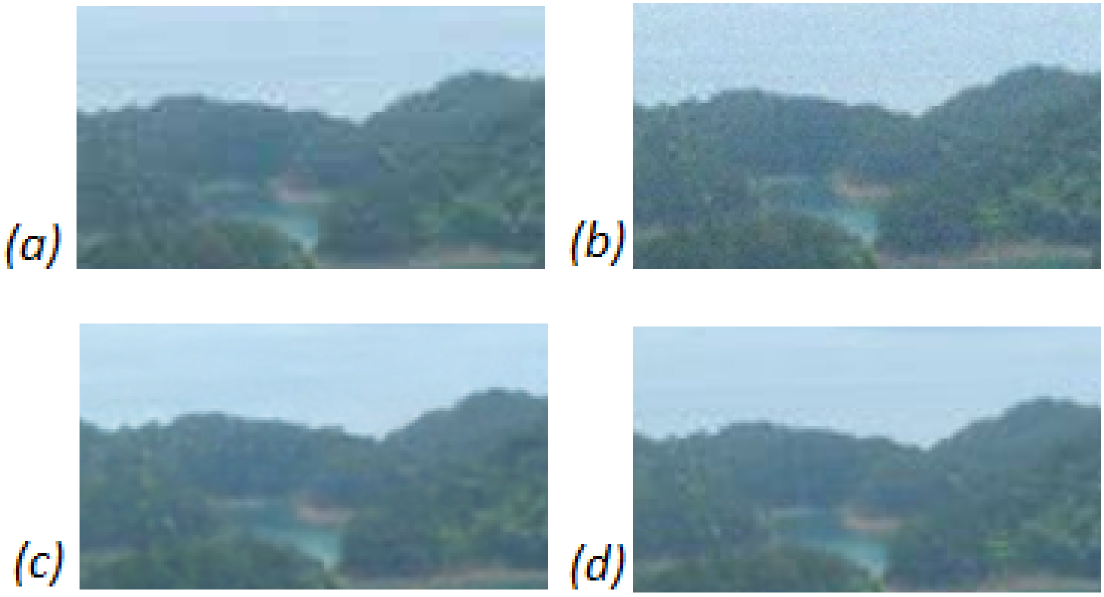


Figure 11: Comparison of visual image quality of each operation (same objective quality) with a region of an image from *Casia v2*. (a) QF50 (b) AWGN (c) HiFiC-*high* (d) original.

Forensics (DIF). We applied High-Fidelity Compression (HiFiC), JPEG compression and AWGN to *Casia v2* images in order to compare their impact on false localization. Our result shows that HiFiC-*high* is the most effective operation to lead to a considerable decrease in performance while maintaining high visual quality of the images. AI-based compression is a new unintended attack for JPEG-related forgery detectors and should be considered in future studies on image forensics. In particular, the following sections evaluate its impact on the recognition of Social Network (SN), which is another topic based on JPEG artifacts.

The detection of DJPEG-C is an important topic for JPEG-related image forensic detectors. Since the emergence of DL in image processing, AI-based compression methods have been proposed. This study is the first to consider AI-based compression with digital image analysis tools. The objective is to understand whether AI-based compression can constitute a new unintended attack for JPEG-related image forensic detectors. To test our hypothesis, we selected the best detector to date, an AI-based compression method and *Casia v2* that contains both splicing and copy-move (all publicly available). We focused our experiment on benign post-processing operations: JPEG and AI-based recompressions (with different quality levels). The evaluation is performed using different metrics (*acrshortap*, F_1 *score* and *accuracy*) to take into account both the impact on detection and image quality (PSNR, SSIM). For similar image quality, AI-based recompression achieves at least twice the performance reduction of JPEG, while maintaining high visual image quality. Thus, AI-based compression is a new unintended attack, which can no longer be ignored in future studies on DIF.

4.3 Two-stream Network for Social Network Recognition

The objective of this section is to propose and develop a methodology to identify the source Social Network (SN) of an image. Each SN has its own fingerprint, which depends on its processing algorithm. Therefore, the main idea is to analyze the artifacts that are left on an image to identify the source. The identification is done blindly, using only the data extracted from the image, without relying on the image header data, which can be easily removed or edited without modifying the image content.

The presented methodology uses two types of feature domains, where artifacts are more easily detectable: the DCT domain and the Photo-Response Non-Uniformity (PRNU) domain. The identification is performed by a trained Deep Neural Network (DNN) in two steps: the artifacts are first extracted, then used for classification. The method is tested on three publicly available datasets. The proposed DNN network is called *two-stream* network, because it takes as input two sets of features (DCT and PRNU). First, the two feature sets are extracted independently of each other. Then, they are concatenated, and finally used for the classification of the source SN.

In the following subsections, we present methods from literature that performed SN recognition (see subsection 4.3.1). Then, we detail the methodology of our proposed network in subsections 4.3.3 and 4.3.2, as well as its architecture (see subsection 4.3.4). Finally, the results are reported in the subsections 4.3.5 (single stream) and 4.3.6 (two-stream).

4.3.1 Related work

Many image processing tasks have achieved excellent performance using Deep Learning (DL). (Roy et al. 2017; Agarwal et al. 2020; Pan et al. 2020). In particular, CNNs have shown great potential and are becoming the standard for solving image problems (Shin et al. 2016). When considering origin recognition from an image, the literature typically refers to *camera recognition*, rather than SN recognition. As we show in the chapter 3, the problem of recognizing the original camera has been widely addressed by researchers in image forensics (Lukas et al. 2006; Balamurugan et al. 2017; Galdi et al. 2019). DL have been adopted by the majority of the most successful methods (Roy et al. 2017; Cozzolino and Verdoliva 2020).

Non-Blind Approach

Source SN recognition from an image is a relatively new and unexplored area of research. Some previous works approach the problem by considering different features and are based on several assumptions. For example, (Giudice et al. 2017) use the filename and metadata, as well as resizing and recompression factors, to identify the source SN using a k-Nearest Neighbors (k-NN) classifier. In this case, they assume that metadata and filenames can be trusted, thus proposing a so-called *non-blind* approach. We are instead interested in a *blind* approach, which trusts only the image content.

Blind Approaches

The number of blind SOTA approaches is very limited and all dedicated to SN identification. Two works are based on the analysis of the characteristics of the DCT blocks. The first one, (Caldelli et al. 2017) propose the use of DCT-block features and a bagged decision tree classifier. The second (Amerini, Uricchio, and Caldelli 2017) also use DCT-block features, but with a CNN. Another blind approach is proposed, with the use of residual noise and a CNN to identify the source SN (Caldelli et al. 2018).

SN processing

These methods are based on the assumption that once an image is uploaded to a SN, it undergoes processing that leaves artifacts on it. These artifacts create a distinctive pattern, usually called a *fingerprint*. For the DCT-block domain, JPEG quantization causes a perturbation of the distribution of coefficients. For the residual noise domain, the Photo-Response Non-Uniformity (PRNU) of the image is modified by the processing proper to each SN.

Two-Stream Network (TSN)

With respect to TSNs, articles from the literature dealing with source SN identification has not yet exploited this architecture. However, in other areas of DIF, the TSN architecture has proven its efficiency. For example, (Amerini, Uricchio, Ballan, et al. 2017) present a method for detecting DJPEG-C. It is based on the analysis of two different features from the spatial and frequency domains, including DCT-block features. (Zhou et al. 2018) propose another TSN for the detection of image manipulations by analyzing the original image and its local noise features. In the following subsections 4.3.2 and 4.3.3, we present two domains that we exploit for our TSN.

4.3.2 DCT block domain

Computing DCT

The DCT-blocks can be generated with the Y-channel of the $YCbCr$ color space of the input image. This one is divided into 8×8 blocks of pixels, then the DCT is calculated for each block. The original image can be reconstructed by applying the reverse process, using the Inverse Discret Cosine Transform (IDCT). JPEG compression uses this technique to reduce the size of the image on disk by storing only the DCT-block coefficients instead of the plain RGB data. Before storing the DCT coefficients, the JPEG compression applies quantization by dividing each coefficient by a predetermined value in the quantization table. The results of the division are then rounded to the nearest integer and compressed with Huffman coding.

DCT coefficients

In our case, the crucial aspect of JPEG compression is the analysis of the distribution of the DCT-block coefficients, which allows us to detect quantization artifacts. Each SN applies its own series of transformations, which modify the distribution of the DCT coefficients (see Fig. 12). Thus, we can train our CNN on these types of features to identify the source SN. However, feeding the network directly with the raw DCT coefficients is not ideal. Indeed, the quantization leaves traces that are difficult to detect by a CNN. It is therefore recommended that the DCT-block coefficients be coded in a way that makes quantization artifacts more detectable to the CNN.

Histograms of DCT

(Amerini, Uricchio, and Caldelli 2017) propose the use of a histogram-based approach that encodes the DCT coefficients by counting the occurrences of a given

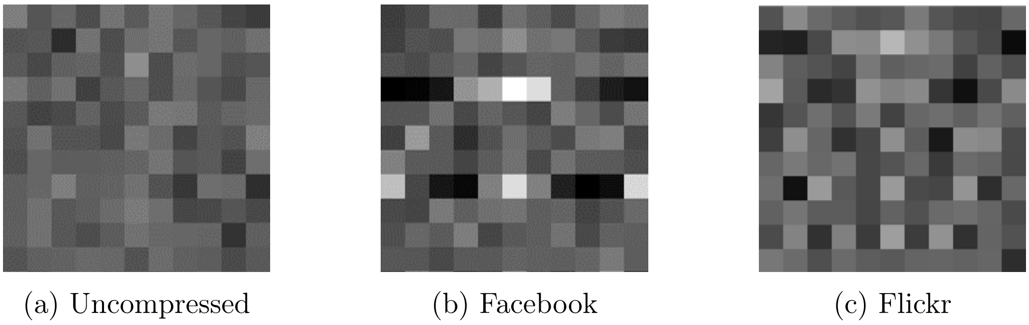


Figure 12: Comparison of the DCT-block features: (a) original; (b) after Facebook processing; and (c) after Flickr processing.

value in the blocks (see figure 13). The image is first cropped into non-overlapping patches of size $N \times N$ to avoid repercussions in the DCT, which are affected by the content and size of the image under consideration (Wang and Zhang 2016). The DCT is then computed for each patch. Finally, for each 8×8 DCT block, the first 9 spatial frequencies in the zig-zag scan order are selected to compute the histogram.

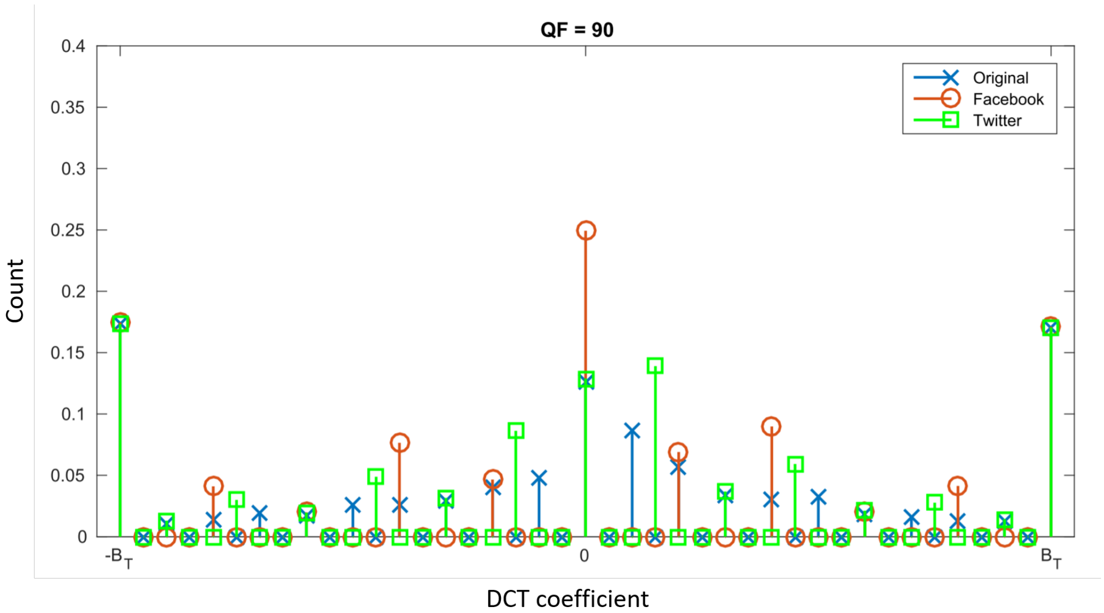


Figure 13: Example of DCT-block coefficient distribution histogram for the same image processed by different SNs. Image taken from (Caldelli et al. 2017).

For each spatial frequency (i, j), the histogram $h(i, j)$ representing the occurrences of a value of the quantized DCT coefficients is constructed. The histogram ranges from -50 to $+50$ (101 bins), because most of the coefficients fall within these values. So, the output of the encoding is $101 \times 9 = 909$ values. Coefficients superior to 50 or inferior to -50 are counted in the last and first bin, respectively (Caldelli et al. 2017). Thanks to this encoding, the pattern due to quantization is extracted and made detectable for a CNN.

Enhanced Encodings

Even if the encoding proposed in the previous work (Amerini, Uricchio, and Caldelli 2017) gives good results, we have studied other encoding methods in order to further improve the performances. One of the limitations of the previously proposed encoding is the loss of information about coefficients superior to $+50$ or inferior to -50. An easy solution would be to increase the range of the histogram, but this would not be ideal with a distribution of small number of values for numerous bins.

We therefore developed a different encoding scheme that makes quantization artifacts even more detectable and uses all values of the DCT-block. This encoding is based on normalizing the DCT coefficients in a limited range of values, before the calculation of the histograms. The normalized value x_n is defined from a coefficient value x and a quantization factor q (see Eq. 19). The normalized values range from: i) 0, indicating that it is likely that the x value was quantized with the q factor; ii) to ± 0.5, indicating that it is rather unlikely. We therefore define the features by computing histograms of the DCT-block coefficients $hist_{p,f}$, for each patch p and spatial frequency f (see Eq. 20).

$$x_n = \frac{x}{q} - round(\frac{x}{q}) \quad (19) \quad \text{and} \quad hist_{p,f}(\frac{x_{p,f}}{q} - round(\frac{x_{p,f}}{q})) \quad (20)$$

Vector Size

The set of histograms extracted from a patch describes the quantization artifacts. It can be used by the CNN to classify the source SN. The disadvantage of this encoding procedure comes from the quantization factor, which is not known in advance. Therefore, we have to try all possible values of the quantization factor q up to 20 to detect artifacts. We selected only the first 9 spatial frequencies of the DCT-block, as in the literature (Amerini, Uricchio, and Caldelli 2017). The values of q for these coefficients are small, so we can set the maximum value to 20. For each value of q ($q = 1, 2, 3, ..., 20$), we compute the histogram of the DCT-block coefficients, ranging from -0.5 to 0.5 (see Eq. 20), with a number of bins equal to 11 (one bin for each decimal interval from [-0.5 to 0.5]). For each image patch, the computed histograms are concatenated to form the feature vector. The final length of this encoding vector corresponds to 20 possible quantization factors by 11 bins by 9 spatial frequencies, which is equal to 1980 values. As demonstrated by the experimental evaluation reported below, the proposed encoding procedure achieves very good performances.

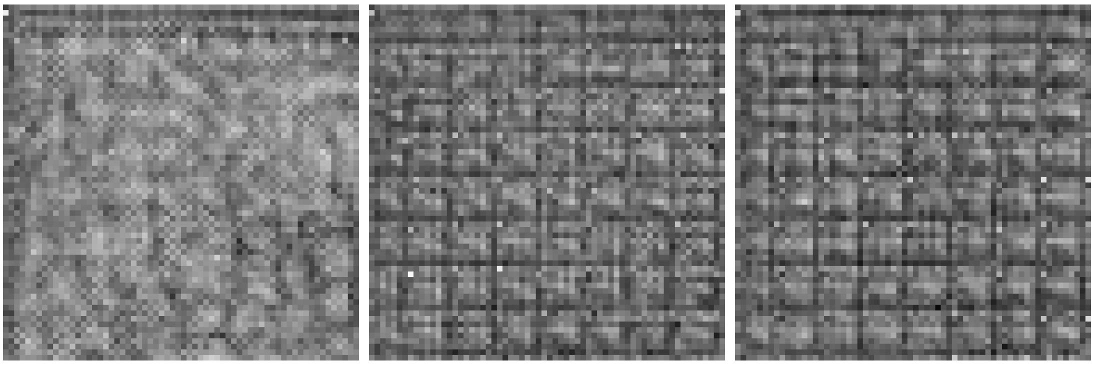
4.3.3 PRNU domain

PRNU Literature

The Photo-Response Non-Uniformity (PRNU) is a distinctive pattern due to imperfections in the silicon wafer during the manufacture of the sensor, different even among cameras of the same model. These imperfections imply that the pixels have different sensitivities to light. PRNU is generally used for the recognition of the source camera (Lukas et al. 2006; Balamurugan et al. 2017). For extraction of PRNU, a recent work provides a pre-trained CNN for image forensic task, called Noiseprint (Cozzolino and Verdoliva 2020). Noiseprint generates a camera fingerprint pattern that has proven to be very useful for image forgery detection. Previous work (Caldelli et al. 2018) has shown that PRNU can be used for an accurate classification of the source SN.

PRNU Modification

In fact, the processing algorithm of a SN affects and slightly modifies the PRNU pattern (see figure 14), but not enough to remove it completely. Therefore, the analysis of the PRNU can help detect artifacts to identify the SN source from images. The features extracted in this domain are different and independent of those from the DCT-blocks. Thus, the combination of both could improve the classification of the source SN.



(a) Uncompressed (b) Facebook (c) Flickr

Figure 14: Comparison of a patch 64 × 64 from Noiseprint. (a) original, (b) Facebook, (c) Flickr.

PRNU Extraction

The PRNU is constant for images taken by the same camera sensor. Hence, the standard approach for its extraction is to apply a denoising filter on multiple images from the same camera. When averaging the extracted noise patterns, the zero-mean Gaussian distributed noise tends to disappear, while only the Sensor Pattern Noise (SPN) remains. There is a clear limitation to this method. To get an accurate extraction of the PRNU, numerous images from the same sensor are required. (Caldelli et al. 2018) apply the same approach and perform image source SN identification based on residual noise extraction. In order to address this issue, they use the residual noise from a single image. Exploiting this approach is quite unstable. In fact, the residual noise is strongly affected by the image content. Moreover, it contains a considerable amount of noise that does not come from the SN uploading process. All these elements make the identification more difficult.

DL-based Extraction

We approach this problem differently by using Noiseprint (Cozzolino and Verdoliva 2020). The scene content is largely removed, and camera-related artifacts are enhanced to create residual noise. Although Noiseprint does not directly extract the PRNU, the generated pattern can be considered as related to the PRNU. In fact, it correlates with the camera sensor artifacts present in the image. The authors suggest that the Noiseprint pattern may be useful for image forensics tasks. We therefore decided to test this network for the identification of SN. We present the results of this evaluation as well as the comparison with previous work based on residual noise (Caldelli et al. 2018) in the subsection 4.3.5. The next subsection 4.3.4 presents the architecture based on the two domains: PRNU and DCT coefficients.

4.3.4 Two-stream Network

We assume that the images uploaded to a SN undergo a certain processing, which leaves particular traces. In particular, JPEG compression and resizing are the most commonly used manipulations. The first is a lossy compression algorithm that compresses the image by quantizing the coefficients of the DCT-block before storing them using the Huffman coding. JPEG quantization disturbs the distribution of the DCT-block coefficients, creating artifacts that can be used to determine the source SN. Resizing is the process of reducing the size of the image into a smaller image through pixel interpolation. Although it is more subtle, interpolation can also create artifacts. Sometimes additional manipulations can be applied to compensate for other transformations that can cause blurring. All of these types of processing generate artifacts in the resulting image, which may be more or less detectable depending on the area being analyzed. These artifacts are usually extracted using pre-processing modules before being introduced into a neural network. In this subsection, we detail the architecture chosen to deal with the artifacts created by the processing of SN.

Proposed Method

Our proposed method combines two domains in which artifacts from image processing of a SN can be detected: DCT-block and PRNU. We assume that the combination of these two domains can improve the diversity of artifacts. CNN is used to perform feature extraction for both domains. We call this approach Two-Stream Network (TSN). Basically, two domains are analyzed separately and their respective CNN are fed by two inputs from the same image. The two inputs pass through a separate and different set of convolutional layers before being concatenated and classified (see figure 15).

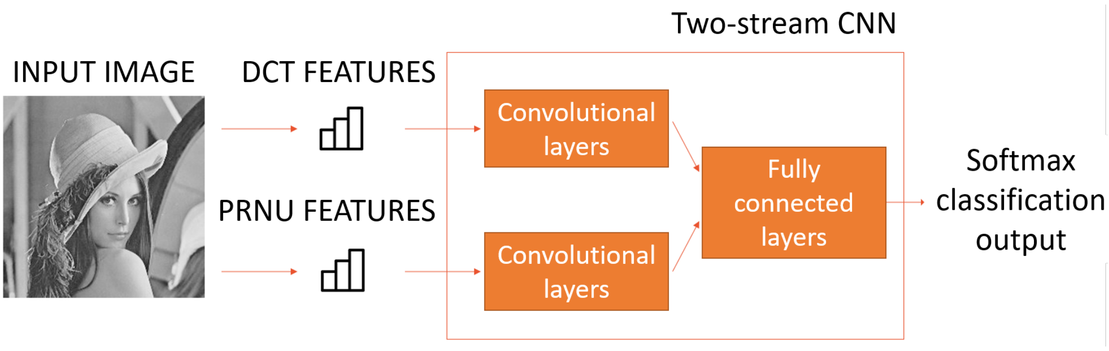


Figure 15: Scheme of the proposed TSN.

CNN Design

The structure of the TSN is developed by firstly designing and evaluating each stream separately. The proposed architecture is illustrated in Figure 16. The design of each stream is different because the input data is of different nature and size. Therefore, the size and number of layers are not the same. The datasets were divided into training, validation and test (80 : 10 : 10). For the DCT stream, both encoding methods (i.e., ours and SOTA) were tested. With this setup, we conducted the hyperparameter tuning phase by introducing intuitive and random changes. At each change, the performances were evaluated using the validation set. The objective of this tuning is to find the best model structure.

The CNN has two inputs: an encoded DCT-block and an image patch of size 64×64 (for Noiseprint). Each input passes through a series of convolution layers

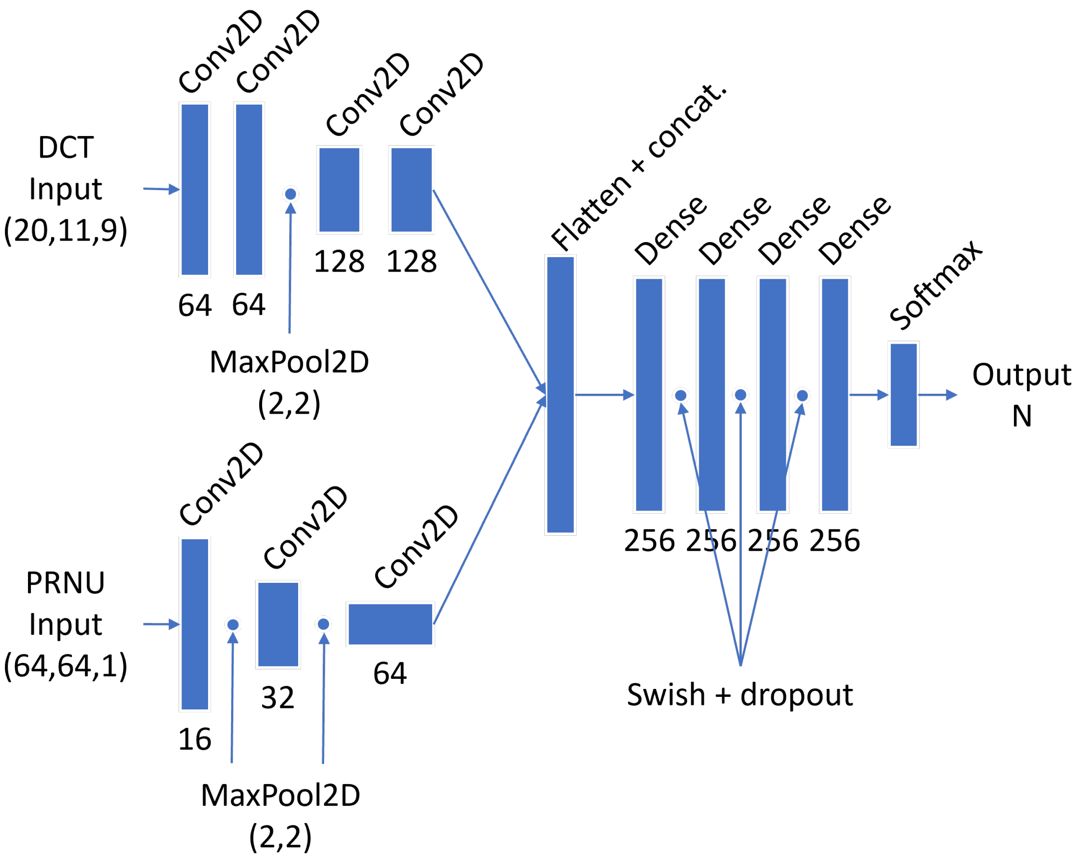


Figure 16: Proposed TSN architecture.

with 3 × 3 kernels, ReLU activation and max pooling layers. The number of filters in the convolutional layer is increased at each max pooling layer. Batch normalization is applied before each convolutional layer, except the first one. This part of the network is based on CNN, which usually applies such layers. The two streams are then flattened and concatenated before the classification layers. The classification layers are Fully-Connected (FC) layers (also called Dense) and use the activation function *swish* (see Eq. 21, where β is a constant).

$$swish(x) := x \times sigmoid(\beta x) = \frac{x}{1 + e^{-\beta x}} \tag{21}$$

Dropout is applied before each FC layer, except for the first and last. The output of the network is a FC layer with *soft max* activation and size N given by the number of classes (3 in our experiments). The loss function is *categorical crossentropy*, and the optimizer is Nesterov-Accelerated Adaptive Moment Estimation (Nadam).

Database

The proposed method adopts a supervised learning technique. Therefore, a labeled dataset is required to train the TSN. The number of publicly available labeled datasets for the source SN is limited, as is the number of works addressing this problem. For the sake of comparison with SOTA, we decided to use the same three datasets as in the previous works.

UCID Social : this dataset is generated by taking the $1,338$ images from the *UCID* database (Schaefer and Stich 2004). All images are compressed with 10 different JPEG Quality Factor (QF) (from 50 to 95 with steps of 5). The images are then uploaded to three SNs (Facebook, Flickr and Twitter) and downloaded. The number of images in the three classes is therefore: $1,338 \times 10 \times 3 = 40,140$. The dataset also contains multi-class images, which were first uploaded to one SN (e.g., Facebook) and then downloaded and reloaded to another SN (e.g., Twitter). These images can be used for multi-class origin identification, which is outside the scope of this research.

IPLab[8]: this dataset is generated by taking 240 images and uploading them to 8 image sharing services, including SNs and messaging applications. The images are then downloaded from these services. In our study, we use only three SNs for our research, namely Facebook, Flickr and Twitter.

Social Public[9]: this dataset is generated by downloading $1,000$ images from three target SNs (Facebook, Flickr, Twitter).

UCID Social and *IPLab* are *controlled environment*, meaning that the data was generated by: i) taking a set of images and uploading them to each SN; ii) then downloading them to create a dataset where each class contains the same original images that have undergone different processing. *Social Public* is an *uncontrolled environment dataset*, which means that the data was generated by downloading a random set of images from the SNs. Thus, the set of images for each class (i.e., SN) is different.

Dataset balance

In order to train a DNN, it is advisable to have a balanced training set, where the number of samples per class is equal for each class. Otherwise, the model might converge by favoring the most represented class. In each of the three selected datasets, the data are perfectly balanced in terms of images: i) $13,380$ images per class in *UCID Social*; ii) $1,000$ images per class in *Social public*; iii) 240 per class in *IPLab*. However, since our training samples are patches of 64×64 pixels, we need to ensure that the datasets are equally balanced at the patch level. Processing

[8] https://iplab.dmi.unict.it/popularitychallenge/

[9] http://lci.micc.unifi.it/labd/2015/01/trustworthiness-and-social-forensic/

through SNs can include resizing. Thus, the final number of patches extracted is going to be different for each class (see figure 17). An unbalanced data set is not ideal for training our TSN.

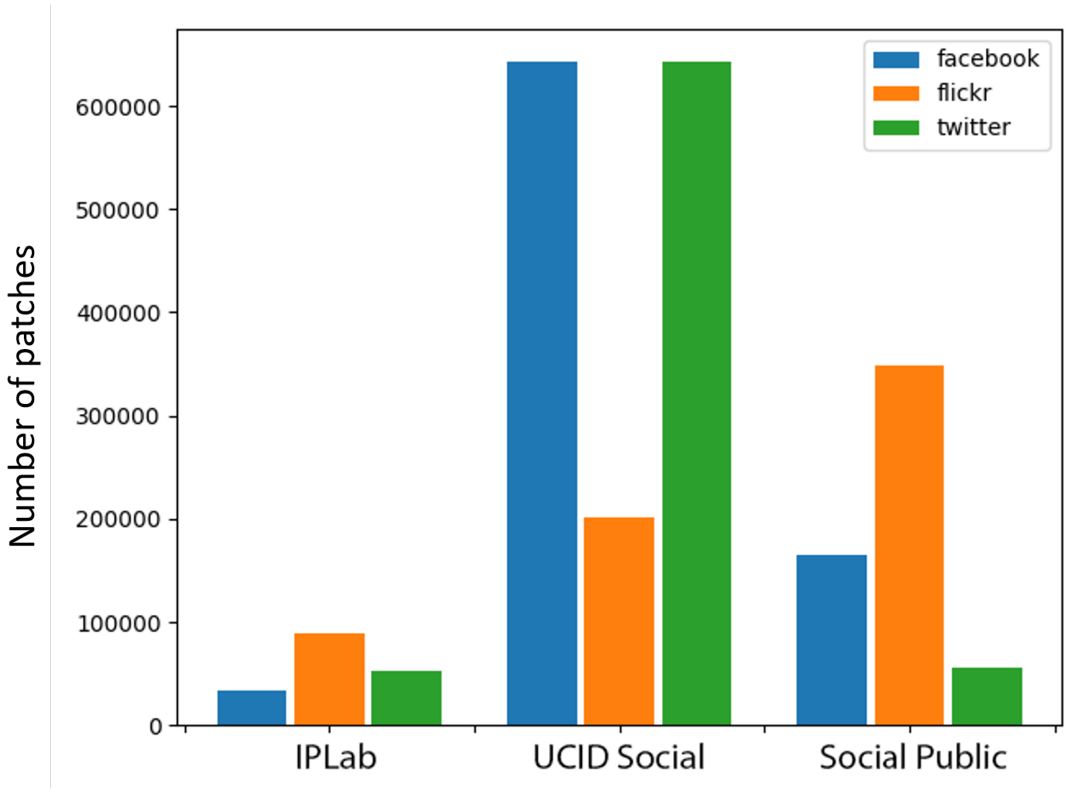


Figure 17: Number of patches per class, for each of the three datasets.

To solve this problem, for a given training dataset, we create an infinite-loop generator for each class in the dataset (Facebook, Flickr, and Twitter). We then interleave the three class generators, taking a sample from each class. In this way, we create an infinite generator of perfectly balanced batches of training data. The same problem of class balance can also arise for the validation and test sets, as the measures used (e.g., *recall* and *accuracy*) can give misleading results if the validation classes are not balanced. We therefore made the validation and test results balanced by defining class weights.

A class weight is a multiplicative factor applied to the calculation of a metric or loss function associated with each sample in a given class. For validation and test sets, the total weight of each class is first calculated by summing the sample weights. If the total class weights are different, a corrective class weight factor f_k is multiplied to each sample weight for each class. More details on class weights will be given in the section 4.3.6.

4.3.5 Single-stream Evaluation

Each dataset (*UCID Social*, *Social Public*, and *IPLab*) is split at the image level in three subsets with the common split (80 : 10 : 10) and evaluated separately.

PRNU-based method

Before evaluating the TSN, the single streams are evaluated separately to validate the developed methods. First, we verified our assumption regarding the use of Noiseprint. Indeed, we want to be sure that the Noiseprint-based stream is appropriate for identifying SN. We therefore evaluated its performance against the residual noise based method (Caldelli et al. 2018). In order to provide a direct comparison, we reproduced the protocol of the SOTA method. They evaluated their method on the *UCID Social* dataset (Facebook, Flickr and Twitter) with the metric *precision* (see Eq. 22, with True Positive (TP) and False Positive (FP)).

$$precision = \frac{TP}{TP + FP} \tag{22}$$

The evaluation is performed on the test set of the patch-level dataset *UCID Social.* The Stratified Repeated Random Sub-sampling Validation (SRRSV) is applied with 5 iterations to validate the performance results. The Noiseprint-based stream achieves an average *accuracy* of 90% and outperforms the SOTA method (about 80%) (see table 26). The results suggest that Noiseprint works very well as a feature extractor for source SN identification.

UCID Social	*Facebook*	*Flickr*	*Twitter*
(Caldelli et al. 2018)	72.80%	93.15%	72.49%
Single-stream Noiseprint	**87.6%**	**95.8%**	**91.5%**

Table 26: Comparison of patch-level classification on *UCID Social* between Single-stream Noiseprint and (Caldelli et al. 2018).

DCT-based method

We also evaluate the DCT-block-based single-stream alone. For this evaluation, the proposed method is compared to the DCT-based SOTA method (Amerini, Uricchio, and Caldelli 2017). Both CNN are very similar, with a difference only in the processing of DCT-based input. Therefore, we decided to determine which of the two DCT-block encoding techniques performs better. We compared the performance of the two encoding approaches with *recall* (see Eq. 23).

$$recall = \frac{TP}{TP + FN} \tag{23}$$

We recognize only a slight performance improvement (*recall* difference < 0.5%) for our enhanced encoding approach. In addition, the convergence speed of the model is faster when we use our encoding. It requires about 20% fewer epochs in the training phase. Because of these two advantages, we adopt our encoding for the TSN.

4.3.6 Two-stream Evaluation

For the evaluation of TSN, we follow a similar approach to the evaluation of the single stream. We evaluate the classification performance for the three classes (from the three databases): Facebook, Flickr and Twitter. In addition, for comparison purposes with SOTA, we adopt the same protocol as (Caldelli et al. 2018). We also perform the evaluation in two steps: first, we evaluate the classification performance at the patch level and then at the image level.

Patch-level evaluation

First, we present the results of the patch-level evaluation. For comparison with the residual noise-based SOTA method, we report the classification *accuracy* in the Tab. 27. The results show that our method achieves better performance values than the SOTA method and has further improved the classification *accuracy* compared to the single-stream Noiseprint method.

UCID Social	*Facebook*	*Flickr*	*Twitter*
(Caldelli et al. 2018)	72.80%	93.15%	72.49%
TSN	**99.32%**	**99.67%**	**97.83%**

Table 27: Comparison of patch-level classification on *UCID Social* between TSN and (Caldelli et al. 2018).

To have a direct comparison with the results of the DCT-based SOTA method (Amerini, Uricchio, and Caldelli 2017), we provide the confusion matrices for the three datasets. The results are validated with a 5-repeat SRRSV. The performance of our TSN at the patch-level is superior to that of the method SOTA (see Tab. 28).

Image-level Evaluation

Considering the original question of source SN identification, we need to perform the evaluation at the image-level. Thus, we simply extract all non-overlapping patches of size 64 × 64 from the image, and classify each of them using networks. Then, we consider the majority-voted class as the predicted class for the whole

UCID Social	**TSN**	**SOTA**	**TSN**	**SOTA**	**TSN**	**SOTA**
	Facebook		*Flickr*		*Twitter*	
Facebook	**97.60%**	96.15%	0.20%	0.19%	2.20%	3.66%
Flickr	0.00%	0.03%	**100%**	99.79%	0.00%	0.18%
Twitter	0.67%	0.59%	0.13%	0.11%	**99.30%**	**99.30%**

Social Public	**TSN**	**SOTA**	**TSN**	**SOTA**	**TSN**	**SOTA**
	Facebook		*Flickr*		*Twitter*	
Facebook	**97.30%**	94.00%	2.52%	6.00%	0.18%	0.00%
Flickr	7.11%	1.76%	89.61%	**92.13%**	3.28%	6.11%
Twitter	0.10%	0.00%	3.49%	13.47%	**96.41%**	86.53%

IPLab	**TSN**	**SOTA**	**TSN**	**SOTA**	**TSN**	**SOTA**
	Facebook		*Flickr*		*Twitter*	
Facebook	**97.30%**	94.00%	2.52%	6.00%	0.18%	0.00%
Flickr	7.11%	1.76%	89.61%	**92.13%**	3.28%	6.11%
Twitter	0.10%	0.00%	3.49%	13.47%	**96.41%**	86.53%

Table 28: Confusion matrices of patch-level classification between TSN and (Amerini, Uricchio, and Caldelli 2017), on three databases: *UCID Social*, *Social Public* and *IPLab*.

image. However, the distribution of the number of patches per image (i.e., image size) is different for each class (Facebook, Flickr, Twitter) in the three datasets (see Figure 17). As a result, some classes may be underrepresented compared to the others. For example, in *Social Public*, Twitter has a much lower overall number of patches than the other SNs (see figure 18).

This unbalanced partition can lead to bias in the model, as it will learn features from some classes over others. To address this problem, we change the weight of each training sample to be inversely proportional to the number of N patches in an image (i.e., $\frac{1}{N}$). However, since we want the overall weight to remain balanced for each class, we also multiply the weight by the average number of patches per image. The final weight for a sample w_{Nk}, from an image with N patches of class k, is thus defined by the number of patches $|P_k|$ and the total number of images I_k of class k (see Eq. 24).

$$w_{Nk} = \frac{|P_k|}{|I_k|} \cdot \frac{1}{N} \tag{24}$$

Then, we again exploit the infinite-loop generator (introduced in subsection 4.3.4) to obtain an equal number of samples for each class in the dataset. The definition of the weights should be modified by taking into account the frequency

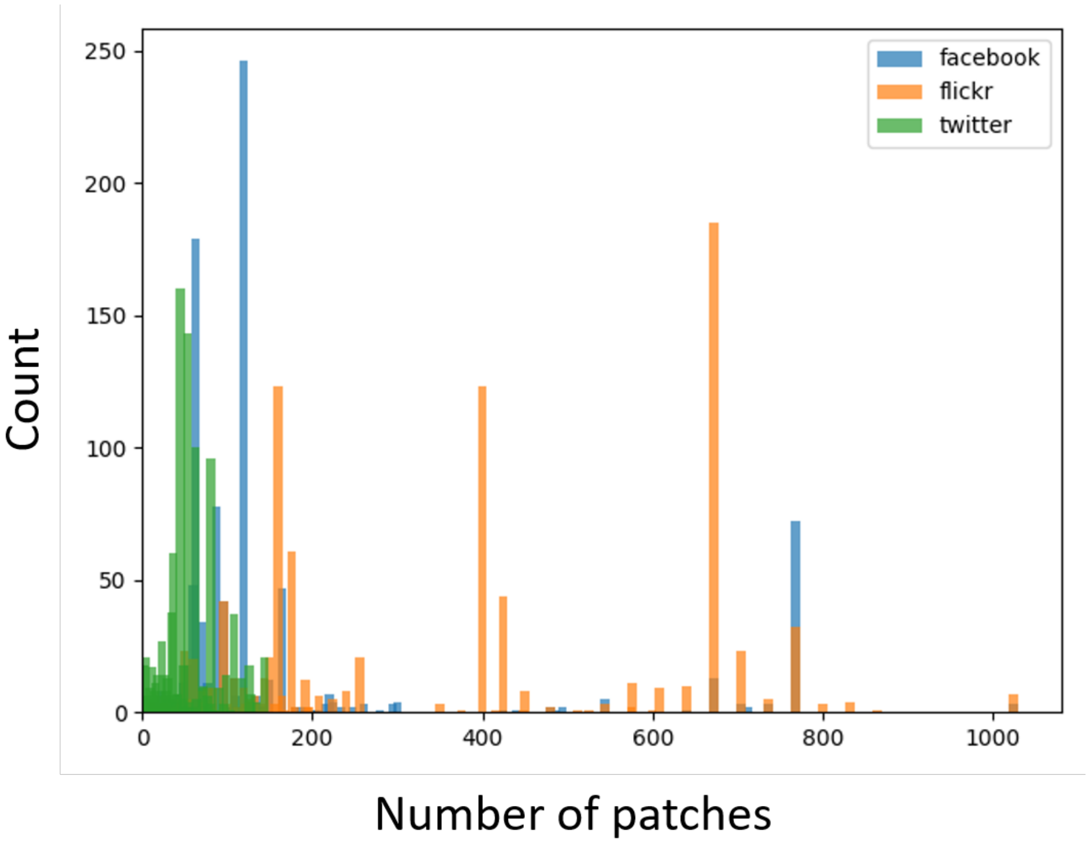


Figure 18: Comparison between the distribution of the number of patches per image in the Facebook, Flickr, and Twitter classes of the *Social Public* dataset. Flickr has many large (> 600 patches) images, while Twitter has only very small images.

of occurrence of the generated patches. For example, the frequency of occurrence F_k of a patch from an image in the training set is defined by its number N of patches and its associated weight $|P_k|$ (see Eq. 25).

$$F_k = N \cdot \frac{1}{|P_k|} = \frac{N}{|P_k|} \tag{25}$$

The goal is to balance the dataset based on the number of patches per image per class with a new class weight factor f_k (also mentioned in subsection 4.3.4). To define f_k, we combine this occurrence frequency F_k with the proposed weight w_{Nk} (see Eq. 26). Therefore, the overall weight for all patches in a given image during the training phase depends only on the number of images per class $|I_k|$, which is perfectly balanced in our datasets. Thus, we obtain the same exact weight for each image during the learning phase, regardless of its class and number of patches.

$$f_k = w_{Nk} * F_k = \frac{|P_k|}{|I_k|} \cdot \frac{1}{N} \cdot \frac{N}{|P_k|} = \frac{1}{|I_k|} \tag{26}$$

We tested the proposed weighting technique by training the TSN *without* and *with* the sample weights. The image-level performance increased from an average 89% to 95% *recall* when tested on *Social Public* and from 95% to 98% *recall* on *IPLab*. The image-level comparison between our TSN and SOTA method (Caldelli et al. 2018) is reported in the Tab. 29. As for the pixel-level performance, our proposed method outperformed the PRNU-based method from the literature.

UCID Social	**Facebook**	**Flickr**	**Twitter**
(Caldelli et al. 2018)	87.35%	97.42%	87.73%
Two-stream (Our)	**100%**	**99.80%**	**98.62%**

Table 29: Comparison of image-level classification on *UCID Social* between TSN and (Caldelli et al. 2018).

The performance comparison at the image-level between our TSN and the other SOTA method (Amerini, Uricchio, and Caldelli 2017) are reported in the Tab. 30. Our proposed method outperforms for all tests except on *Social Public* for Twitter class (100% against 98.74%).

Conclusion

We propose a TSN that achieves 98% correct classification of three SNs (Facebook, Flickr and Twitter) on average over three test datasets. Our architecture is based on two domains: PRNU and DCT coefficients. We have improved these two features, with a new method to encode the DCT coefficients and the use of Noiseprint. We propose solutions for problems related to unbalanced datasets, from patch-level network training to image-level unbiased classification. Finally, the results produced by our TSN outperform those of SOTA at both the patch- and image-levels.

UCID Social	**TSN**	**SOTA**	**TSN**	**SOTA**	**TSN**	**SOTA**
	Facebook		*Flickr*		*Twitter*	
Facebook	**98.20%**	97.37%	0.20%	0.00%	1.40%	2.63%
Flickr	0.00%	0.00%	**100%**	**100%**	0.00%	0.00%
Twitter	0.00%	0.00%	0.00%	0.00%	**100%**	**100%**

Social Public	**TSN**	**SOTA**	**TSN**	**SOTA**	**TSN**	**SOTA**
	Facebook		*Flickr*		*Twitter*	
Facebook	**91.21%**	88.24%	0.00%	0.00%	8.79%	11.76%
Flickr	0.00%	0.99%	**98.15%**	97.03%	1.85%	1.98%
Twitter	0.10%	0.00%	1.23%	0.00%	98.67%	**100%**

IPLab	**TSN**	**SOTA**	**TSN**	**SOTA**	**TSN**	**SOTA**
	Facebook		*Flickr*		*Twitter*	
Facebook	**97.86%**	96.01%	2.14%	3.99%	0.00%	0.00%
Flickr	2.45%	1.68%	**97.55%**	97.06%	3.28%	1.26%
Twitter	0.00%	0.00%	0.00%	1.26%	**100%**	98.74%

Table 30: Confusion matrices of image-level classification between TSN and (Amerini, Uricchio, and Caldelli 2017), on three databases: *UCID Social*, *Social Public* and *IPLab*.

Recognizing Social Network (SN) from an image is a relatively new area of research in the field of DIF. Classification of SN may be a crucial element for the growing number of social media crime cases, such as cyberbullying. This study considers SOTA approaches addressing this problem and proposes a new methodology to improve the results obtained to date. Our identification technique is based on the idea that SNs perform some processing on downloaded images. These operations, such as resizing or compression, leave some artifacts on the images. We propose to use the DCT features and the residual noise analysis of the images to detect these artifacts. A Two-Stream Network (TSN), which combines inputs from both artifact domains, is trained to classify the SN of the images. We performed the classification on three different datasets: *UCID Social*, *Social Public*, and *IPLab*. This section explores two domains, proposes strategies for handling unbalanced datasets, provides details on the SN. Finally, this section presents the results obtained which outperform the current SOTA methods.

4.4 Impact on Social Network Recognition

Recognition of Social Network (SN) is a topic of Digital Image Forensics (DIF), which has been addressed by few methods based on Deep Learning (DL). The previous section 4.3 introduces these State-Of-The-Art (SOTA) approaches dedicated to SN identification. The Two-Stream Network (TSN) that we propose have outperformed them on three databases specialized for SN (with three classes: Facebook, Twitter and Flickr). The recognition is notably performed by analyzing the Discrete Cosine Transform (DCT) coefficients, which are modified by the processing algorithms of SNs. Thus, AI-based compression could have an impact on classification performances, as it is the case for forgery detection (see section 4.2).

This section study the impact of manipulations as attack on the recognition of SN. The subsection 4.4.1 present the different attacks. The context of SN recognition is detailed in the subsection 4.4.2. The proposed framework of evaluation and the results are reported in the subsections 4.4.3 and 4.4.4.

4.4.1 Robustness to Counter-Forensics

Attacks in DIF

Most of the SOTA methods have based their classification on specific artifacts, such as Photo-Response Non-Uniformity (PRNU) or DCT features. In fact, the assumption behind these approaches is that SNs apply different processing manipulations when users download images. Most of these manipulations are JPEG compression and resizing, which leave traces on the downloaded images. Each SN has its own processing algorithm, resulting in unique traces from one SN to another. This phenomenon is essential for identifying the source SN, but can also be a point of attack to fool these recognition methods. In other areas of DIF, such as source camera recognition, methods are often evaluated based on their robustness to manipulations (i.e., post-processing operations). Most of these manipulations are used as attacks to hide artifacts essential for identification. In the case of SN recognition, JPEG compression could be one of the main attacks as it is expected to impact DCT-based features.

JPEG-AI

Recently, the JPEG organization mentioned the upcoming new compression standard that will be based on DL architecture: JPEG-AI[10]. This announcement follows the emergence of recent AI-based compression methods, mainly composed of auto-encoders. First, the JPEG organization evaluated these new compression

[10] https://jpeg.org/jpegai/index.html

solutions for standardization. Then, they presented more explanations about their future standard, which will aim to provide better compression for humans and machines. The proposed architecture consists of an encoder, a bottleneck and three outputs: the (usual) decoder and two others for image processing and computer vision tasks. Finally, the proposals will be presented and discussed at the 96^{th} JPEG meeting (July 2022), and the JPEG-AI standard should be available in the next few years (estimated in April 2024).

Robustness Issue

Robustness evaluation is important in DIF tasks, including SN recognition, because classification is always based on artifacts. The application of manipulations could mask or even remove traces that are unique to each individual SN. Classification methods could be fooled and their performance affected. Despite its relevance, there is no work to date addressing the topic of post-processing operations as an attack on SN recognition. The objective of this study is to evaluate the impact of such manipulations on SN recognition. This work is particularly focused on AI-based compression, which will be the next standard of the JPEG organization and will probably be democratized in the next few years to essential areas of image processing. To evaluate its impact, we study the SOTA methods, in order to select the best one to date (see subsection 4.4.2).

4.4.2 Context of Social Network Recognition

For a long time, DIF methods have used mathematical statistics to extract the appropriate features for their tasks. With the emergence of the DL, their architecture has moved towards the Convolutional Neural Networks (CNNs), rather known for their efficiency in image analysis. In parallel to the CNNs, pre-processing modules have often been added upstream of the models, in order to extract the essential artifacts. Similarly, recognition methods for SNs are mainly based on DL. Despite their recent novelty, they already reach high performances. These results mainly come from the analysis of artifacts related to SN processing algorithms, such as the PRNU and DCT features. In what follows, we summarize the three methods for recognizing SN that focus specifically on these artifacts (i.e., PRNU and DCT features). These methods are detailed in the previous section (see subsection 4.3.1).

Literature

(Amerini, Uricchio, and Caldelli 2017) based their method on the analysis of DCT-based features. The pipeline is quite similar to the one proposed by (Caldelli et al. 2017), with an CNN of 2 convolutional layers followed by 3 Fully-Connected

(FC) layers. The main difference comes from the pre-processing module, which is dedicated to DCT-based feature patches. First, patches of size 64×64 are cropped and the DCT coefficients are extracted from 8×8 blocks. Then, histograms consisting of 101 bins (range -50 to $+50$ for the coefficients) are computed for the first 9 spatial frequencies of the blocks. Finally, a vector of size 909×1 (101 bins and 9 spatial frequencies) is computed from these histograms and fed to the network.

(Caldelli et al. 2018) propose to use PRNU to recognize SNs. Classification is performed by a CNN of 4 convolution layers, followed by 2 Fully-Connected (FC) layers. First, the residual noise is obtained using a PRNU extraction module, then small patches of size 64×64 are cropped and introduced into the network. Then, the PRNU fingerprint $hatK$ is computed from a group of M images by applying a minimum variance estimator from their residual noises W_i. Finally, the residual noise is obtained from the original I_i images and their noise-free versions I_i^{den} (see equations 27 and 28).

$$W_i = I_i - I_i^{den} \qquad (27) \qquad \text{and} \qquad \hat{K} = \frac{\sum_{i=1}^{M} W_i I_i}{\sum_{i=1}^{M} (I_i)^2} \qquad (28)$$

Two-stream Network

The section 4.3 propose a method based on both DCT and PRNU features. The objective of this work is to improve performance by combining two essential feature domains: the DCT and PRNU features. Thus, the architecture is a TSN. The subnetwork based on DCT is fed by a vector of size 1980, which is not made with binary-limited histograms, and thus contains all their information. The PRNU-based subnetwork is Noiseprint (Cozzolino and Verdoliva 2020), a noise extractor based on CNN. Noiseprint is particularly well known and has already proven its efficiency in DIF tasks. Compared to the common method of extracting PRNU, Noiseprint is pre-trained, so it can provide the residual noise of any image. The TSN obtained the best results in most cases compared to the DCT-based method (Amerini, Uricchio, and Caldelli 2017) (see Tab. 30, without (Caldelli et al. 2018) because it was not evaluated on all databases, and did not perform as well as the others). Based on this analysis of the literature on SN recognition, we decided to use TSN, as it obtained the best performance on several databases.

4.4.3 AI-based Compression and SN Recognition

Proposed Framework

The objective of our study is to provide a first evaluation of the impact of AI-based image compression on the source SN recognition task. Thus, we set up a framework that brings together the best method for identifying SN to date, an AI-based compression model, and three databases with at least three classes in common. In addition, all elements of our framework are publicly available and reproducible.

SN Detector

Based on the results of the State-Of-The-Art (SOTA) methods, which are illustrated in 4.4.2 (see subsection 4.3.1 for details), TSN[11] is the best method to date. This network improve two methods of the literature, respectively based on DCT and PRNU features, and merge them into one TSN. They proved that each of the subnetworks performed better than the SOTA CNN (see subsection 4.3.5). Moreover, the TSN method outperforms the literature methods, both at the pixel- and image-level (see subsection 4.3.6). This method is based on two domains (PRNU and DCT) which could be particularly affected by the application of manipulations. We therefore decided to analyze the impact of AI-based compression on this TSN.

AI-based Compression

When it comes to AI-based compression, the literature is vast, with many open source solutions. However, the JPEG organization has reviewed early learning-based compression solutions to summarize the progress made. We decided to select the High-Fidelity Compression (HiFiC)[12] (Mentzer et al. 2020), as it was the first to use Generative Adversarial Networks (GAN). In addition, it provides high perceptual fidelity for the reconstructed images. They explain that the HiFiC was trained using several quality measures (Peak Signal to Noise Ratio (PSNR), Multi-Scale Structural Similarity (MS-SSIM), etc.) while maintaining a low bit rate (half that of SOTA methods). Moreover, this AI-based method can compress images with three different quality levels: *low*, *medium* and *high*.

To perform a thorough analysis, we compare this new solution to the existing conventional solution: the JPEG compression. We also included three other manipulations in our study: Additive White Gaussian Noise (AWGN), Gaussian Blurring (GB) and Median Filter (MF). We chose these post-processing operations because they have been particularly used in the DIF literature (Bayar and Stamm 2018a). As there are different image qualities for HiFiC, we selected two opposite levels: 1) the one that can be considered as a strong attack (i.e. *low*), which means with a decrease in image quality; 2) and the second one with a limited visual degradation (i.e. *high*). This assumption is based on the image quality of

[11] https://github.com/francescotescari/social

[12] https://github.com/Justin-Tan/high-fidelity-generative-compression

both compressions, obtained with the Structural Similarity (SSIM). In order to have a fair comparison, we also selected two settings per manipulation, of equal image quality in terms of SSIM.

Databases

For our evaluation, we chose the same databases as for the TSN: *UCID Social*, *Social Public*, and *IPLab*; and in particular their three common classes: Facebook, Flickr, and Twitter. The *Social UCID* is obtained from the Uncompressed Color Image Database (UCID) (Schaefer and Stich 2004), which contains $1,338$ images that were first compressed with 10 different JPEG QFs (50 to 95 with a step of 5), and then uploaded to the desired SNs. This way of creating a database leads to a dataset in a *controlled environment*, where images are first selected, then uploaded to a SN, and finally downloaded. *IPLab* is also a controlled environment dataset that was generated from 240 images, downloaded to 8 different SNs. In contrast, *Social Public* is considered an *uncontrolled environment* dataset because it is composed of images directly from the SNs. It contains $3,000$ images downloaded from three different SNs.

4.4.4 Experimental Results

Based on these elements from our proposed framework, we have applied AI-based and JPEG compressions, as well as the other manipulations, to images from the selected databases. As we explained earlier, based on the resulting images quality obtained with HiFiC-*high* and -*low*, we have selected two parameters for each manipulation and created two groups that match the image qualities obtained (see tab. 31).

Manipulations	**HiFiC**	**JPEG**	**AWGN**	**GB**	**MF**
First group	*High*	$QF = 25$	$\sigma = 2.5$	$size = 3$	$size = 3$
SSIM	0.88	0.89	0.88	0.9	0.87
Second group	*Low*	$QF = 10$	$\sigma = 6$	$size = 7$	$size = 5$
SSIM	0.78	0.79	0.77	0.77	0.75

Table 31: Details of both level of quality for each manipulation, according to the SSIM.

These groups are made according to an objective metrics (SSIM), but some manipulations as HiFiC have less visual impact than other (see Fig. 19). The first group corresponds to attacks with limited visual degradation and an average SSIM of 0.88, while the second group corresponds to strong attacks resulting in image degradation with an average SSIM of 0.77.

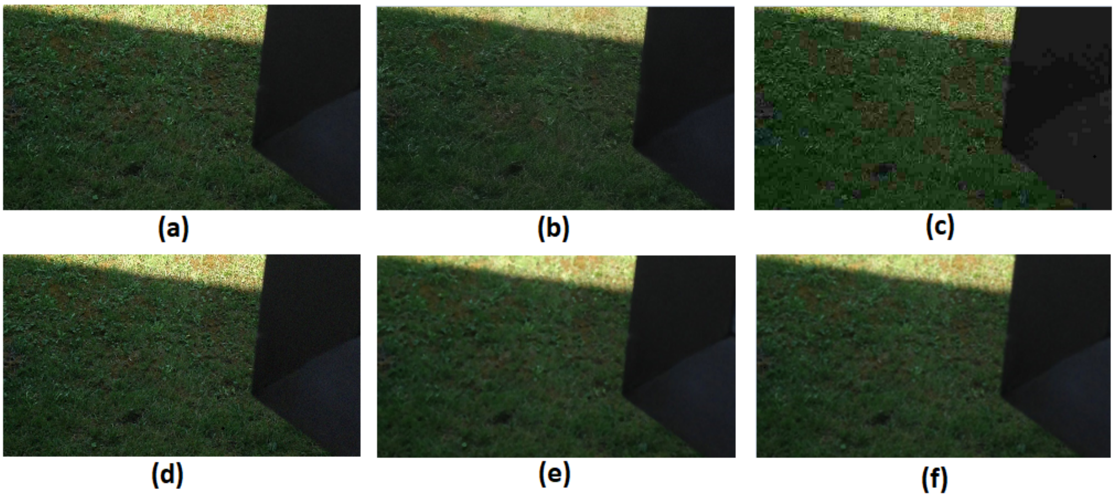


Figure 19: Comparison of visual image quality of each manipulation (same objective quality) with a region of an image from IPLAB - Flickr. (a) original (b) HiFiC-*low* (c) JPEG10 (d) AWGN6 (e) MF5 (f) GB7.

We conducted our evaluation on the three selected databases (*IPLAB*, *UCID Social*, and *Social Public*) and for each manipulation to obtain the performances of SN recognition, in the context of post-processing manipulations. The results are reported in Tab. 32, according to the database, the SN and the manipulation used. In order to compare all post-processing operations and their impact on recognition performances, we also added the loss L, which is the average between the recognition *accuracies* on the original images Acc_o and the manipulated images Acc_m (see Eq. 29) of the SNs.

$$L = \frac{\sum_{i=1}^{K} Acc_o(i) - Acc_m(i)}{K} \tag{29}$$

Moreover, as we selected various databases, in terms of composition of environment (i.e. *controlled* and *uncontrolled*), we have computed the mean drop, which corresponds to the average of each loss along the three databases.

In a strong attack context, which means a degradation of the image quality (i.e. the second group) and an average SSIM of 0.77, the chosen manipulation does not matter. Indeed, with the exception of AWGN ($\sigma = 6$), all post-processing operations have a similar impact on the recognition performance of SN, with an average loss of *accuracy* percentage of 24 points. However, for the first group, the performance drop is not the same from one manipulation to another. In an attack context with limited visual degradation, HiFiC-*high* results in the largest drop in performance. Indeed, the average loss of the percentage of *accuracy* is 20 points for HiFiC-*high* against 10 for the other manipulations (more or less two times more).

UCID Social	**Original**	**HiFiC**		**JPEG**		**AWGN**		**GB**		**MF**	
		High	Low	25	10	2.5	6	3	7	3	5
Facebook	98.20%	76%	73.6%	89.9%	71.3%	86.9%	82.31%	97.3%	70.6%	92.6%	76.4%
Flickr	100%	71.3%	60.1%	89.5%	69.4%	95.9%	71.3%	81.3%	72%	84.1%	72.7%
Twitter	100%	76%	73.3%	90.6%	72%	94.3%	83%	86%	69.5%	89.4%	78.6%
Loss (of %)		24.97	30.4	9.4	28.5	7.03	20.53	11.2	28.7	10.7	23.5
Social Public	**Original**	**HiFiC**		**JPEG**		**AWGN**		**GB**		**MF**	
		High	Low	25	10	2.5	6	3	7	3	5
Facebook	91.21%	80.6%	77.5%	82.8%	74.4%	93.1%	85.6%	83.5%	76.1%	85.4%	75.8%
Flickr	98.15%	70.6%	74%	80.9%	75.4%	93.3%	83.1%	80.8%	86%	91.5%	71.8%
Twitter	98.67%	66.9%	54.6%	78.6%	67.9%	91.2%	84.6%	78.7%	68.7%	81.5%	68.2%
Loss (of %)		23.31	27.31	14.91	23.44	3.48	11.58	15.01	19.08	9.88	24.08
IPLab	**Original**	**HiFiC**		**JPEG**		**AWGN**		**GB**		**MF**	
		High	Low	25	10	2.5	6	3	7	3	5
Facebook	97.86%	70.6%	62%	86.2%	82.1%	98.9%	74.5%	87.6%	75.9%	78.5%	69.4%
Flickr	97.55%	91.4%	85.6%	86.4%	86.2%	73%	68.1%	92.1%	82%	87.2%	66%
Twitter	100%	95%	89.9%	91.8%	72.8%	88%	72.7%	76.7%	55.5%	74.1%	71.8%
Loss (of %)		12.80	19.30	10.34	18.1	11.84	26.70	13	27.34	18.54	29.40
Mean loss (of %)		**20.36**	**25.67**	11.55	23.35	7.45	19.6	13.07	25.04	13.04	25.66

Table 32: Results of TSN for image-wise classification, performed on *UCID Social*, *Social Public* and *IPLab* with different manipulations.

The results obtained by our analysis show that to attack the recognition of SN without degrading the quality of an image too much, AI-based compression is the best choice.

Conclusion

For the first time, AI-based compression and SN recognition are studied together to evaluate possible influences. The impact of AI-based compression on the performance of SN identification is compared to traditional JPEG compression, as well as to three other manipulations. We created two groups according to the quality of the image obtained with SSIM: the first with limited visual degradation, and the second with strong quality downgrades. In the context of high quality degradation, all operations lead to a similar decrease in performance, while HiFiC-*high* achieves the largest decrease in performance for attacks with limited degradation. Therefore, AI-based compression should be included in the group of post-processing operations to evaluate the robustness of DIF methods, and in particular for SN recognition.

The few methods of literature on SN recognition has already achieved high performance. Unlike other DIF topics, there is no work that deals with counter analysis for SN recognition. Thus, we analyzed the impact of image manipulations on the performance of the TSN. In particular, we focused our study on the AI-based compression, which tends to become the new compression solution with the future standard JPEG-AI (from the JPEG organization). To perform a fair analysis, we compared AI-based compression with conventional JPEG compression, and also included three other manipulations: MF, GB, and AWGN that are often used to evaluate the robustness of DIF methods. From these manipulations, we created two groups based on the quality of the image obtained by the SSIM, which correspond respectively to attacks with high (average SSIM of 0.77) and low (average SSIM of 0.88) image degradation. In the context of high image quality degradation, all manipulations lead to a similar drop in performance, while for quality-preserving attacks, AI-based compression is able to achieve a drop twice as large as the other manipulations. Overall, this study is the first work to study the interference between both AI-based methods that are SN recognition and image compression.

5 Conclusion and Perspectives

5.1 Protocols for Issues of Camera Recognition

The study revealed some problems with camera recognition. All of these problems are associated with camera fingerprints. In fact, they are unique to each camera, which requires a robustness assessment. The open-set scenario is dedicated to this phenomenon, but the literature has shown a lack of databases for the assessment. Moreover, the most difficult task, camera device recognition, has been understudied. Therefore, one chapter of our thesis (see chapter 3) focuses on protocols to tackle these issues.

Robustness Study

The open-set scenario is devoted to this aspect, with *unknown* cameras. However, there are only a few articles that deal with this classification. Thus, the first protocol we proposed is a robustness study for DL architectures, based on transfer learning and fine-tuning approaches. The protocol is performed with *Dresden* and is composed of three experiments: i) one for the basic classification; ii) two for the open-set scenario. The results obtained show that DenseNet201 and partial fine-tuning are to be preferred for large databases. The best contribution of this protocol is to confirm the problem of camera fingerprints for two aspects: 1) it is difficult to classify *unknown* cameras; ii) the more cameras there are, the more difficult the classification is. This robustness study is not sufficient, especially when compared with other areas of image processing. In fact, the use of a single database does not correspond to a complete evaluation of methods based on DL.

Multi-databases Protocol

This phenomenon is even more crucial for camera recognition due to the uniqueness of camera fingerprints. However, most of SOTA methods are evaluated only on *Dresden* (and sometimes on a private dataset, not publicly available). Therefore, we proposed a multi-database protocol to solve this problem for camera model identification, which is the most discussed application. We selected three databases dedicated to camera recognition, with various and complementary parameters. *SOCRatES* (62 models for 101 cameras) and *Forchheim* (25 models for 27 cameras) are specialized in smartphones, while *Dresden* (27 models for 73 cameras) consists of sensor cameras. In comparison to the Tab. 11, the comparison is possible between the SOTA methods (see Tab. 19). The method dedicated to triple classification (Ding et al. 2019) performed better than the others on most experiments (6 out of 9). Moreover, the results are quite similar to those reported in their paper and do not decrease too much depending on the databases. This robustness could come from its application (i.e., triple classification), where the most *difficult* label is addressed (i.e., the device).

More reliable Protocol

Another problem is highlighted by the comparison of DL-based methods on *Dresden* (see Tab. 11). Indeed, only four methods, including one by basic classification, deal with camera recognition. This is due to the difficulty of classifying camera devices, according to the performance of the SOTA methods. In fact, this recognition challenge is due to the similarity of camera fingerprints for devices of the same model. This aspect has been highlighted in the literature, but has never been fully addressed. We have proposed a more reliable protocol for device verification $(1 - to - 1)$ composed of three levels of difficulty: *basic*, *intermediate* and *advanced*. The main contribution of this protocol is our camera selection based on these difficulty levels. In particular, this selection avoids the uncontrolled distribution of cameras in the databases. Our results reveal the problem of SOTA methods to perform in a difficult context (i.e. cameras of the same model). In particular, they highlight that none of the best SOTA methods can be better than random classification on *Dresden*, for the *advanced* level. Thus, our protocol composed of three levels of difficulty should be used by the literature to perform a reliable evaluation of their methods.

Comparison Modules

Future work for camera recognition could arise from the last proposed protocol. Indeed, some particular loss functions could be used in the Siamese Neural Network (SNN) comparison module, which could help to solve the problem of close camera fingerprints. The principle is to compare two feature vectors to conclude whether the input images are from the same camera. We consider two classification methods for this similarity module that could be exploited. The goal of each of them is to reduce the distance for coherent pairs (same device) and to increase it for non-coherent pairs through a particular loss function.

The first module uses the contrastive loss, which is based on the ground truth t, the score s and the margin set to 1 (Eq. 30). The margin represents the value at which the pair is considered different. The contrastive loss is combined with the Euclidean distance (Eq. 31) of the subarray outputs (e.g. xl and xr).

$$L = \frac{1}{2}ts^2 + \frac{1}{2}(1-t)\max(margin - s, 0)^2 \quad (30) \text{ and } \quad s = \|xl - xr\| \quad (31)$$

According to the pair given as input, if the ground truth is equal to 1 (Eq. 32), the distance s is minimized while it is maximized if it is equal to 0 (Eq. 33). The margin is used to adjust the constraint, as it sets the allowed distance between two images to be considered as different. This parameter allows adapting the meaning of positive and negative pairs.

$$L_0 = \frac{1}{2}\max(margin - s, 0)^2 \qquad (32) \qquad \text{and} \qquad L_1 = \frac{1}{2}ts^2 \qquad (33)$$

The second similarity approach is really different, as it requires three objects. The anchor a, which represents the point of comparison, the positive p and the negative n, which are used to create respectively similar and dissimilar pairs with the anchor object. The Euclidean distance of the two pairs d_p and d_n as well as the margin are used in the calculation of the triplet loss (Eq. 34).

$$L = \max(d_p - d_n + margin, 0) \tag{34}$$

Using a positive and negative object in the loss function, the model follows two strategies: minimize similar inputs and maximize dissimilar inputs. The margin is also used to indicate the maximum distance to be reached between two dissimilar objects. This similarity approach is even more complete than the contrastive approach, as it includes a limiting parameter and moves positive objects closer and negative objects further away, respectively.

Our research on camera recognition problems is divided into three phases, each of which is devoted to a protocol. The robustness study highlighted the challenge of classifying *unknown* cameras and the increasing difficulty with the number of cameras. However, it was not sufficient, as other image processing domains typically evaluate their methods on multiple databases. Therefore, we proposed our multi-database protocol for camera model identification. The crucial element brought by this study is the possibility of a fair comparison between SOTA methods. Notably, the triple classification-based method outperformed the others (within the studied set), highlighting the potential importance of classifying device. We therefore proposed our final, more reliable protocol, consisting of three levels of increasing difficulty (*basic*, *intermediate* and *advanced*). This protocol reveals the issue of SOTA methods to recognize cameras in a difficult context, with performances similar to random classification on Dresden. A research perspective could concern comparison modules for the Siamese network with the use of contrastive or triplet loss functions. These are particularly dedicated to the discrimination of negative and positive pairs.

5.2 Impact of AI-based Compression

The literature review shows an increase in counter-forensic approaches based on Deep Learning (DL). Most of these approaches use misclassification or adversarial generation techniques. On the other hand, classical methods focus more on particular artifacts, such as camera or compression traces. Some forgery detection methods have shown the potential negative impact of manipulation on their performances. With the emergence of compression solutions based on Artificial Intelligence (AI), we decided to study the potential impact of such compression on certain tasks DIF.

Impact on Forgery Detection

The first task addressed in our thesis was forgery detection, which is one of the main topics of DIF. In particular, compression artifacts are used by the methods to detect falsifications. We therefore selected CAT-Net - the best JPEG-related detector to date, High-Fidelity Compression (HiFiC) - a compression solution based on AI and *Casia v2* - a database that contains both splicing and copy-move. In this first study, the goal was to understand whether such compression can be considered as an unintended attack. Indeed, recompression can be applied in a benign manner (i.e., without the intention of counter-analysis). Our evaluation therefore focused on JPEG and AI-based recompression. We also added a manipulation that is usually applied as an attack to degrade forensic artifacts: Additive White Gaussian Noise (AWGN). To perform a fair study, we set an image quality level with SSIM. Our assumptions about AI-based compression were correct, as HiFiC-*high* results in the largest drop in performance at the same image quality. Thus, HiFiC and more generally AI-based compression should be considered for counter-forensic as an unintended attack.

TSN for SN recognition

The second task using compression artifacts that we have discussed is Social Network (SN) recognition. However, this task is quite new, and the methods are limited. We therefore decided to propose a Two-Stream Network (TSN) to exploit the PRNU and DCT features. We tried to improve these features with Noiseprint and a new coding approach for the PRNU and DCT domains, respectively. To perform fair evaluations, we selected the same databases as in the literature: *UCID Social*, *Social Public* and *IPLab*, with three classes (Flickr, Facebook and Twitter). First, we performed a single-stream evaluation to verify our proposed improvements. Both subnetworks performed better than or similar to the respective SOTA method. Then, we performed a two-stream evaluation and the overall results exceeded the literature results at the pixel- and image-level.

Impact on SN recognition

After having proposed a method for recognizing SN, we decided to evaluate the impact of AI-based compression on its performance. Indeed, this manipulation proved to be an efficient attack on the detection of forgery, and we wanted to extend our analysis to another task of DIF. Since we have already verified the potential impact of this benign operation on compression artifacts, we expanded our set of post-processing operations to five: 1) HiFiC; 2) JPEG compression; 3) Median Filter (MF); 4) Gaussian Blurring (GB); 5) Additive White Gaussian Noise (AWGN). For this study, we performed two groups of manipulations according to their image quality: i) strong image degradation with an average SSIM of 0.77; ii) limited quality downgrade with an average SSIM of 0.88. The performances obtained during our evaluation confirm the first hypothesis made on forgery detection. In the context of image quality preserving attacks, HiFiC-*high* achieves a drop twice as high as other manipulations. For attacks with high degradation, the manipulation does not matter, as the performance drop is quite similar.

AI-based compression Database

Our evaluations of the impact of post-processing manipulations on DIF tasks have shown the importance of AI-based compression as an unintended attack. With the development of DL-based methods in image processing domains, other compression solutions like HiFiC could be proposed. In addition, with the future JPEG-AI compression standard, it will be more and more common to see images compressed based on AI.

Our chapter on AI-based compression highlighted the importance of double compression for forensic tasks. Indeed, compression artifacts are created during such applications. Some articles in the literature are devoted to classify images as single or double compressed. However, these methods only take into account JPEG compression. It could be interesting to integrate AI-based compression into this process, or even the future standard JPEG-AI. In fact, as the number of AI-based compressed images will increase, mixed compressed images could appear. This phenomenon may become more pronounced with the standardization of JPEG-AI. For example, SNs will probably adopt this compression as a new standard for their processing algorithm.

One prospect for future work is the creation of a database that mixes both JPEG and AI-based compressions. The goal will be to create four different types of double compression: 1) common double JPEG; 2) new double AI-based; 3) first mix JPEG/AI-based; 4) second mix AI-based/JPEG. In this case, the way to apply the detection of Double JPEG-Compression (DJPEG-C) will change. The objective will be to identify these four different types of compression.

5.3 Video Forensics

Another development path for DIF with DL concerns videos. Indeed, some methods have already tackled this subject with the detection and localization of fakes like the copy-move (D'Amiano et al. 2019; D'Amiano et al. 2015) based on the patch-match algorithm or the splicing (Mandelli et al. 2018). (Bakas et al. 2018) tackle Moving Picture Experts Group (MPEG) double compression and (Gan et al. 2019) expose a general method of forgery detection.

Camera recognition is also tackled with different approaches: (B. Hosler et al. 2019) present a classical recognition, while (Mayer et al. 2020) address an open video scenario (with *unknown* cameras). (Mullan et al. 2017) propose a residual-based comparison method to detect forensic features for forgery detection and also for camera recognition.

There are two main video databases for forensic analysis of digital video: *Vision* (Shullani et al. 2017) (presented in 2.3.1) and the *Video Authentication and Camera Identification Database (Video-ACID)* (B. C. Hosler et al. 2019). *Vision* contains $34,427$ images and $1,914$ videos from 35 portable devices. *Video-ACID* consists of over $12,000$ videos from 46 physical devices. Although video forensics is a topic that has already been addressed by many methods based on DL for various purposes, there are still several tasks to be achieved.

Our hypothesis for the research on AI-based compression is the possibility of a new unintended attack against compression-related DIF tasks. We therefore conducted an initial study on forgery detection to test our assumption. We compared two benign operations: AI-based and JPEG compressions. The results obtained confirm the status of the AI-based compression as an unintentional attack against forgery detection. Thus, we wanted to extend our hypothesis to another task of DIF based on the analysis of compression artifacts: recognition of SN. First, we proposed a Two-Stream Network (TSN), based on PRNU and DCT features. Our method outperformed the SOTA methods on three databases dedicated to SN recognition. Next, we selected five post-processing operations, including AI-based compression, to evaluate their impact in two cases: 1) with strong image degradation; 2) with limited quality downgrade. In the first case, the manipulation does not matter, while in the second, AI-based compression leads to a decrease twice as important as the other operations. A future work could concern the study of double compression in mixed cases, as AI-based compression will become more and more common. In fact, one prospect could be the creation of a database for double compression detection, with four cases: 1) double JPEG; 2) double AI-based; 3) JPEG/AI-based; 4) AI-based/JPEG.

5.4 Publications

This thesis gathers the work and the research of the past three years in the field of Digital Image Forensics (DIF). We wrote an amount of seven publications, whose five are already published, one is accepted and one is submitted. The first article to be published is (Berthet and J.-L. Dugelay 2020) for Visual Communications and Image Processing (VCIP), which detail and explain the different preprocessing modules of the literature. The series of publications on camera recognition has begun with the publication of the robustness study (Berthet and J. Dugelay 2021) in the French colloquium COmpression et Representation des Signaux Audiovisuels (CORESA). Then, the multiple databases (Berthet and J. Dugelay 2022) and the more reliable protocols (Berthet et al. 2022) have been published consecutively. The first one in Media, Watermarking and Security Forensics (MWSF) from the Electronic Imaging (EI) symposium and the second in International Conference on Pattern Recognition Application & Methods (ICPRAM). In parallel to these works dedicated to camera recognition, we have published a first article on the TSN for SN recognition (Berthet et al. 2021) in CyberWorlds (CW). The article on the impact of AI-based compression on forgery detection has been accepted in International Conference on Image Processing (ICIP) (2022). The one for the impact on SN recognition should be submitted soon.

References

Agarwal, R., D. Khudaniya, A. Gupta, and K. Grover. 2020. "Image Forgery Detection and Deep Learning Techniques: A Review," 1096–1100. 2020 4th International Conference on Intelligent Computing / Control Systems (ICICCS). https://doi.org/10.1109/ICICCS48265.2020.9121083.

Al Banna, M. H., M. Ali Haider, M. J. Al Nahian, M. M. Islam, K. A. Taher, and M. S. Kaiser. 2019. "Camera Model Identification using Deep CNN and Transfer Learning Approach," 626–630. 2019 International Conference on Robotics, Electrical / Signal Processing Techniques (ICREST). https://doi.org/10.1109/ICREST.2019.8644194.

Ali Qureshi, M., and M. Deriche. 2014. "A review on copy move image forgery detection techniques," 1–5. 2014 IEEE 11th International Multi-Conference on Systems, Signals Devices (SSD14), February. https://doi.org/10.1109/SSD.2014.6808907.

———. 2015. "A bibliography of pixel-based blind image forgery detection techniques." *Signal Processing: Image Communication* 39:46–74. ISSN: 0923-5965. https://doi.org/https://doi.org/10.1016/j.image.2015.08.008. http://www.sciencedirect.com/science/article/pii/S0923596515001393.

Amerini, I., L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra. 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. https://doi.org/10.1109/TIFS.2011.2129512.

Amerini, I., L. Ballan, R. Caldelli, A. Del Bimbo, and L. Del Tongo G. Serra. 2013. "Copy-move forgery detection and localization by means of robust clustering with J-Linkage." *Signal Processing: Image Communication* 28 (6): 659–669. ISSN: 0923-5965. https://doi.org/https://doi.org/10.1016/j.image.2013.03.006. https://www.sciencedirect.com/science/article/pii/S0923596513000453.

Amerini, I., T. Uricchio, L. Ballan, and R. Caldelli. 2017. "Localization of JPEG Double Compression Through Multi-domain Convolutional Neural Networks." *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* 1865–1871.

Amerini, I., T. Uricchio, and R. Caldelli. 2017. "Tracing images back to their social network of origin: A CNN-based approach," 1–6. 2017 IEEE Workshop on Information Forensics / Security (WIFS). https://doi.org/10.1109/WIFS.2017.8267660.

Ansari, M. D., S. P. Ghrera, and V. Tyagi. 2014. “Pixel-Based Image Forgery Detection: A Review.” *IETE Journal of Education* 55 (1): 40–46. https://doi.org/10.1080/09747338.2014.921415. https://doi.org/10.1080/09747338.2014.921415.

Asati, S., and P.R. Pardhi. 2014. “Exposing Digital Image Forgeries by Illumination Color Classification.” *International Journal of Engineering Trends and Technology* 18 (December): 269–271. https://doi.org/10.14445/22315381/IJETT-V18P255.

Ascenso, J., P. Akyazi, F. Pereira, and T. Ebrahimi. 2020. “Learning-based image coding: early solutions reviewing and subjective quality evaluation,” edited by Peter Schelkens and Tomasz Kozacki, 11353:164–176. International Society for Optics and Photonics, SPIE. https://doi.org/10.1117/12.2555368. https://doi.org/10.1117/12.2555368.

Bakas, J., A. K. Bashaboina, and R. Naskar. 2018. “MPEG Double Compression Based Intra-Frame Video Forgery Detection using CNN,” 221–226. 2018 International Conference on Information Technology (ICIT). https://doi.org/10.1109/ICIT.2018.00053.

Balamurugan, B., S. Maghilnan, and M. R. Kumar. 2017. “Source camera identification using SPN with PRNU estimation and enhancement,” 1–6. 2017 International Conference on Intelligent Computing / Control (I2C2). https://doi.org/10.1109/I2C2.2017.8321801.

Ballé, J., D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. 2018. “Variational image compression with a scale hyperprior.” *arXiv preprint arXiv:1802.01436.*

Bappy, J. H., C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury. 2019. “Hybrid LSTM and Encoder–Decoder Architecture for Detection of Image Forgeries.” *IEEE Transactions on Image Processing* 28, no. 7 (July): 3286–3300. https://doi.org/10.1109/tip.2019.2895466. https://doi.org/10.1109%5C%2Ftip.2019.2895466.

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

Barni, M., Z. Chen, and B. Tondi. 2016. “Adversary-aware, data-driven detection of double JPEG compression: How to make counter-forensics harder,” 1–6. 2016 IEEE International Workshop on Information Forensics / Security (WIFS). https://doi.org/10.1109/WIFS.2016.7823902.

Barni, M., A. Costanzo, and L. Sabatini. 2010. “Identification of cut amp; paste tampering by means of double-JPEG detection and image segmentation,” 1687–1690. Proceedings of 2010 IEEE International Symposium on Circuits / Systems. https://doi.org/10.1109/ISCAS.2010.5537505.

Barni, M., K. Kallas, E. Nowroozi, and B. Tondi. 2018. *On the Transferability of Adversarial Examples Against CNN-Based Image Forensics.* arXiv: 1811.01629 [cs.CR].

Barni, M., E. Nowroozi, and B. Tondi. 2017. “Higher-order, adversary-aware, double JPEG-detection via selected training on attacked samples,” 281–285. 2017 25th European Signal Processing Conference (EUSIPCO). https://doi.org/10.23919/EUSIPCO.2017.8081213.

Barni, M., M. C. Stamm, and B. Tondi. 2018. “Adversarial Multimedia Forensics: Overview and Challenges Ahead,” 962–966. 2018 26th European Signal Processing Conference (EUSIPCO). https://doi.org/10.23919/EUSIPCO.2018.8553305.

Bayar, B., and M. C. Stamm. 2016. “A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer,” 5–10. IH&#38;MMSec ’16. Vigo, Galicia, Spain: ACM. ISBN: 978-1-4503-4290-2. https://doi.org/10.1145/2909827.2930786. http://doi.acm.org/10.1145/2909827.2930786.

———. 2017a. “Augmented convolutional feature maps for robust CNN-based camera model identification,” 4098–4102. https://doi.org/10.1109/ICIP.2017.8297053.

———. 2017b. “Design Principles of Convolutional Neural Networks for Multimedia Forensics.” Media Watermarking, Security, / Forensics.

———. 2017c. “On the robustness of constrained convolutional neural networks to JPEG post-compression for image resampling detection,” 2152–2156. IEEE International Conference on Acoustics, Speech / Signal Processing (ICASSP), March. https://doi.org/10.1109/ICASSP.2017.7952537.

———. 2018a. “Constrained Convolutional Neural Networks: A New Approach Towards General Purpose Image Manipulation Detection.” *IEEE Transactions on Information Forensics and Security* 13, no. 11 (November): 2691–2706. https://doi.org/10.1109/TIFS.2018.2825953.

Bayar, B., and M. C. Stamm. 2018b. "Towards Open Set Camera Model Identification Using a Deep Learning Framework," 2007–2011. 2018 IEEE International Conference on Acoustics, Speech / Signal Processing (ICASSP). https://doi.org/10.1109/ICASSP.2018.8462383.

Bayram, S., H. Sencar, N. Memon, and I. Avcibas. 2005. "Source camera identification based on CFA interpolation," 3:III–69. IEEE International Conference on Image Processing 2005, September. https://doi.org/10.1109/ICIP.2005.1530330.

Bayram, S., H. T. Sencar, and N. Memon. 2009. "An efficient and robust method for detecting copy-move forgery," 1053–1056. 2009 IEEE International Conference on Acoustics, Speech / Signal Processing, April. https://doi.org/10.1109/ICASSP.2009.4959768.

Berthet, A., and J-L Dugelay. 2020. "A review of data preprocessing modules in digital image forensics methods using deep learning," 281–284. 2020 IEEE International Conference on Visual Communications / Image Processing (VCIP). https://doi.org/10.1109/VCIP49819.2020.9301880.

Berthet, A., and J.L. Dugelay. 2021. "Étude comparative de l'apprentissage par transfert pour l'identification des caméras." In *CORESA 2021, 21eme Colloque sur la COmpression et REprésentation des Signaux Audiovisuels,* edited by UCA - Université Côte d'Azur. Sophia Antipolis, November.

———. 2022. "Comparative study of DL-based methods performance for camera model identification with multiple databases." In *Electronic Imaging - MWSF 2022, Media Watermarking, Security, and Forensics Conference,* edited by IS&amp;T. January.

Berthet, A., C. Galdi, and J.L. Dugelay. 2022. "Towards a more reliable and reproducible protocol of source camera recognition." In *ICPRAM 2022, 11th Intertational Conference on Pattern Recognition Applications and Methods,* edited by Insticc. February.

Berthet, A., F. Tescari, C. Galdi, and J.-L. Dugelay. 2021. "Two-stream Convolutional Neural Network for Image Source Social Network Identification," 229–237. 2021 International Conference on Cyberworlds (CW). https://doi.org/10.1109/CW52790.2021.00047.

Bianchi, T., and A. Piva. 2012a. "Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps." *IEEE Transactions on Information Forensics and Security* 7 (2): 842–848. https://doi.org/10.1109/TIFS.2011.2170836.

Bianchi, T., and A. Piva. 2012b. "Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts." *IEEE Transactions on Information Forensics and Security* 7 (3): 1003–1017. https://doi.org/10.1109/TIFS.2012.2187516.

Bondi, L., L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro. 2017. "First Steps Toward Camera Model Identification With Convolutional Neural Networks." *IEEE Signal Processing Letters* 24, no. 3 (March): 259–263. https://doi.org/10.1109/LSP.2016.2641006.

Bondi, L., S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro. 2017. "Tampering Detection and Localization Through Clustering of Camera-Based CNN Features," 1855–1864. 2017 IEEE Conference on Computer Vision / Pattern Recognition Workshops (CVPRW). https://doi.org/10.1109/CVPRW.2017.232.

Bunk, J., J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. S. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson. 2017. "Detection and Localization of Image Forgeries Using Resampling Features and Deep Learning," 1881–1889. 2017 IEEE Conference on Computer Vision / Pattern Recognition Workshops (CVPRW), July. https://doi.org/10.1109/CVPRW.2017.235.

Caldelli, R., I. Amerini, and C. T. Li. 2018. "PRNU-based Image Classification of Origin Social Network with CNN," 1357–1361. 2018 26th European Signal Processing Conference (EUSIPCO). https://doi.org/10.23919/EUSIPCO.2018.8553160.

Caldelli, R., R. Becarelli, and I. Amerini. 2017. "Image Origin Classification Based on Social Network Provenance." *IEEE Transactions on Information Forensics and Security* 12 (6): 1299–1308. https://doi.org/10.1109/TIFS.2017.2656842.

Carrara, F., F. Falchi, R. Caldelli, G. Amato, R. Fumarola, and R. Becarelli. 2017. "Detecting adversarial example attacks to deep neural networks," 1–7. 2008 IEEE Computer Society Conference on Computer Vision / Pattern Recognition Workshops, June.

———. 2019. "Adversarial image detection in deep neural networks." *Multimedia Tools and Applications* (February). https://doi.org/10.1007/s11042-018-5853-4. https://doi.org/10.1007/s11042-018-5853-4.

Castillo Camacho, I., and K. Wang. 2021. "A Comprehensive Review of Deep-Learning-Based Methods for Image Forensics." *Journal of Imaging,* Special Issue Image and Video Forensics, 7 (4): 69:1–39. https://doi.org/10.3390/jimaging7040069. https://hal.archives-ouvertes.fr/hal-03237610.

Celiktutan, Avcibas, Sankur, Ayerden, and Capar. 2006. "Source Cell-phone Identification," 1–3. 2006 IEEE 14th Signal Processing / Communications Applications. https://doi.org/10.1109/SIU.2006.1659882.

Chaumont, M. 2019. "Deep Learning in steganography and steganalysis from 2015 to 2018," arXiv: 1904.01444 `[cs.CR]`.

Chen, C., X. Zhao, and M. C. Stamm. 2018. "Mislgan: An Anti-Forensic Camera Model Falsification Framework Using A Generative Adversarial Network," 535–539. 2018 25th IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2018.8451503.

———. 2019. "Generative Adversarial Attacks Against Deep-Learning-Based Camera Model Identification." *IEEE Transactions on Information Forensics and Security,* 1–1. https://doi.org/10.1109/TIFS.2019.2945198.

Chen, J., X. Kang, Y. Liu, and Z. J. Wang. 2015. "Median Filtering Forensics Based on Convolutional Neural Networks." *IEEE Signal Processing Letters* 22, no. 11 (November): 1849–1853. https://doi.org/10.1109/LSP.2015.2438008.

Chen, M., J. Fridrich, M. Goljan, and J. Lukas. 2008. "Determining Image Origin and Integrity Using Sensor Noise." *IEEE Transactions on Information Forensics and Security* 3, no. 1 (March): 74–90. https://doi.org/10.1109/TIFS.2007.916285.

Chen, Y., Y. Huang, and X. Ding. 2017. "Camera model identification with residual neural network," 4337–4341. 2017 IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2017.8297101.

Chen, Z., B. Tondi, X. Li, R. Ni, Y. Zhao, and M. Barni. 2017. "A gradient-based pixel-domain attack against SVM detection of global image manipulations," 1–6. 2017 IEEE Workshop on Information Forensics / Security (WIFS). https://doi.org/10.1109/WIFS.2017.8267668.

Choi, H., H. Jang, D. Kim, J. Son, S. Mun, S. Choi, and H. Lee. 2017. "Detecting composite image manipulation based on deep neural networks," 1–5. 2017 International Conference on Systems, Signals / Image Processing (IWSSIP). https://doi.org/10.1109/IWSSIP.2017.7965621.

Choi, K. S., E. Y. Lam, and K. K. Y. Wong. 2006a. "Automatic source camera identification using the intrinsic lens radial distortion." *Optics express* 14 (December): 11551–65. https://doi.org/10.1364/OE.14.011551.

Choi, K. S., E. Y. Lam, and K. K. Y. Wong. 2006b. "Source camera identification using footprints from lens aberration," edited by Nitin Sampat, Jeffrey M. DiCarlo, and Russel A. Martin, 6069:172–179. International Society for Optics and Photonics, SPIE. https://doi.org/10.1117/12.649775. https://doi.org/10.1117/12.649775.

Christlein, V., C. Riess, J. Jordan, C. Riess, and E. Angelopoulou. 2012. "An Evaluation of Popular Copy-Move Forgery Detection Approaches." *IEEE Transactions on Information Forensics and Security* 7 (6): 1841–1854. https://doi.org/10.1109/TIFS.2012.2218597.

Cogranne, R., Q. Giboulot, and P. Bas. 2019. "The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis," 125–137. IH&amp;MMSec'19. Paris, France: Proceedings of the ACM Workshop on Information Hiding / Multimedia Security. ISBN: 9781450368216. https://doi.org/10.1145/3335203.3335726. https://doi.org/10.1145/3335203.3335726.

Costanzo, A., I. Amerini, R. Caldelli, and M. Barni. 2014. "Forensic Analysis of SIFT Keypoint Removal and Injection." *IEEE Transactions on Information Forensics and Security* 9 (9): 1450–1464. https://doi.org/10.1109/TIFS.2014.2337654.

Cozzolino, D., G. Poggi, and L. Verdoliva. 2015. "Efficient Dense-Field Copy–Move Forgery Detection." *IEEE Transactions on Information Forensics and Security* 10 (November). https://doi.org/10.1109/TIFS.2015.2455334.

———. 2017. "Recasting Residual-based Local Descriptors as Convolutional Neural Networks: an Application to Image Forgery Detection." *ArXiv,* abs/1703.04615.

Cozzolino, D., and L. Verdoliva. 2016. "Single-image splicing localization through autoencoder-based anomaly detection," 1–6. 2016 IEEE International Workshop on Information Forensics / Security (WIFS). https://doi.org/10.1109/WIFS.2016.7823921.

———. 2018. "Camera-based Image Forgery Localization using Convolutional Neural Networks," 1372–1376. 2018 26th European Signal Processing Conference (EUSIPCO). https://doi.org/10.23919/EUSIPCO.2018.8553581.

———. 2020. "Noiseprint: A CNN-Based Camera Model Fingerprint." *IEEE Transactions on Information Forensics and Security* 15:144–159. https://doi.org/10.1109/TIFS.2019.2916364.

D'Amiano, L., D. Cozzolino, G. Poggi, and L. Verdoliva. 2015. "Video forgery detection and localization based on 3D patchmatch," 1–6. 2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW). https://doi.org/10.1109/ICMEW.2015.7169805.

D'Amiano, L., D. Cozzolino, G. Poggi, and L. Verdoliva. 2019. "A PatchMatch-Based Dense-Field Algorithm for Video Copy–Move Detection and Localization." *IEEE Transactions on Circuits and Systems for Video Technology* 29 (3): 669–682. https://doi.org/10.1109/TCSVT.2018.2804768.

Dang-Nguyen, D.-T., C. Pasquini, V. Conotter, and G. Boato. 2015. "RAISE – A Raw Images Dataset for Digital Image Forensics." ACM Multimedia Systems, March.

de Carvalho, T.J., C. Riess, E. Angelopoulou, H. Pedrini, and A. de Rezende Rocha. 2013. "Exposing Digital Image Forgeries by Illumination Color Classification." *IEEE Transactions on Information Forensics and Security* 8 (7): 1182–1194. https://doi.org/10.1109/TIFS.2013.2265677.

De Rosa, A., M. Fontani, M. Massai, A. Piva, and M. Barni. 2015. "Second-Order Statistics Analysis to Cope With Contrast Enhancement Counter-Forensics." *IEEE Signal Processing Letters* 22 (8): 1132–1136. https://doi.org/10.1109/LSP.2015.2389241.

Ding, X., Y. Chen, Z. Tang, and Y. Huang. 2019. "Camera Identification Based on Domain Knowledge-Driven Deep Multi-Task Learning." *IEEE Access* 7:25878–25890. https://doi.org/10.1109/ACCESS.2019.2897360.

Dirik, A. E., and A. Karaküçük. 2014. "Forensic use of photo response non-uniformity of imaging sensors and a counter method." *Opt. Express* 22, no. 1 (January): 470–482. https://doi.org/10.1364/OE.22.000470. http://www.opticsexpress.org/abstract.cfm?URI=oe-22-1-470.

Donahue, J., Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. 2013. "DECAF: A Deep Convolutional Activation Feature for Generic Visual Recognition." *ArXiv* abs/1310.1531.

Dong, J., W. Wang, and T. Tan. 2013. "CASIA Image Tampering Detection Evaluation Database," 422–426. July. https://doi.org/10.1109/ChinaSIP.2013.6625374.

Farid, H. 2006. "Digital Image Ballistics from JPEG Quantization" (January).

Farid, H. 2009a. "Exposing Digital Forgeries From JPEG Ghosts." *IEEE Transactions on Information Forensics and Security* 4, no. 1 (March): 154–160. https://doi.org/10.1109/TIFS.2008.2012215.

———. 2009b. "Image forgery detection." *IEEE Signal Processing Magazine* 26, no. 2 (March): 16–25. https://doi.org/10.1109/MSP.2008.931079.

Ferrara, P., T. Bianchi, A. De Rosa, and A. Piva. 2012. "Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts." *IEEE Transactions on Information Forensics and Security* 7, no. 5 (October): 1566–1577. https://doi.org/10.1109/TIFS.2012.2202227.

Filler, T., J. Fridrich, and M. Goljan. 2008. "Using sensor pattern noise for camera model identification," 1296–1299. San Diego, CA, USA: 2008 15th IEEE International Conference on Image Processing.

Fiscus, J., H. Guan, Y. Lee, A. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, and X. Jin. 2020. *NIST Media Forensic Challenge (MFC) Evaluation 2020 - 4th Year DARPA MediFor PI meeting.*

Fridrich, J. 2009. "Digital image forensics." *IEEE Signal Processing Magazine* 26, no. 2 (March): 26–37. https://doi.org/10.1109/MSP.2008.931078.

Fridrich, J., and J. Kodovsky. 2012. "Rich Models for Steganalysis of Digital Images." *IEEE Transactions on Information Forensics and Security* 7, no. 3 (June): 868–882. https://doi.org/10.1109/TIFS.2012.2190402.

Galdi, C., F. Hartung, and J.-L. Dugelay. 2019. "SOCRatES: A Database of Realistic Data for SOurce Camera REcognition on Smartphones," 648–655. January. https://doi.org/10.5220/0007403706480655.

Galdi, C., M. Nappi, and J.-L. Dugelay. 2015. "Combining Hardwaremetry and Biometry for Human Authentication via Smartphones," 406–416. Springer. ISBN: 978-3-319-23234-8.

Gallagher, A. C., and T. Chen. 2008. "Image authentication by detecting traces of demosaicing," 1–8. 2008 IEEE Computer Society Conference on Computer Vision / Pattern Recognition Workshops, June. https://doi.org/10.1109/CVPRW.2008.4562984.

Gan, Y., J. Yang, and W. Lai. 2019. "Video Object Forgery Detection Algorithm Based on VGG-11 Convolutional Neural Network," 575–580. 2019 International Conference on Intelligent Computing, Automation / Systems (ICICAS). https://doi.org/10.1109/ICICAS48597.2019.00126.

Geradts, Z., J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh. 2001. "Methods for identification of images acquired with digital cameras." SPIE Optics East.

Giudice, O., A. Paratore, M. Moltisanti, and S. Battiato. 2017. "A Classification Engine for Image Ballistics of Social Data," 625–636. Cham: Springer International Publishing. ISBN: 978-3-319-68548-9.

Gloe, T., and R. Böhme. 2010. "The Dresden Image Database for Benchmarking Digital Image Forensics," 1584–1590. SAC '10. Sierre, Switzerland: ACM. ISBN: 978-1-60558-639-7. https://doi.org/10.1145/1774088.1774427. http://doi.acm.org/10.1145/1774088.1774427.

Goljan, M., J. Fridrich, and M. Chen. 2011. "Defending Against Fingerprint-Copy Attack in Sensor-Based Camera Identification." *IEEE Transactions on Information Forensics and Security* 6 (1): 227–236. https://doi.org/10.1109/TIFS.2010.2099220.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. *Generative Adversarial Networks.* arXiv: 1406.2661 [stat.ML].

Goodfellow, I. J., J. Shlens, and C. Szegedy. 2015. *Explaining and Harnessing Adversarial Examples.* arXiv: 1412.6572 [stat.ML].

Gragnaniello, D., F. Marra, G. Poggi, and L. Verdoliva. 2018. *Analysis of adversarial attacks against CNN-based image forgery detectors.* arXiv: 1808.08426 [cs.CV].

Guan, H., M. Kozak, E. Robertson, Y. Lee, A. N. Yates, A. Delgado, D. Zhou, T. Kheyrkhah, J. Smith, and J. Fiscus. 2019. "MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation," 63–72. 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). https://doi.org/10.1109/WACVW.2019.00018.

Güera, D., Y. Wang, L. Bondi, P. Bestagini, S. Tubaro, and E. J. Delp. 2017. "A Counter-Forensic Method for CNN-Based Camera Model Identification," 1840–1847. 2017 IEEE Conference on Computer Vision / Pattern Recognition Workshops (CVPRW). https://doi.org/10.1109/CVPRW.2017.230.

Hadwiger, B., and C. Riess. 2020. *The Forchheim Image Database for Camera Identification in the Wild.* ICPR Workshops.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep Residual Learning for Image Recognition," 770–778. 2016 IEEE Conference on Computer Vision / Pattern Recognition (CVPR), June. https://doi.org/10.1109/CVPR.2016.90.

Hosler, B., O. Mayer, B. Bayar, X. Zhao, C. Chen, J. A. Shackleford, and M. C. Stamm. 2019. "A Video Camera Model Identification System Using Deep Learning and Fusion," 8271–8275. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech / Signal Processing (ICASSP). https://doi.org/10.1109/ICASSP.2019.8682608.

Hosler, B. C., X. Zhao, O. Mayer, C. Chen, J. A. Shackleford, and M. C. Stamm. 2019. "The Video Authentication and Camera Identification Database: A New Database for Video Forensics." *IEEE Access* 7:76937–76948. https://doi.org/10.1109/ACCESS.2019.2922145.

Hsu, Y.-F., and S.-F. Chang. 2006. "Detecting Image Splicing Using Geometry Invariants and Camera Characteristics Consistency." Toronto, Canada: International Conference on Multimedia / Expo.

Huang, F., J. Huang, and Y. Q. Shi. 2010. "Detecting Double JPEG Compression With the Same Quantization Matrix." *IEEE Transactions on Information Forensics and Security* 5, no. 4 (December): 848–856. https://doi.org/10.1109/TIFS.2010.2072921.

Huang, G., Z. Liu, and K. Q. Weinberger. 2016. "Densely Connected Convolutional Networks." *CoRR* abs/1608.06993. arXiv: 1608.06993. http://arxiv.org/abs/1608.06993.

Huang, J., and C. X. Ling. 2005. "Using AUC and accuracy in evaluating learning algorithms." *IEEE Transactions on Knowledge and Data Engineering* 17, no. 3 (March): 299–310. https://doi.org/10.1109/TKDE.2005.50.

Huang, X., S. Wang, and G. Liu. 2018. "Detecting Double Jpeg Compression with Same Quantization Matrix Based on Dense Cnn Feature," 3813–3817. 2018 25th IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2018.8451569.

Johnson, M. K., and H. Farid. 2005. "Exposing Digital Forgeries by Detecting Inconsistencies in Lighting," 1–10. MM&#38;Sec '05. New York, NY, USA: ACM. ISBN: 1-59593-032-9. https://doi.org/10.1145/1073170.1073171. http://doi.acm.org/10.1145/1073170.1073171.

Johnson, M. K., and H. Farid. 2006. "Exposing Digital Forgeries Through Chromatic Aberration," 48–55. MM&#38;Sec '06. Geneva, Switzerland: ACM. ISBN: 1-59593-493-6. https://doi.org/10.1145/1161366.1161376. http://doi.acm.org/10.1145/1161366.1161376.

———. 2007. "Exposing Digital Forgeries in Complex Lighting Environments." *IEEE Transactions on Information Forensics and Security* 2 (3): 450–461. https://doi.org/10.1109/TIFS.2007.903848.

Junior, P. R. M., L. Bondi, P. Bestagini, S. Tubaro, and A. Rocha. 2019. "An In-Depth Study on Open-Set Camera Model Identification." *IEEE Access* 7:180713–180726. https://doi.org/10.1109/access.2019.2921436. https://doi.org/10.1109%5C%2Faccess.2019.2921436.

Kang, X., T. Qin, and H. Zeng. 2015. "Countering median filtering anti-forensics and performance evaluation of forensics against intentional attacks," 483–487. 2015 IEEE China Summit / International Conference on Signal / Information Processing (ChinaSIP). https://doi.org/10.1109/ChinaSIP.2015.7230449.

Karaküçük, A., and A. E. Dirik. 2015. "Adaptive photo-response non-uniformity noise removal against image source attribution." *Digital Investigation* 12:66–76. ISSN: 1742-2876. https://doi.org/https://doi.org/10.1016/j.diin.2015.01.017. https://www.sciencedirect.com/science/article/pii/S1742287615000183.

Kharrazi, M., H.T. Sencar, and N. Memon. 2004. "Blind source camera identification," vol. 1, 709–712 Vol. 1. 2004 International Conference on Image Processing, 2004. ICIP '04. https://doi.org/10.1109/ICIP.2004.1418853.

Kim, D., H. Jang, S. Mun, S. Choi, and H. Lee. 2018. "Median Filtered Image Restoration and Anti-Forensics Using Adversarial Networks." *IEEE Signal Processing Letters* 25 (2): 278–282. https://doi.org/10.1109/LSP.2017.2782363.

Kim, D.-H, and H.-Y Lee. 2017. "Image manipulation detection using convolutional neural network." *International Journal of Applied Engineering Research* 12 (January): 11640–11646.

Kim, H.-G., J.-S. Park, D.-G. Kim, and H.-K. Lee. 2018. "Two-stream neural networks to detect manipulation of JPEG compressed images." *Electronics Letters* 54 (6): 354–355. https://doi.org/https://doi.org/10.1049/el.2017.4444. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/el.2017.4444.

Kirchner, M., and R. Bohme. 2008. "Hiding Traces of Resampling in Digital Images." *IEEE Transactions on Information Forensics and Security* 3, no. 4 (December): 582–592. https://doi.org/10.1109/TIFS.2008.2008214.

Kirchner, M., and J. Fridrich. 2010. "On Detection of Median Filtering in Digital Images," 7541:754110. February. https://doi.org/10.1117/12.839100.

Kirchner, M., and T. Gloe. 2009. "On resampling detection in re-compressed images," 21–25. 2009 First IEEE International Workshop on Information Forensics / Security (WIFS), December. https://doi.org/10.1109/WIFS.2009.5386489.

Kirchner, M., and C. Johnson. 2020. *SPN-CNN: Boosting Sensor-Based Source Camera Attribution With Deep Learning.* https://doi.org/10.48550/ARXIV.2002.02927. https://arxiv.org/abs/2002.02927.

Korus, P. 2017. "Digital image integrity – a survey of protection and verification techniques." *Digital Signal Processing* 71:1–26. ISSN: 1051-2004. https://doi.org/https://doi.org/10.1016/j.dsp.2017.08.009. http://www.sciencedirect.com/science/article/pii/S1051200417301938.

Korus, P., and J. Huang. 2017. "Multi-Scale Analysis Strategies in PRNU-Based Tampering Localization." *IEEE Transactions on Information Forensics and Security* 12 (4): 809–824. https://doi.org/10.1109/TIFS.2016.2636089.

Krawetz, N. 2007. "A Picture ' s Worth . . . Digital Image Analysis and Forensics."

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks," 1097–1105. Curran Associates, Inc.

Kroese, D. P., T. J. Brereton, T. Taimre, and Z. I. Botev. 2014. "Why the Monte Carlo method is so important today." *Wiley Interdisciplinary Reviews: Computational Statistics* 6:386–392.

Kuzin, A., A. Fattakhov, I. Kibardin, V. I. Iglovikov, and R. Dautov. 2018. "Camera Model Identification Using Convolutional Neural Networks," 3107–3110. 2018 IEEE International Conference on Big Data (Big Data). https://doi.org/10.1109/BigData.2018.8622031.

Kwon, M.-J., S.-H. Nam, I.-J. Yu, H.-K. Lee, and C. Kim. 2021. "Learning JPEG Compression Artifacts for Image Manipulation Detection and Localization." *arXiv preprint arXiv:2108.12947.*

Lai, S., and R. Böhme. 2011. “Countering Counter-Forensics: The Case of JPEG Compression,” 6958:285–298. May. ISBN: 978-3-642-24177-2. https://doi.org/10.1007/978-3-642-24178-9_20.

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. “Gradient-based learning applied to document recognition.” *Proceedings of the IEEE* 86, no. 11 (November): 2278–2324. https://doi.org/10.1109/5.726791.

Li, C. 2010. “Source Camera Identification Using Enhanced Sensor Pattern Noise.” *IEEE Transactions on Information Forensics and Security* 5 (2): 280–287. https://doi.org/10.1109/TIFS.2010.2046268.

Li, H., W. Luo, and J. Huang. 2012. “Countering anti-JPEG compression forensics,” 241–244. 2012 19th IEEE International Conference on Image Processing. https://doi.org/10.1109/ICIP.2012.6466840.

Lin, H.-J., C.-W. Wang, and Y.-T. Kao. 2009. “Fast copy-move forgery detection.” *WSEAS Transactions on Signal Processing* 5 (May): 188–197.

Lin, M., Q. Chen, and S. Yan. 2014. *Network In Network.* arXiv: 1312.4400 [cs.NE].

Lin, T.-Y., M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. 2014. “Microsoft COCO: Common Objects in Context.” ECCV.

Lin, X., J.-H. Li, S.-L. Wang, A.-W.-C. Liew, F. Cheng, and X.-S. Huang. 2018. “Recent Advances in Passive Digital Image Security Forensics: A Brief Review.” Cybersecurity, *Engineering* 4 (1): 29–39. ISSN: 2095-8099. https://doi.org/https://doi.org/10.1016/j.eng.2018.02.008. http://www.sciencedirect.com/science/article/pii/S2095809917307890.

Lin, Z., J. He, X Tang, and C.-K. Tang. 2009. “Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis.” *Pattern Recognition* 42 (11): 2492–2501. ISSN: 0031-3203. https://doi.org/https://doi.org/10.1016/j.patcog.2009.03.019. http://www.sciencedirect.com/science/article/pii/S0031320309001198.

Liu, Z., H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. 2022. *A ConvNet for the 2020s.* https://doi.org/10.48550/ARXIV.2201.03545. https://arxiv.org/abs/2201.03545.

Long, Y., and Y. Huang. 2006. “Image Based Source Camera Identification using Demosaicking,” 419–424. 2006 IEEE Workshop on Multimedia Signal Processing. https://doi.org/10.1109/MMSP.2006.285343.

Lukas, J., and J. Fridrich. 2003. "Estimation of primary quantization matrix in double compressed JPEG images," 5–8. Proc. Digital forensic research workshop.

Lukas, J., J. Fridrich, and M. Goljan. 2006. "Digital Camera Identification From Sensor Pattern Noise." *IEEE Transactions on Information Forensics and Security* 1 (July): 205–214. https://doi.org/10.1109/TIFS.2006.873602.

Luo, W., Z. Qu, J. Huang, and G. Qiu. 2007. "A Novel Method for Detecting Cropped and Recompressed Image Block," 2:II-217-II–220. 2007 IEEE International Conference on Acoustics, Speech / Signal Processing - ICASSP '07, April. https://doi.org/10.1109/ICASSP.2007.366211.

Macdonald, H. 2004. "NRCS Photo Gallery," https://serc.carleton.edu/introgeo/interactive/examples/morrisonpuzzle.html.

Madry, A., A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. 2019. *Towards Deep Learning Models Resistant to Adversarial Attacks.* arXiv: `1706.06083 [stat.ML]`.

Mahdian, B., and S. Saic. 2008. "Blind Authentication Using Periodic Properties of Interpolation." *IEEE Transactions on Information Forensics and Security* 3, no. 3 (September): 529–538. https://doi.org/10.1109/TIFS.2004.924603.

———. 2009. "Using noise inconsistencies for blind image forensics." Special Section: Computer Vision Methods for Ambient Intelligence, *Image and Vision Computing* 27 (10): 1497–1503. ISSN: 0262-8856. https://doi.org/https://doi.org/10.1016/j.imavis.2009.02.001. http://www.sciencedirect.com/science/article/pii/S0262885609000146.

———. 2010. "A bibliography on blind methods for identifying image forgery." *Signal Processing: Image Communication* 25 (6): 389–399. ISSN: 0923-5965. https://doi.org/https://doi.org/10.1016/j.image.2010.05.003. http://www.sciencedirect.com/science/article/pii/S0923596510000536.

Mahfoudi, G., B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and M. Pic. 2019. "DEFACTO: Image and Face Manipulation Dataset," 1–5. 2019 27th European Signal Processing Conference (EUSIPCO). https://doi.org/10.23919/EUSIPCO.2019.8903181.

Mandelli, S., P. Bestagini, S. Tubaro, D. Cozzolino, and L. Verdoliva. 2018. "Blind Detection and Localization of Video Temporal Splicing Exploiting Sensor-Based Footprints," 1362–1366. 2018 26th European Signal Processing Conference (EUSIPCO). https://doi.org/10.23919/EUSIPCO.2018.8553511.

Mandelli, S., D. Cozzolino, P. Bestagini, L. Verdoliva, and S. Tubaro. 2020. "CNN-Based Fast Source Device Identification." *IEEE Signal Processing Letters* 27:1285–1289. https://doi.org/10.1109/LSP.2020.3008855.

Marra, F., D. Gragnaniello, and L. Verdoliva. 2018. "On the vulnerability of deep learning to adversarial attacks for camera model identification." *Signal Process. Image Commun.* 65:240–248.

Marra, F., D. Gragnaniello, L. Verdoliva, and G. Poggi. 2020. "A Full-Image Full-Resolution End-to-End-Trainable CNN Framework for Image Forgery Detection." *IEEE Access* 8:133488–133502. https://doi.org/10.1109/ACCESS.2020.3009877.

Marra, F., F. Roli, D. Cozzolino, C. Sansone, and L. Verdoliva. 2014. "Attacking the triangle test in sensor-based camera identification," 5307–5311. 2014 IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2014.7026074.

Mayer, O., B. Hosler, and M. C. Stamm. 2020. "Open Set Video Camera Model Verification," 2962–2966. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech / Signal Processing (ICASSP). https://doi.org/10.1109/ICASSP40776.2020.9054261.

Mayer, O., and M. C. Stamm. 2018. "Learned Forensic Source Similarity for Unknown Camera Models," 2012–2016. 2018 IEEE International Conference on Acoustics, Speech / Signal Processing (ICASSP). https://doi.org/10.1109/ICASSP.2018.8462585.

———. 2020. "Forensic Similarity for Digital Images." *IEEE Transactions on Information Forensics and Security* 15:1331–1346. https://doi.org/10.1109/TIFS.2019.2924552.

Mazumdar, A., J. Singh, Y. S. Tomar, and P. K. Bora. 2018. "Universal Image Manipulation Detection using Deep Siamese Convolutional Neural Network." *ArXiv* abs/1808.06323.

Meena, K. B., and V. Tyagi. 2019. "Image Forgery Detection: Survey and Future Directions," 163–194. April. ISBN: 978-981-13-6351-1. https://doi.org/10.1007/978-981-13-6351-1_14.

Mehrish, A., A. V. Subramanyam, and S. Emmanuel. 2019. "Joint Spatial and Discrete Cosine Transform Domain-Based Counter Forensics for Adaptive Contrast Enhancement." *IEEE Access* 7:27183–27195. https://doi.org/10.1109/ACCESS.2019.2901345.

Mentzer, F., G. Toderici, M. Tschannen, and E. Agustsson. 2020. "High-Fidelity Generative Image Compression," 33:11913–11924. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/8a50bae297807da9e97722a0b3fd8f27-Paper.pdf.

Mullan, P., D. Cozzolino, L. Verdoliva, and C. Riess. 2017. "Residual-based forensic comparison of video sequences," 1507–1511. 2017 IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2017.8296533.

Ng, T., and S. Chang. 2004. "A Data Set of Authentic and Spliced Image Blocks."

NIST. 2019. *Media Forensics Challenge 2019.* https://www.nist.gov/itl/iad/mig/media-forensics-challenge-2019-0.

Pan, D., L. Sun, R. Wang, X. Zhang, and R. O. Sinnott. 2020. "Deepfake Detection through Deep Learning," 134–143. 2020 IEEE/ACM International Conference on Big Data Computing, Applications / Technologies (BDCAT). https://doi.org/10.1109/BDCAT50828.2020.00001.

Papernot, N., P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. 2015. *The Limitations of Deep Learning in Adversarial Settings.* arXiv: 1511.07528 [cs.CR].

Park, J., D. Cho, W. Ahn, and H.-K. Lee. 2018. "Double JPEG Detection in Mixed JPEG Quality Factors using Deep Convolutional Neural Network." Proceedings of the European Conference on Computer Vision (ECCV).

Peng, A., H. Zeng, X. Lin, and X. Kang. 2015. "Countering anti-forensics of image resampling," 3595–3599. 2015 IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2015.7351474.

Pengpeng, Y., R. Ni, and Y. Zhao. 2017. "Recapture Image Forensics Based on Laplacian Convolutional Neural Networks," 10082:119–128. February. ISBN: 978-3-319-53464-0. https://doi.org/10.1007/978-3-319-53465-7_9.

Pham, N. T., J.-W. Lee, G.-R. Kwon, and C.-S. Park. 2019. "Hybrid Image-Retrieval Method for Image-Splicing Validation." *Symmetry* 11 (1). ISSN: 2073-8994. https://doi.org/10.3390/sym11010083. https://www.mdpi.com/2073-8994/11/1/83.

Popescu, A. C., and H. Farid. 2004. "Exposing digital forgeries by detecting duplicated image regions."

Popescu, A. C., and H. Farid. 2005a. “Exposing digital forgeries by detecting traces of resampling.” *IEEE Transactions on Signal Processing* 53, no. 2 (February): 758–767. https://doi.org/10.1109/TSP.2004.839932.

———. 2005b. “Exposing digital forgeries in color filter array interpolated images.” *IEEE Transactions on Signal Processing* 53, no. 10 (October): 3948–3959. https://doi.org/10.1109/TSP.2005.855406.

Al-Qershi, O. M., and B. E. Khoo. 2013. “Passive detection of copy-move forgery in digital images: State-of-the-art.” *Forensic Science International* 231 (1): 284–295. ISSN: 0379-0738. https://doi.org/https://doi.org/10.1016/j.forsciint.2013.05.027. http://www.sciencedirect.com/science/article/pii/S0379073813002971.

Qian, Y., J. Dong, W. Wang, and T. Tan. 2015. “Deep learning for steganalysis via convolutional neural networks.” *Proceedings of SPIE - The International Society for Optical Engineering* 9409 (March). https://doi.org/10.1117/12.2083479.

Rafi, A. M., T. I. Tonmoy, U. Kamal, Q. M. J. Wu, and Md. K. Hasan. 2019. *RemNet: Remnant Convolutional Neural Network for Camera Model Identification.* https://doi.org/10.48550/ARXIV.1902.00694. https://arxiv.org/abs/1902.00694.

Raj, A., and D. Sankar. 2019. “Counter Forensics: A New PRNU Based Method for Image Source Anonymization,” 1–7. 2019 IEEE International Conference on Electrical, Computer / Communication Technologies (ICECCT). https://doi.org/10.1109/ICECCT.2019.8868948.

Rao, Y., and J. Ni. 2016. “A deep learning approach to detection of splicing and copy-move forgeries in images,” 1–6. 2016 IEEE International Workshop on Information Forensics / Security (WIFS), December. https://doi.org/10.1109/WIFS.2016.7823911.

Rao, Y., J. Ni, and H. Zhao. 2020. “Deep Learning Local Descriptor for Image Splicing Detection and Localization.” *IEEE Access* 8:25611–25625. https://doi.org/10.1109/ACCESS.2020.2970735.

Redi, J., W. Taktak, and J.-L. Dugelay. 2011. “Digital image forensics: A booklet for beginners.” *Multimedia Tools Appl.* 51 (October): 133–162. https://doi.org/10.1007/s11042-010-0620-1.

Roy, A., R. S. Chakraborty, U. Sameer, and R. Naskar. 2017. “Camera Source Identification Using Discrete Cosine Transform Residue Features and Ensemble Classifier,” 1848–1854. 2017 IEEE Conference on Computer Vision / Pattern Recognition Workshops (CVPRW). https://doi.org/10.1109/CVPRW.2017.231.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, et al. 2015. *ImageNet Large Scale Visual Recognition Challenge.* arXiv: 1409.0575 `[cs.CV]`.

Salloum, R., Y. Ren, and C.-C. J. Kuo. 2018. “Image Splicing Localization Using A Multi-Task Fully Convolutional Network (MFCN).” *J. Visual Communication and Image Representation* 51:201–209.

Sameer, V. U., R. Naskar, and S. Modalavalasa. 2019. “Mitigating Adaptive PRNU Denoising in Camera Model Identification: An Anti–Counter Forensic Approach,” 903–907. TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON). https://doi.org/10.1109/TENCON.2019.8929355.

Schaefer, G., and M. Stich. 2004. “UCID: An uncompressed color image database,” 5307:472–480. January. https://doi.org/10.1117/12.525375.

Schöttle, P., A. Schlögl, C. Pasquini, and R. Böhme. 2018. “Detecting Adversarial Examples - a Lesson from Multimedia Security,” 947–951. 2018 26th European Signal Processing Conference (EUSIPCO). https://doi.org/10.23919/EUSIPCO.2018.8553164.

Sedighi, V., J. Fridrich, and R. Cogranne. 2016. “Toss that BOSSbase, Alice!” Media Watermarking, Security, and Forensics. San Francisco, United States: IS&T intl symposium on Electronic Imaging, February. https://hal.archives-ouvertes.fr/hal-01303577.

Sencar, H., and N. Memon. 2008. “Overview of State-of-the-Art in Digital Image Forensics.” 3 (November). https://doi.org/10.1142/9789812836243_0015.

Sheng, G., and Q. Su. 2014. “Erasing the JPEG Compression Artifacts: An Improved Counter-Forensic Algorithm Based on Parameter Adjustment,” 321–324. 2014 Ninth International Conference on Broadband / Wireless Computing, Communication / Applications. https://doi.org/10.1109/BWCCA.2014.83.

Shin, H., H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. 2016. “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning.” *IEEE Transactions on Medical Imaging* 35 (5): 1285–1298. https://doi.org/10.1109/TMI.2016.2528162.

Shullani, D., M. Fontani, M. Iuliani, O. Alshaya, and A. Piva. 2017. “VISION: a video and image dataset for source identification.” *EURASIP Journal on Information Security* 2017 (October): 15. https://doi.org/10.1186/s13635-017-0067-2.

Silva, E., T. Carvalho, A. Ferreira, and A. Rocha. 2015. “Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes.” *Journal of Visual Communication and Image Representation* 29:16–32. ISSN: 1047-3203. https://doi.org/https://doi.org/10.1016/j.jvcir.2015.01.016. https://www.sciencedirect.com/science/article/pii/S1047320315000231.

Simonyan, K., and A. Zisserman. 2014. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *CoRR* abs/1409.1556.

Singh, G., and K. Singh. 2019. “Counter JPEG Anti-Forensic Approach Based on the Second-Order Statistical Analysis.” *IEEE Transactions on Information Forensics and Security* 14 (5): 1194–1209. https://doi.org/10.1109/TIFS.2018.2871751.

Society, IEEE SP. 2018. “Camera Model Identification Competition.” *IEEE’s Signal Processing Society,* https://www.kaggle.com/competitions/sp-society-camera-model-identification/overview.

Stamm, M. C., S. K. Tjoa, W. S. Lin, and K. J. R. Liu. 2010. “Undetectable image tampering through JPEG compression anti-forensics,” 2109–2112. 2010 IEEE International Conference on Image Processing, September. https://doi.org/10.1109/ICIP.2010.5652553.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. “Going Deeper with Convolutions.” Computer Vision / Pattern Recognition (CVPR). http://arxiv.org/abs/1409.4842.

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. 2014. *Intriguing properties of neural networks.* arXiv: 1312.6199 [cs.CV].

Tan, M., and Q. V. Le. 2019. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," https://doi.org/10.48550/ARXIV.1905.11946. https://arxiv.org/abs/1905.11946.

Tan, W., Y. Wu, P. Wu, and B. Chen. 2019. "A Survey on Digital Image Copy-Move Forgery Localization Using Passive Techniques." *JNM : Journal of New Media* 1 (June): 11–25. https://doi.org/10.32604/jnm.2019.06219.

Tang, H., R. Ni, Y. Zhao, and X. Li. 2017. "Detection of various image operations based on CNN," 1479–1485. 2017 Asia-Pacific Signal / Information Processing Association Annual Summit / Conference (APSIPA ASC). https://doi.org/10.1109/APSIPA.2017.8282267.

Tariang, D. B., R. S. Chakraborty, and R. Naskar. 2019. "A Robust Residual Dense Neural Network For Countering Antiforensic Attack on Median Filtered Images." *IEEE Signal Processing Letters* 26 (8): 1132–1136. https://doi.org/10.1109/LSP.2019.2922498.

Toderici, G., D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell. 2017. "Full resolution image compression with recurrent neural networks," 5306–5314. Proceedings of the IEEE conference on Computer Vision / Pattern Recognition.

Tralic, D., I. Zupancic, S. Grgic, and M. Grgic. 2013. "CoMoFoD - New Database for Copy-Move Forgery Detection," 49–54. Proc. 55th International Symposium ELMAR-2013, September.

Tsai, M.-J., and G.-H. Wu. 2006. "Using Image Features to Identify Camera Sources," 2:II–II. 2006 IEEE International Conference on Acoustics Speech / Signal Processing Proceedings. https://doi.org/10.1109/ICASSP.2006.1660338.

Tuama, A., F. Comby, and M. Chaumont. 2016. "Camera model identification with the use of deep convolutional neural networks," 1–6. 2016 IEEE International Workshop on Information Forensics / Security (WIFS), December. https://doi.org/10.1109/WIFS.2016.7823908.

Valenzise, G., V. Nobile, M. Tagliasacchi, and S. Tubaro. 2011. "Countering JPEG anti-forensics," 1949–1952. 2011 18th IEEE International Conference on Image Processing. https://doi.org/10.1109/ICIP.2011.6115854.

Valenzise, G., M. Tagliasacchi, and S. Tubaro. 2013. "Revealing the Traces of JPEG Compression Anti-Forensics." *IEEE Transactions on Information Forensics and Security* 8, no. 2 (February): 335–349. https://doi.org/10.1109/TIFS.2012.2234117.

Valenzise, G., M. Tagliasacchi, and S. Tubaro. 2014. "Detectability-quality trade-off in JPEG counter-forensics," 5337–5341. 2014 IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2014.7026080.

Van, L. T., S. Emmanuel, and M. S. Kankanhalli. 2007. "Identifying Source Cell Phone using Chromatic Aberration," 883–886. 2007 IEEE International Conference on Multimedia / Expo, July. https://doi.org/10.1109/ICME.2007.4284792.

Van Lanh, T., K. Chong, S. Emmanuel, and M. S. Kankanhalli. 2007. "A Survey on Digital Camera Image Forensic Methods," 16–19. 2007 IEEE International Conference on Multimedia / Expo, July. https://doi.org/10.1109/ICME.2007.4284575.

Verdoliva, L. 2020. *Media Forensics and DeepFakes: an overview.* arXiv: 2001.06564 [cs.CV].

Verma, V., D. Singh, and N. Khanna. 2020. "Block-level double JPEG compression detection for image forgery localization." *arXiv preprint arXiv:2003.09393.*

Wang, J., K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al. 2020. "Deep high-resolution representation learning for visual recognition." *IEEE transactions on pattern analysis and machine intelligence* 43 (10): 3349–3364.

Wang, M., Z. Chen, W. Fan, and Z. Xiong. 2014. "Countering anti-forensics to wavelet-based compression," 5382–5386. 2014 IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2014.7026089.

Wang, Q., and R. Zhang. 2016. "Double JPEG compression forensics based on a convolutional neural network." *EURASIP Journal on Information Security* 2016 (December). https://doi.org/10.1186/s13635-016-0047-y.

Wei, Y., X. Bi, and B. Xiao. 2018. "C2R Net: The Coarse to Refined Network for Image Forgery Detection," 1656–1659. 2018 17th IEEE International Conference On Trust, Security. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00245.

Wen, B., Y. Zhu, R. Subramanian, T. Ng, X. Shen, and S. Winkler. 2016. "COVERAGE — A novel database for copy-move forgery detection," 161–165. 2016 IEEE International Conference on Image Processing (ICIP). https://doi.org/10.1109/ICIP.2016.7532339.

Wu, J., K. Feng, and M. Tian. 2020. "Review of Imaging Device Identification Based on Machine Learning." In *Proceedings of the 2020 12th International Conference on Machine Learning and Computing,* 105–110. ICMLC 2020. Shenzhen, China: Association for Computing Machinery. ISBN: 9781450376426. https://doi.org/10.1145/3383972.3384037. https://doi.org/10.1145/3383972.3384037.

Wu, Y., W. Abd-Almageed, and P. Natarajan. 2018a. "BusterNet: Detecting Copy-Move Image Forgery with Source/Target Localization." In *Proceedings of the European Conference on Computer Vision (ECCV).* September.

———. 2018b. "Image Copy-Move Forgery Detection via an End-to-End Deep Neural Network." In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV),* 1907–1915. https://doi.org/10.1109/WACV.2018.00211.

Wu, Y., W. AbdAlmageed, and P. Natarajan. 2019. "ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features," 9535–9544. 2019 IEEE/CVF Conference on Computer Vision / Pattern Recognition (CVPR). https://doi.org/10.1109/CVPR.2019.00977.

Wu, Z., M. C. Stamm, and K. J. R. Liu. 2013. "Anti-forensics of median filtering," 3043–3047. 2013 IEEE International Conference on Acoustics, Speech / Signal Processing. https://doi.org/10.1109/ICASSP.2013.6638217.

Xiao, J., J. Hays, K. Ehinger, A. Oliva, and A. Torralba. 2010. "SUN Database: Large-scale Scene Recognition from Abbey to Zoo." IEEE Conference on Computer Vision / Pattern Recognition.

Xu, G., H.-Z. Wu, and Y. Shi. 2016. "Ensemble of CNNs for Steganalysis: An Empirical Study," 103–107. June. https://doi.org/10.1145/2909827.2930798.

Yang, P., D. Baracchi, R. Ni, Y. Zhao, F. Argenti, and A. Piva. 2020. "A Survey of Deep Learning-Based Source Image Forensics." *Journal of Imaging* 6 (3). ISSN: 2313-433X. https://doi.org/10.3390/jimaging6030009. https://www.mdpi.com/2313-433X/6/3/9.

Ye, S., Q. Sun, and E. Chang. 2007. "Detecting Digital Image Forgeries by Measuring Inconsistencies of Blocking Artifact," 12–15. 2007 IEEE International Conference on Multimedia / Expo, July. https://doi.org/10.1109/ICME.2007.4284574.

Zampoglou, M., S. Papadopoulos, and I. Kompatsiaris. 2016. “Large-scale evaluation of splicing localization algorithms for web images.” *Multimedia Tools and Applications* (September). https://doi.org/10.1007/s11042-016-3795-2.

Zampoglou, M., S. Papadopoulos, and Y. Kompatsiaris. 2015. “Detecting image splicing in the wild (WEB).” In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW),* 1–6. https://doi.org/10.1109/ICMEW.2015.7169839.

Zeng, H., T. Qin, X. Kang, and L. Liu. 2014. “Countering anti-forensics of median filtering,” 2704–2708. 2014 IEEE International Conference on Acoustics, Speech / Signal Processing (ICASSP), May.

Zhan, Y., Y. Chen, Q. Zhang, and X. Kang. 2017. “Image Forensics Based on Transfer Learning and Convolutional Neural Network,” 165–170. IH&#38;MMSec ’17. Philadelphia, Pennsylvania, USA: ACM. ISBN: 978-1-4503-5061-7. https://doi.org/10.1145/3082031.3083250. http://doi.acm.org/10.1145/3082031.3083250.

Zhang, Y., J. Goh, L. L. Win, and V. L. L. Thing. 2016. “Image Region Forgery Detection: A Deep Learning Approach.” SG-CRC.

Zhao, M., B. Wang, F. Wei, M. Zhu, and X. Sui. 2020. “Source Camera Identification Based on Coupling Coding and Adaptive Filter.” *IEEE Access* 8:54431–54440. https://doi.org/10.1109/ACCESS.2019.2959627.

Zhao, W., Y. Pengpeng, R. Ni, Y. Zhao, and W. Li. 2019. “Cycle GAN-Based Attack on Recaptured Images to Fool both Human and Machine: 17th International Workshop, IWDW 2018, Jeju Island, Korea, October 22-24, 2018, Proceedings,” 83–92. January. ISBN: 978-3-030-11388-9. https://doi.org/10.1007/978-3-030-11389-6_7.

Zhong, J., and C. Pun. 2020. “An End-to-End Dense-InceptionNet for Image Copy-Move Forgery Detection.” *IEEE Transactions on Information Forensics and Security* 15:2134–2146. https://doi.org/10.1109/TIFS.2019.2957693.

Zhou, P., X. Han, V. I. Morariu, and L. S. Davis. 2018. “Learning Rich Features for Image Manipulation Detection.” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 1053–1061.