

Review

# A Survey of Deep Learning-Based Source Image Forensics

Pengpeng Yang <sup>1,2</sup>, Daniele Baracchi <sup>3</sup>, Rongrong Ni <sup>1,2,\*</sup>, Yao Zhao <sup>1,2</sup>, Fabrizio Argenti <sup>3</sup>  and Alessandro Piva <sup>3,\*</sup> 

<sup>1</sup> Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China; 14120339@bjtu.edu.cn (P.Y.); yzhao@bjtu.edu.cn (Y.Z.)

<sup>2</sup> Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing Jiaotong University, Beijing 100044, China

<sup>3</sup> Department of Information Engineering, University of Florence, Via di S. Marta, 3, 50139 Florence, Italy; daniele.baracchi@unifi.it (D.B.); fabrizio.argenti@unifi.it (F.A.)

\* Correspondence: rrni@bjtu.edu.cn (R.N.); alessandro.piva@unifi.it (A.P.)

Received: 7 February 2020; Accepted: 27 February 2020; Published: 4 March 2020



**Abstract:** Image source forensics is widely considered as one of the most effective ways to verify in a blind way digital image authenticity and integrity. In the last few years, many researchers have applied data-driven approaches to this task, inspired by the excellent performance obtained by those techniques on computer vision problems. In this survey, we present the most important data-driven algorithms that deal with the problem of image source forensics. To make order in this vast field, we have divided the area in five sub-topics: source camera identification, recaptured image forensic, computer graphics (CG) image forensic, GAN-generated image detection, and source social network identification. Moreover, we have included the works on anti-forensics and counter anti-forensics. For each of these tasks, we have highlighted advantages and limitations of the methods currently proposed in this promising and rich research field.

**Keywords:** image forensics; multimedia forensics; source identification; data driven methods

---

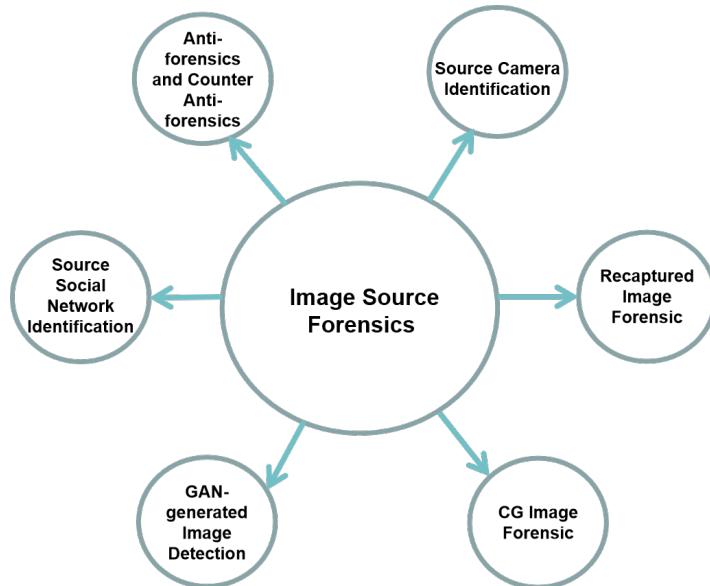
## 1. Introduction

With the development of modern techniques, digital imaging has become an important component in our daily life. It is easy for us to capture digital images with devices such as smartphones and digital reflex cameras, embellish them by using photo-editing software, and then upload them to social network platforms to share the important moments of our life with our friends. The ease with which we handle digital images, however, is a double-edged sword. Forged images are becoming more and more widespread in our life and seeing is no longer believing [1,2], especially with the advent of techniques based on artificial intelligence (AI) such as generative adversarial networks (GAN) [3] that can be exploited by malicious actors to spread “fake news” [4].

In order to verify the authenticity and integrity of a digital image, a number of techniques, known collectively as “digital image forensics” [5], were developed during the last years. Within this research area, source image forensics tries to answer the general question “where is this digital image from?”, and to do so multiple sub-topics can be explored, represented in Figure 1: source camera identification, recaptured image forensic, computer graphics (CG) image forensic, GAN-generated image detection, and source social network identification.

In the past the task of source image forensics has been dealt with several algorithms based on statistical analysis and pattern recognition. More recently, improvements in computing capabilities sparked a renewed interest in techniques based on machine learning. In particular, deep learning-based

schemes have been successfully applied in the field of source image forensics, and have proved their effectiveness in various competitions [6,7].



**Figure 1.** The structure of the presented review.

Techniques based on artificial neural networks have been known for many years and have gone by many names. A series of breakthroughs in 2006 [8–10], however, made the use of deep neural networks viable and gave rise to the field known as deep learning [11,12]. After that, methods based on deep learning have consistently achieved remarkable results on a series of tasks such as handwritten digital recognition and image classification, often beating competing approaches based on conventional schemes. A number of specialized techniques have been developed in the field of computer vision, including convolutional neural networks (CNN), recurrent neural networks (RNN), and generative adversarial networks (GAN). Among them, CNNs have been shown to be effective when dealing with image-related tasks, and have been subsequently adopted as a basis for numerous digital image forensics methods.

Basic components of CNNs consist mainly of convolutional layers, pooling layers, and activation functions, which are stacked together to construct the architecture of CNNs. According to Khan et al. [13], recent innovations in CNNs building and training can be categorized as structural reformulation, parameters optimizations, regularization, loss function. Among those, structural reformulation plays the most important role in improving the performance, and can be divided into seven different classes: spatial exploitation, depth, multi-path, width, channel boosting, feature map exploitation, and attention based CNNs. Typical CNN architectures such as AlexNet [14], VGGNet [15], GoogleNet [16], ResNet [17], DenseNet [18], Xception [19], SENet [20], Siamese Network [21], are well known. We refer the interested reader to the review by Khan et al. [13], Gu et al. [22] for further details about deep learning.

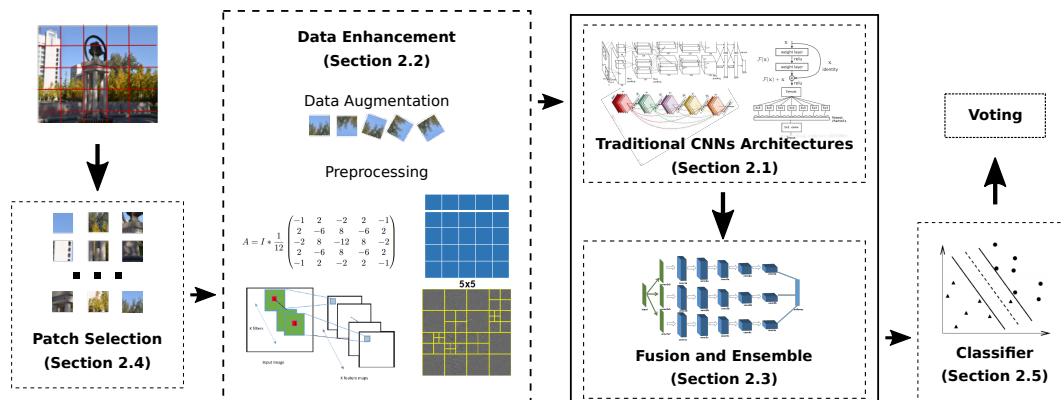
The impressive results obtained by deep learning-based methods in image source forensics motivate us to provide a comprehensive review of those approaches in such a way as to allow a neophyte to come into this field with some help. In this survey paper, we did our best to collect all the related paper which have been published in journals, conferences, and arXiv. We observed that all of them have some common modules, and **we propose a unifying framework** where all these schemes fit, thus simplifying their comparison. The framework is visually represented in Figure 2. First, the input, a full resolution image, is usually cropped into small and fixed-size pixel patches. Then, all patches are processed, or a patch selection strategy can be applied to choose the patches that are more useful for the following task. Next, these patches can be pre-processed by a spatial filter to

improve their signal-to-noise ratio (SNR). After that, they are fed to a convolutional neural network (CNN). Classification of a single patch can be achieved either by having a softmax layer at the end of the network, or by training a separate classifier on the features extracted by the last layer of the CNN. In the end, the classification result for the original image can be obtained by voting among the pixel patches. It should be noted that our general framework includes all the ideas used in the methods that we have reviewed; until now, however, no single method is using all those techniques. In Figure 2, optional parts are denoted by a dotted contour. Starting from the general framework, we have divided the reviewed algorithms in the following five separate sub-sections, according to their main contributions:

1. adoption of traditional convolutional neural networks (T.CNN) for source camera identification tasks;
2. improvement of performance by using data enhancement (D.A.), including data augmentation and data preprocessing;
3. improvement of performance through fusion and ensemble (F./E.);
4. improvement of performance by means of patch selection (P.S.);
5. adoption of different classifiers (C.).

The most significant network parameters of the reviewed works are summarized in Table 1 and Table 2. Then, in Tables 3 and 4, we have summarized the experimental settings and the performance achieved by those architectures, as well as by some conventional CNNs.

The remaining part of this review is organized as follows. First of all, in Section 2 we will describe the deep learning-based methods that deal with the most important topic of this area, i.e., source camera identification. Then, in Sections 3 to 6 we will present an overview on the methods dealing with the remaining sub-topics: recaptured image forensic, CG image forensic, GAN-generated image detection, and source social network identification, respectively. In Section 7 we will describe anti-forensics and counter-anti-forensics algorithms based on deep learning methods. A description of the results obtained by the described techniques will be given in Section 8. Finally, in Section 9, we will provide our analysis and conclusion on deep learning based source image forensics.



**Figure 2.** The framework of the deep learning-based algorithms for source forensics. (Section X.X) indicates the subsection where the related technique is described in detail.

## 2. Source Camera Identification

One of the hot topics in multimedia forensics is **source camera identification**, the purpose of which is to trace where an image is from. Identifying the source camera is an important step in pointing out the owner of **illicit** images (e.g., crime scenes, terroristic act scenes, etc.) and ensuring the security and trustworthiness of such digital data.

## 2.1. Traditional Convolutional Neural Networks (T.CNN)

Early works focused on applying to the problem of source camera identification traditional convolutional neural networks consisting of stacked convolutional layers. To the best of our knowledge, deep learning-based schemes for source camera identification were firstly introduced by Bondi et al. [23]. This path-breaking method used a simple architecture with five layers, including three convolutional steps and two fully connected layers, working on  $48 \times 48$  patches. The authors tested their network for both instance- and model-level camera identification, obtaining accuracies of 29.8% and 72.9%, respectively. Moreover, model-level camera identification increased to 94.1% on full resolution images with a voting strategy on the respective image patches. Focusing on images captured by mobile phones, which nowadays are the most popular image acquisition devices, Freire-Obregón et al. [24] proposed a six-layer CNN architecture, including two convolutional layers, one max pooling layer and three fully connected layers. The activation function used in this work was the Leaky Rectified Linear Unit (L-ReLU) which, as reported by the authors, led to slightly better performance than those obtained by using ReLU activations. Then, Huang et al. [25] presented an architecture similar to the one proposed by Bondi et al. [23], and the authors were able to improve over the accuracy obtained by Freire-Obregón et al. [24] by using Batch Normalization and more convolutional layers. Following along the line of deeper CNN architectures, Yao et al. [26] (code available at <https://github.com/grasses/Camera-Identification>) put forward a 13-layer convolutional neural network. The proposed method is robust against JPEG compression and noise adding; however, it is not resistant to re-scaling operation.

Chen et al. [27] investigated the use of a residual neural network (ResNet) with 26 layers for source camera identification, and proved its effectiveness with multiple experiments: the accuracies obtained for brand-, model-, and device-level identification were 99.12%, 94.73%, and 45.81%, respectively. According to their paper, ResNet has better performance than AlexNet, GoogleNet, and the scheme from Bondi et al. [23]. Ding et al. [28] extended this last method by combining ResNet architecture with a multi-task learning strategy, further improving the performance. The three tasks (brand-level, model-level, and sensor-level classification) are integrated into one framework and share the weights of the CNN architecture.

Several works [29–33] applied to source camera identification new architectures from computer vision such as DenseNet, XceptionNet. Marra et al. [29] used XceptionNet to obtain overall accuracies of 95.15% for pixel patches and 99.31% for the full resolution image by using a voting strategy.

All the works cited so far are intended to solve source camera identification in a closed-set scenario, where there is the assumption that we have a perfect knowledge of all the devices that could possibly capture the query images, which means that the acquisition devices for test data are the same used for acquiring training data. A more realistic case, however, is the open-set scenario, where information about query images is not completely known. Recently, a number of deep learning methods for this more challenging scenario have been proposed. Bayar and Stamm [34] presented two different schemes to address the open-set problem for camera model identification, which aims to judge whether the device that captured the query image is known or unknown. The first one uses, in place of a classification layer, a confidence score mapping with a thresholding strategy to evaluate whether the true source camera model is known or unknown. The other approach uses a different classifier on features extracted by a CNN. Mayer and Stamm [35] tried to measure similarity among images by using a siamese network. Features are extracted from the last layer of a CNN and fed into a siamese network to learn a measurement of source similarity, which allows for verifying if two query images are captured by same device or not.

It should be noted that the evolution of CNN architectures proposed for Source Camera Identification tasks closely follow the one of architectures proposed for computer vision tasks. This is somewhat to be expected as CNNs are well-suited for that kind of tasks, and thus a great deal of research on neural architectures produced by the community was focused on solving those problems.

Given the encouraging results obtained so far, it is logical to assume that the forensics community will keep on building on new CV architectures.

## 2.2. Data Enhancement (D.E.)

Data enhancement, including data augmentation and pre-processing, has been widely used as an effective way to improve the performance when dealing with computer vision tasks. These techniques have also been adopted for source camera identification schemes. Bondi et al. [36,37] normalized the images by subtracting the pixel-wise mean value, which is a popular way to center the data and helps the network to learn faster. Kamal et al. [31] applied five different data augmentation operations: random crops, random rotations, image manipulations (including JPEG compression, gamma correlation, and resizing), images addition, and empirical mode decomposition. It should be noted that the authors increased the training set size by collecting more images from Flickr (<https://www.flickr.com/>). The results demonstrated that adding more images has a great impact to performance: considering that deep learning-based methods are data-driven, increasing the number of training examples definitely leads to higher detection accuracies. Team GPU\_muscle [30,32] collected more than 500 GiB of photos from various resources (Flickr, Yandex.Fotki, Wikipedia Commons, mobile reviews, and others) and obtained an accuracy of more than 98% by training traditional CNNs. In addition, using manipulated images such as enhanced images with gamma correction, JPEG compressed images, and images transformed by resampling operation, to some extent, enhances the robustness of the CNN model.

Preprocessing techniques to improve the signal-to-noise ratio (SNR) of input data have been introduced following the intuition that the main difference between computer vision and image source identification tasks lies in the importance of image contents. Usually, computer vision tasks are seriously dependent on the image contents, whereas the opposite is true when dealing with source camera identification. In the latter case, the correct class to be attributed to an example is heavily dependent on the noise component introduced by camera acquisition. Based on this observation, some researches [38–42] proposed to reduce the interference of the image contents using two kinds of preprocessing. The first technique is based on the idea of applying a denoising filter  $F$  to the input image  $I$ , and then subtracting the result of that operation from  $I$  itself, thus obtaining the noise  $N$ :

$$N = I - F(I). \quad (1)$$

Tuama et al. [39] chose a wavelet-based denoiser, as filters of that kind have been widely used in model-based schemes based on Photo Response Non Uniformity (PRNU) for source camera identification. Bayar and Stamm [38] evaluated the effect of a median filter with  $3 \times 3$  windows (MFR). The second preprocessing technique is based on the idea that the noise can be easily obtained by using a spatial filter  $G$ :

$$N = I * G \quad (2)$$

Tuama et al. [39] also tested the effectiveness of an high-pass (HP) filter. According to that work, HP filters yield better results than wavelet-based denoiser when used in CNN-based schemes. Ding et al. [28] evaluated the case of gaussian filter residuals with three scales ( $3 \times 3, 5 \times 5, 7 \times 7$ ) and verified their effectiveness.

Following the idea of spatial filters, Yang et al. [42,43] presented self-learning filters as a way to further improve the SNR. Self-learning filters can be trained together with the rest of the network. Inspired by SPAM features [44], Bayar and Stamm [45] (code available at <https://gitlab.com/MISLgit/constrained-conv-TIFS2018>) proposed a novel constrained convolution which ensures that learned high-pass filters are within a given bound. In particular, the central weight of the convolutional kernel is set to  $-1$  and the sum of the other weights is equal to  $1$ . Instead of designing the filter, Wang et al. [41] used local binary patterns (LBP) to code the image. The images are first processed by

LBP coding operation in the preprocessing step, and then fed into CNN architecture. Self-learning filters, constrained convolutions, and LBP coding are reported to outperform high-pass filters.

Recently, Zuo [40] evaluated how the performance of CNN models are impacted by the use of two preprocessing techniques: laplacian gaussian smoothing filters and non-local means denoising filters. Results indicate that the CNN model without pre-processing provides better performances. The author gave a possible explanation from perspective of the training data and strategy. It should be noted, however, that the dataset used was only composed of images captured by three camera models, and that the number of epochs during training phase was small due to limitations in the available computing power. For these reasons, extended experiments should be conducted to verify the effect of preprocessing in future.

The analysis of data enhancement techniques used in source camera identification tasks highlight an important difference with respect to classical DE methods used in machine learning. Commonly used augmentation algorithms used by the ML community focus on helping the network become rotation- or scale-invariant by presenting image contents in different conditions. In source camera identification tasks, on the contrary, most DE technique aims to reduce the influence of image contents by filtering out information deemed not useful. As those two classes of enhancement methods are not mutually exclusive, future works could attempt to combine them to come up with a more comprehensive way to make the network more robust.

### 2.3. Fusion and Ensemble (F./E.)

Fusion and ensemble strategies aim to enhance performance by fusing multiple models and features together. Yang et al. [42] constructed a content-adaptive fusion network by merging three models together, thus significantly increasing the overall accuracy with respect to the single models. Bayar and Stamm [38] combined constrained convolutions and MFR at the first layer of the CNN architecture and obtained a slight improvement over a constrained CNN. Kamal et al. [31] used the ensemble feature of DenseNet201 trained using three image scales ( $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$ ), which is beneficial to the CNN model. Ferreira et al. [33] proposed to integrate InceptionNet and XceptionNet architectures to boost performance.

### 2.4. Patch Selection (P.S.)

It is well known that the PRNU noise that can be extracted from an image is related with its contents. Smooth, non-saturated areas with high luminance are good for PRNU estimation. Furthermore, content-adaptive processes such as CFA demosaicing and JPEG compression can be applied during image acquisition. Therefore, choosing different areas in the image could have some effect on the performance. Based on this fact, a good strategy for choosing the best pixel patches to be used for CNN training can be essential to obtain higher performance. Bondi et al. [36] (code available at <https://github.com/polimi-ispl/camera-model-identification-with-cnn>) only select for training the pixel patches whose average values are close to half of the image dynamic range. Another criterion for patch selection, which aims to find the better textured pixel patches with the half of the image dynamic, was proposed by Kamal et al. [31], Bondi et al. [37]. In this approach the quality value for a pixel patch is computed from its variance and mean. Pixel patches with higher measure value are used to train the CNN model. Güera et al. [46] proposed a CNN-based solution to estimate, for each pixel patch, a value representing the camera-model-attribution reliability. Yang et al. [43] adopted a different approach where pixel patches were separated into three subsets according to their mean and variance. Then, a different CNN model would be trained on each subset. Finally, query pixel patches would be classified using the model corresponding to their characteristics.

It appears that, currently, the dynamic range of a patch is considered to be the best descriptor for its usefulness for the task at hand. Experimental results confirm that such an approach is sensible. It would however be interesting to explore more diverse descriptor, perhaps taking into account the peculiarities of the neural network that will be subsequently used to classify the selected patches.

## 2.5. Classifier (C.)

Some computer vision works [47–49] claim that using a separate SVM classifier on the features learned by a CNN instead of a softmax layer can improve classification performance. In the same way, some image forensics researchers have recently explored if the adoption of different classifiers can improve the performance. In particular, Huang et al. [25], Kamal et al. [31], Bondi et al. [36,37] and Bayar and Stamm [34,38] proposed two-stage learning strategies by feeding the features extracted by a CNN model into a different classifier such as a Support Vector Machine (SVM), Extremely Randomized Trees (ERT), cosine similarity measure, nearest mean score, and deep learning architecture with squeeze and excitation block. The results confirm that these classifiers can achieve better performance with respect to simple softmax layers.

## 2.6. Summary

The most significant network parameters and the experimental settings of the reviewed works are summarized in Table 1, where we have identified twelve main architectures (using the short name  $A_1$ ,  $A_2$ , ...,  $A_{12}$ ). Then, in Table 3 we have summarized the experimental settings and the performance achieved by those architectures, as well as by some conventional CNNs.

## 3. Recaptured Image Forensic

Recaptured image forensic deals with the task of establishing whether an image has been recaptured or not, that is if it has been generated by capturing a printed picture or a screen display with an acquisition device. Yang et al. [50] presented an effective and practical deep learning-based method to address this problem: Laplacian Convolutional Neural Networks (L-CNN). In this technique the Laplacian filter is embedded into the first layer of a CNN to improve the noise signal ratio introduced by recapture operations. In that paper five different kinds of high-pass filters have been evaluated. Experimental results showed that performance obtained using laplacian filter is better than the one obtained by using other high-pass filters, or by not using filters at all. According to the paper, L-CNNs achieve 96% detection accuracy even when applied to  $64 \times 64$  image patches. To the best of our knowledge, this is the first work based on deep learning to detect recaptured images. Choi et al. [51] tested a nine-layer CNN on  $64 \times 64$  patches and obtained a detection result on the original images by using a voting strategy. The authors report slightly improved results over model-based methods. Li et al. [52] proposed a new framework by combining CNNs with RNNs. Instead of using a Laplacian filter in the first layer, the authors considered the convolutional operation as the preprocessing. The weights of the convolutional operator can be automatically learned during the training phase. Features extracted from trained CNN model were then fed into a recurrent neural network to classify the images. Aforementioned algorithms were evaluated on a small-scale dataset. Recently, Agarwal et al. [53] developed a diverse large-scale dataset for evaluating recaptured image forensic techniques. The dataset consists of 14,500 recaptured images and 14,500 original images. Those images were captured by various devices such as cameras, displays, scanners, printers. The authors also proposed an eight-layer CNN with 16 different kinds of gaussian filtering residuals in the first layer. The reported detection accuracy was up to 99.9% for  $64 \times 64$  pixels patches, which is a great improvement over model-based schemes.

The most significant network parameters and the experimental settings of the reviewed works are summarized in Table 2, where we have identified four main architectures (using the short name  $B_1$ ,  $B_2$ , ...,  $B_4$ ). Then, in Table 4 we have summarized the experimental settings and the performance achieved by those architectures, as well as by some conventional CNNs.

We can notice how most of the techniques proposed for recaptured image forensics are base on some kind of manual or automatic filtering. As an image's subject is not useful for establishing whether the image has been recaptured, those filters attempts to discard that information while, at the same time, highlighting the traces left by the recapturing operation.

**Table 1.** Summary of the reviewed architectures for source camera identification. CC and IC denote Constrained and Instant convolutional operations, respectively. GAP denotes the presence of a global average pooling operation. BN denotes the presence of batch normalization operations.

| Architecture | Input Size                | Preprocessing | Convolutional Part  |            |          |    | Fully Connected Part |          |            |         |
|--------------|---------------------------|---------------|---------------------|------------|----------|----|----------------------|----------|------------|---------|
|              |                           |               | N Layers            | Activation | Pooling  | BN | GAP                  | N Layers | Activation | Dropout |
| A1 [23]      | $48 \times 48 \times 3$   | -             | 3                   | ReLU       | Max      | -  | -                    | 1        | ReLU       | ✓       |
| A2 [24]      | $32 \times 32 \times 3$   | -             | 2                   | L-ReLU     | Max      | -  | -                    | 2        | L-ReLU     | ✓       |
| A3 [25]      | $36 \times 36 \times 3$   | -             | 3                   | ReLU       | Avg      | ✓  | -                    | 1        | ReLU       | ✓       |
| A4 [26]      | $64 \times 64 \times 3$   | -             | 13                  | ReLU       | Max      | -  | -                    | 2        | ?          | ✓       |
| A5 [27]      | $256 \times 256 \times 3$ | -             | 1 Conv, 12 Residual | ReLU       | -        | -  | ✓                    | -        | -          | -       |
| A6 [36]      | $64 \times 64 \times 3$   | -             | 4                   | ?          | Max      | -  | -                    | 1        | ReLU       | -       |
| A7 [37]      | $64 \times 64 \times 3$   | -             | 10                  | ?          | Max      | -  | -                    | 1        | ReLU       | -       |
| A8 [38]      | $256 \times 256 \times 2$ | IC + CC       | 4                   | TanH       | Max, Avg | ✓  | -                    | 2        | TanH       | -       |
| A9 [39]      | $256 \times 256$          | HP            | 3                   | ReLU       | Max      | -  | -                    | 2        | ReLU       | ✓       |
| A10 [41]     | $256 \times 256 \times 3$ | LBP           | 3                   | ReLU       | Max      | ✓  | -                    | 2        | ReLU       | ✓       |
| A11 [42]     | $64 \times 64 \times 3$   | -             | 6                   | ReLU       | Avg      | ✓  | ✓                    | -        | -          | -       |
| A12 [43]     | $64 \times 64 \times 3$   | -             | 1 Conv, 3 Residual  | ReLU       | Avg      | -  | ✓                    | -        | -          | -       |

**Table 2.** Summary of the reviewed architectures for recaptured image forensics (RF), CG image detection (CGI), GAN-generated images detection (GAN), and Source social networks identification (SSN). Lap denotes Laplacian filter, GR denotes the Guassian residuals. Col + Tex is the combination of Cb,Cr and the texture filtering responses by Schmidt filter bank. Filters denotes a combination of three kinds of high-pass filters used for steganalysis and DCT-His denotes the histogram of DCT coefficents.

| Architecture | Input size | Preprocessing             | Convolutional part |            |         |     |             | Fully connected part |            |           |
|--------------|------------|---------------------------|--------------------|------------|---------|-----|-------------|----------------------|------------|-----------|
|              |            |                           | N Layers           | Activation | Pooling | BN  | GAP / Stats | N Layers             | Activation | Dropout   |
| RF           | B1 [50]    | $N \times N \times 3$     | Lap                | 5          | ReLU    | Avg | ✓           | GAP                  | -          | -         |
|              | B2 [53]    | $64 \times 64 \times 1$   | GR                 | 6          | L-ReLU  | -   | ✓           | -                    | 1          | L-ReLU    |
|              | B3 [52]    | $32 \times 32 \times 3$   | Conv               | 2          | ReLU    | Avg | ✓           | -                    | 1          | ?         |
|              | B4 [51]    | $64 \times 64 \times 3$   | -                  | 6          | ReLU    | Max | -           | -                    | 2          | ReLU      |
| CGI          | C1 [54]    | $32 \times 32 \times 3$   | -                  | 6          | ReLU    | -   | -           | -                    | 2          | ReLU + BN |
|              | C2 [55]    | $96 \times 96$            | Col + Tex          | 4          | ReLU    | Avg | ✓           | -                    | 1          | ?         |
|              | C3 [56]    | $650 \times 650$          | Filters            | 5          | ReLU    | Avg | ✓           | GAP                  | -          | -         |
|              | C4 [57]    | NxN                       | Conv               | 3          | ReLU    | Max | ✓           | -                    | 1          | ReLU      |
|              | C5 [58]    | $100 \times 100 \times 1$ | -                  | 2          | -       | -   | -           | Stats                | 1          | ReLU      |
| GAN          | D1 [59]    | $N \times N \times 3$     | Lap                | 3          | L-ReLU  | Max | -           | -                    | 2          | L-ReLU    |
| SSN          | E1 [60]    | $64 \times 64$            | DCT-His            | 2          | ReLU    | Max | -           | -                    | 1          | ReLU      |
|              | E2 [61]    | $64 \times 64$            | PRNU               | 4          | ReLU    | Max | -           | -                    | 1          | ReLU      |

**Table 3.** The experimental setting for different algorithms. In this table, DA, FE, PS, C denote respectively data augmentation, fusion and ensemble, patch selection, and classifiers. The ratio between training and test data is shown in the column “Train : Test”. For the performance-patch/voting, the numbers between parenthesis denote the number of models/sensors in the test set. It should be noted that the evaluation matrix for 8, 20 is that  $0.7 \times (\text{accuracy of unaltered images}) + 0.3 \times (\text{accuracy of manipulated images})$ . Some works evaluated the performance on multiple datasets; only the most representative ones are shown in this table.

| Arch. | Input Size              | D.A.                      | F/E. | P.S. | C. | Train : Test | Dataset | Perf. (Patch) |                            | Perf. (Voting) |             |             |
|-------|-------------------------|---------------------------|------|------|----|--------------|---------|---------------|----------------------------|----------------|-------------|-------------|
|       |                         |                           |      |      |    |              |         | Model         | Sensor                     | Model          | Sensor      |             |
| [23]  | A1                      | $48 \times 48 \times 3$   | -    | -    | -  | Softmax      | 7:3     | Dresden [62]  | 72.9% (27)                 | 29.8% (74)     | 94.1% (27)  | -           |
| [24]  | A2                      | $32 \times 32 \times 3$   | -    | -    | -  | Softmax      |         | MICHE-I [63]  | 98.1% (3)                  | 91.1% (5)      | -           | -           |
| [25]  | A3                      | $36 \times 36 \times 3$   | -    | -    | -  | SVM          | 8:2     | Dresden [62]  | -                          | -              | -           | 99.9% (10)  |
| [26]  | A4                      | $64 \times 64 \times 3$   | -    | -    | ✓  | Softmax      | 3:2     | Dresden [62]  | 93% (25)                   | -              | >98% (25)   | -           |
| [27]  | A5                      | $256 \times 256 \times 3$ | -    | -    | -  | Softmax      | 7:3     | Dresden [62]  | 94.7% (27)                 | 45.8% (74)     | -           | -           |
|       | A6                      | $64 \times 64 \times 3$   |      |      |    |              |         |               | -                          | 80.77% (35)    | -           | 97.47% (35) |
| [29]  | DenseNet-40             | $32 \times 32 \times 3$   | -    | -    | -  | Softmax      | 8:2     | VISION [64]   | -                          | 87.96% (35)    | -           | 95.06% (35) |
|       | DenseNet-121            | $224 \times 224 \times 3$ |      |      |    |              |         |               | -                          | 93.88% (35)    | -           | 99.10% (35) |
|       | XceptionNet             | $299 \times 299 \times 3$ |      |      |    |              |         |               | -                          | 95.15% (35)    | -           | 99.31% (35) |
| [31]  | DenseNet-201 + SE-Block | $256 \times 256 \times 1$ | ✓    | ✓    | ✓  | SE-block     | 3.2:1   | SPC2018 [7]   | 98.37% (10, weighted)      | -              | -           | -           |
| [36]  | A6                      | $64 \times 64 \times 3$   | -    | -    | ✓  | SVM          |         | Dresden [62]  | 93% (18)                   | -              | >95 % (18)  | -           |
| [37]  | A7                      | $64 \times 64 \times 3$   | -    | -    | ✓  | Softmax      |         | Dresden [62]  | 94.93% (18)                | -              | -           | -           |
| [38]  | A8                      | $256 \times 256 \times 2$ | ✓    | ✓    | -  | ET           | 4:1     | Dresden [62]  | 98.58% (26)                | -              | -           | -           |
| [39]  | A9                      | $256 \times 256$          | -    | -    | -  | Softmax      | 8:2     | Dresden [62]  | 98.99% (12)<br>98.01% (14) | -              | -           | -           |
| [41]  | A10                     | $256 \times 256 \times 3$ | ✓    | -    | -  | Softmax      | 8:2     | Dresden [62]  | 98.78% (12)<br>97.41% (14) | -              | -           | -           |
| [43]  | A12                     | $64 \times 64 \times 3$   | ✓    | ✓    | ✓  | Softmax      | 4:1     | Dresden [62]  | -                          | 97.03% (9)     | -           | -           |
| [32]  | DenseNet-161            | $480 \times 480 \times 3$ | ✓    | -    | -  | Softmax      |         | SPC2018 [7]   | 98% (10, weighted)         | -              | -           | -           |
| [42]  | A11                     | $64 \times 64 \times 3$   | ✓    | ✓    | -  | Softmax      | 4:1     | Dresden [62]  | -                          | 94.14% (9)     | -           | -           |
| [33]  | Inception-Xception      | $299 \times 299$          | -    | ✓    | ✓  | Softmax      |         | SPC2018 [7]   | 93.29% (10, weighted)      | -              | -           | -           |
| [28]  | ResNet-modified         | $48 \times 48 \times 3$   | ✓    | -    | -  | Softmax      |         | Dresden [62]  | -                          | -              | 79.71% (27) | 53.4% (74)  |

**Table 4.** The experimental setting for different algorithms for recaptured image forensics (RF), CG image detection (CGI), and Source social networks identification (SSN). In this table, DA, FE, PS, C denote respectively data augmentation, fusion and ensemble, patch selection, and classifiers. The ratio between training and test data is shown in the column “Train : Test”. For the performance-patch/voting, the numbers between parenthesis denote the patch sizes, when applicable.

|     | Arch. | Input Size | D.A.                      | E/E. | P.S. | C. | Train : Test | Dataset  | Perf. (Patch)               | Perf. (Voting)  |              |
|-----|-------|------------|---------------------------|------|------|----|--------------|----------|-----------------------------|---|--------------|
| RF  | [50]  | B1         | $N \times N \times 3$     | ✓    | -    | -  | Softmax      | 1:1      | NTU-Rose [65]<br>LCD_R [66] | 99.74% (512)<br>99.30% (256)<br>98.48% (128)<br>95.23% (64) |              |
|     | [53]  | B2         | $64 \times 64 \times 1$   | ✓    | ✓    | -  | Softmax      | 8:2      | LS-D [53]                   | 99.90%  |              |
|     | [52]  | B3         | $32 \times 32 \times 3$   | ✓    | -    | -  | Softmax      | 1:1      | ASTAR [67]<br>NTU-Rose [65] | 86.78%<br>96.93%  |              |
|     |       |            |                           |      |      |    |              | ICL [68] | 97.79%                      | 99.54% (64)   |              |
| CGI | [51]  | B4         | $64 \times 64 \times 3$   | -    | -    | -  | Softmax      | 1:1      | ICL [68]                    | 85.73%  | 96.60%       |
|     | [54]  | C1         | $32 \times 32 \times 3$   | -    | -    | -  | Softmax      | 3:1      | Columbia [69]               | 98%   |              |
|     | [70]  | ResNet50   | $224 \times 224$          | -    | -    | -  | Softmax      | 5-f CV   | DSTok [71]                  | 96.1%   |              |
|     | [55]  | C2         | $96 \times 96$            | ✓    | ✓    | -  | Softmax      | 13:4     | 3Dlink [55]                 | 90.79%  | 94.87% (192) |
|     | [56]  | C3         | $650 \times 650$          | ✓    | -    | -  | Softmax      | 9:8      | WIFS [58]                   | 99.95%  | 100%         |
|     | [72]  | ResNet50   | ?                         | ✓    | -    | -  | Softmax      | 7:1      | Columbia [69]               | 98%   |              |
|     | [57]  | C4         | $233 \times 233$          | ✓    | -    | ✓  | Softmax      | 3:1      | Columbia [69]               | 85.15%  | 93.20%       |
|     | [58]  | C5         | $100 \times 100 \times 1$ | -    | ✓    | -  | MLP          | 8:2      | WIFS [58]                   | 84.80%  | 93.20%       |
|     | [73]  | VGG19      |                           | -    | ✓    | ✓  | MLP          | 5:2      | WIFS [58]                   | 96.55%  | 99.89%       |
|     | [74]  | ResNet50   | $224 \times 224 \times 3$ | -    | -    | -  | SVM          |          | DSTok [71]                  | 94%   |              |
| SSN | [60]  | E1         | $64 \times 64$            | ✓    | -    | -  | Softmax      | 9:1      | UCID [75]<br>PUBLIC [75]    | 98.41%<br>87.60%  | 95% (Avg.)   |
|     |       |            |                           |      |      |    |              |          | IPLAB [76]                  | 90.89%  |              |
|     | [61]  | E2         | $64 \times 64$            | ✓    | -    | -  | Softmax      | 9:1      | UCID [75]<br>VISION [64]    | 79.49%<br>98.50%  | 90.83%       |
|     |       |            |                           |      |      |    |              |          | IPLAB [76]                  |   | 83.85%       |

#### 4. Computer Graphics Image Forensic

Another possible source of a digital image is represented by Computer Graphics algorithms, so proper methods to detect this kind of origin have also been developed. In this field, Yu et al. [54] evaluated VGG-based architectures for CG image detection and found that their performance could be improved by dropping max-pooling layers. The authors explained that pooling layers could lead to a loss of association between adjacent pixels. Therefore, they presented a six-layer CNN without any pooling layer and achieved detection accuracies of over 98% on  $32 \times 32$  patches. He [70] tested several training strategies for VGG-19 and ResNet-50 architectures. Specifically, transfer learning technique was applied during the training phase and a fine-tuned ResNet-50 model was found to have the best performance. The authors report an average detection accuracy of about 96.1% on DSTok dataset. Building on the hybrid CNN-RNN approach of Li et al. [52], He et al. [55] presented a similar framework with a dual-path CNN to identify CG images. In this approach,  $96 \times 96$  patches are firstly converted to the YCbCr color space. Then, the luminance component is processed by a Schmidt filter bank to generate 13 different kinds of filtered responses. Lastly, the pair of chrominance components Cb,Cr and the filtered responses of the luminance component are separately fed into a four-layer CNN. Using this technique, the authors improved over the results obtained by Yu et al. [54] by 4 percentage points. Yao et al. [56] designed a five-layer CNN where the inputs are preprocessed by using high-pass filters. In this work, the authors explored three high-pass filters that were first introduced in the field of steganalysis: SQUARE5x5, SQUARE3x3, and EDGE3x3. Cui et al. [72] evaluated the use of ResNet-50 using the PRNU noise as input, and their architecture achieved a detection accuracy of 98% on Columbia Photographic Images and Photorealistic Computer Graphics Dataset. Instead of using fixed filters in the preprocessing step, Quan et al. [57] (code available at <https://github.com/weizequan/NIvsCG>) proposed a CNN with 32 convolutional operations in the first layer so that the weights of convolutional operator can be learned during the training phase. The results reported in their paper show that the performance of this method is better than the conventional methods, like Geo [77], SPAM [44], and Mfra [78], which indicate the effectiveness of CNN-based method on computer graphics (CG) image forensic. Rahmouni et al. [58] (code available at <https://github.com/NicoRahm/CGvsPhoto>) presented a novel statistical features extraction (SFE) layer and embed it between the last convolutional layer and the first fully connected layer. The SFE layer would extract four features: mean, variance, maximum, and minimum. The authors also explored feeding those features into different classifiers such as LDA and SVM. According to their paper, the best results are obtained by the CNN model trained in an end-to-end way. Continuing on this path, Nguyen et al. [73] improved detection performance by using a more powerful feature extractor: VGG-19. In this approach, the outputs of the convolutional operations before first three max pooling layers were extracted and the final features were calculated by computing their mean and variance. Then, three groups of those final features were fed into a fused 1-D CNN with two convolutional and three fully connected layers. Sharing a similar idea with the last two methods, De Rezende et al. [74] explored using other feature extractors and different classifiers: softmax, k-nearest neighbors, XGBoost, and SVM. In the end, ResNet-50 was chosen and the outputs of its 49th layer were used as features. The authors report that combining ResNet50 with a SVM classifier with RBF kernel achieved the best performance.

The most significant network parameters and the experimental settings of the reviewed works are summarized in Table 2, where we have identified five main architectures (using the short name C1, C2, ..., C5). Then, in Table 4 we have summarized the experimental settings and the performance achieved by those architectures, as well as by some conventional CNNs.

It can be noted that many recent works adopt approaches based on the extraction of statistical features from filtered images. All of them, however, only compute simple indicators such as mean and variance. It would be interesting to explore the possibility of using more sophisticated statistical features and whether those variant could yield an improvement on photo-realistic CG images.

## 5. GAN-Generated Image Detection

In recent years, a number of deep learning techniques capable of generating fake multimedia contents has been developed. Those methods, collectively called “[deepfakes](#)”, include autoencoders (AE) and generative adversarial networks (GAN). This pose significant challenges to the forensics community, as the contents generated by those techniques are much more realistic than the ones generated by computer graphics algorithms. Numerous researches and competitions [79] have focus on the detection of deepfake multimedia, such as the MFC2018 and DFDC launched by NIST and Facebook, respectively, and some surveys on this topic have been published. Verdoliva [80] presented an overview of media forensics and deepfakes. Nguyen et al. [81] came up with a survey of algorithms used to create deepfakes and, more importantly, methods to detect deepfakes proposed in the literature to date. Tolosana et al. [79] provided a thorough review of techniques for manipulating face images including DeepFake methods, and methods to detect such manipulations.

The most recent topic related to source image forensics is the detection of content generated by means of GANs. Here, Marra et al. [82] evaluated the performance of several image forensic detectors and popular computer vision CNN architectures on GAN-generated images detection. More specifically, the authors used four image forensic detectors: the method proposed by Fridrich and Kodovsky [83], the one by Cozzolino et al. [84], the one by Bayar and Stamm [85], and the one by Rahmouni et al. [58]. The reviewed CNN architectures were: DenseNet [18], InceptionNet v3 [86], XceptionNet [19], and the Cycle-GAN [87] discriminator. Experimental results showed that XceptionNet has the highest average detection accuracy even for images that have undergone Twitter-like compression. Haodong et al. [88] reported on an experimental investigation about the effectiveness of forensic detectors for GAN generated image detection, and in particular fake face images generated by Deep Convolutional Generative Adversarial Networks (DCGAN) [89] and Wasserstein Adversarial Networks (WGAN) [90]. Four approaches were evaluated in this work: GAN discriminator, face quality assessment, Inception score, and VGG-features with FLD. The best performance was obtained by using VGG-features with FLD. However, its generalization performances are limited when test set images are generated by different GAN schemes than the one used for training. Focusing on the differences in color composition between original and GAN generated images, Li et al. [91] presented a method where a feature set based on co-occurrences matrices is used to capture color image statistics. Firstly, the color image is transformed to RGB, HSV, and YCbCr spaces. Then, residual images would be generated for RGB, H, S, Cb, Cr channels, and co-occurrence matrices for all the residual images would be calculated. The extracted features would be finally fed into a binary classifier. The proposed method was evaluated on three public faces datasets (celebA [92], HQ-CelebA [93], LFW [94]) with four kinds of generated images (deep feature consistent variational auto-encoder (DFC-VAE) [95], DCGAN, WGAN-Gradient penalty (WGAN-GP) [96], Progressive Growing Generative Adversarial Networks (PGGAN) [93]), and it was able to obtain better performance than the one of model-based texture feature set [97]. McCloskey and Albright [98] extracted features from color and saturation space to detect PG-GAN generated image. On one hand, the standard rg chromaticity space is applied and the bivariate histograms of r, g components are fed into a INH network [99]. On the other hand, two groups of saturation measurements are extracted as features and SVM is used to classify PG-GAN generated images. The dataset produced in conjunction with the US National Institute of Standards and Technology’s Media Forensics Challenge 2018 was used to evaluate the performance of those two schemes. According to the report, saturation statistics provided better performance. Mo et al. [59] expected that the main difference between the original and GAN-generated images would be reflected on the residual domain. Therefore, they presented a three-layer CNN with a Laplacian filter preprocessing to identify fake face images generated by PG-GAN. From the perspective of artificial fingerprints, Marra et al. [100] explored a PRNU-based scheme for GAN-generated image detection. Three GAN architectures are considered in this work: Cycle-GAN, Pro-GAN, and Star-GAN. The results demonstrated that those GAN schemes would leave artificial fingerprints into the generated images.

The most significant network parameters and the experimental results of the reviewed works are summarized in Table 2, and in Table 5.

Based on our review of GAN-generated image detection works, we can broadly classify the proposed methods in three categories: existing detection methods; approaches based on the analysis of color characteristics of the images; and techniques based on the analysis of images' residual/noise. However, GANs are a very popular topic among machine learning researchers, and thus we expect GAN-generated images to become increasingly difficult to be detected. Existing techniques are likely to become obsolete in a short time, and for this reason forensics researchers will need to keep building new, effective detection methods.

**Table 5.** The statistical table for GAN-generated image detection. GAN shows the GAN model used to generate the images. Method represents the detection algorithm. Performance is the obtained accuracy unless otherwise specified. Only the best performance described in each paper is reported in this table.

|                       | GAN   | Dataset                                     | Method                           | Performance                    |
|-----------------------|---|---|----------------------------------|--------------------------------|
| [82]                  | Cycle-GAN [87]                                    | Cycle-GAN Data [87]                         | Cycle-GAN Discriminator [87]     | 83.58%                         |
|                       |   |   | Fridrich and Kodovsky [83]       | 94.40%                         |
|                       |   |   | Cozzolino et al. [84]            | 95.07%                         |
|                       |   |   | Bayar and Stamm [85]             | 84.86%                         |
|                       |   |   | Rahmouni et al. [58]             | 85.71%                         |
|                       |   |   | DenseNet [18]                    | 89.19%                         |
|                       |   |   | InceptionNet V3 [86]             | 89.09%                         |
| [88]                  | DC-GAN<br>W-GAN                                   | CelebA [92]                                 | XceptionNet [19]                 | 94.49%                         |
|                       |   |   | DCGAN Discriminator              | 95.51%                         |
|                       |   |   | VGG+FLD                          | >90 % (DC-GAN)<br>>94% (W-GAN) |
|                       |   |   |                                  |                                |
| [91]                  | DFC-VAE<br>DCGAN<br>WGAN-GP<br>PGGAN              | CelebAHQ [93]<br>CelebA [92]<br>LFW [94]    | Co-Color                         | 100%                           |
|                       |   |   |                                  |                                |
|                       |   |   | Lap-CNN                          | 96.3%                          |
|                       |   |   | RG-INHNet<br>Saturation Features | 0.56 (AUC)<br>0.7 (AUC)        |
| [59]<br>[98]<br>[100] | PG-GAN<br>GAN<br>Cycle-GAN<br>Pro-GAN<br>Star-GAN | CelebAHQ [93]<br>MFS2018 [6]<br>MFS2018 [6] | PRNU-based method                | 0.999 (AUC)                    |
|                       |   |   |                                  |                                |

## 6. Source Social Networks Identification

In recent years, social networks such as Facebook, Google+, and Twitter became more and more important in the daily life of a large part of the world population. According to Caldelli et al. [75], on average 350 million photo are uploaded daily on Facebook and around 60 millions monthly on Flickr. To the best of our knowledge, Amerini et al. [60] were the firsts to propose a CNN-based algorithm to identify from which social network a query image has been downloaded. The authors indicated that image manipulations applied by social networks usually include compressing as JPEG and resizing the original file, and that different social networks use different parameters for those operations. Therefore, it is possible to identify the source social network by looking at the discrete cosine transform (DCT) coefficients of the resulting image. Inspired by the work of Wang and Zhang [101], the authors extracted from each image the histogram of the first nine DCT coefficients and fed them into a simple 4-layer 1D CNN, obtaining great performance on source social network identification. Caldelli et al. [61] presented another scheme based on a 2D CNN with a preprocessing step. In the pre-processing stage, the PRNU noise of the image would be extracted. This noise would be then fed into a 6-layer 2D CNN. This technique obtains performance comparable with the one from the first method.

The most significant network parameters and the experimental settings of the reviewed works are summarized in Table 2, where we have identified the two main architectures as *E1* and *E2*. Then, in Table 4 we have summarized the experimental settings and the achieved performance.

The works presented in this section assume that images published on a social network will be re-encoded as JPEG after being processed by some unknown set of operations. Even if this is a sensible assumption, there is no guarantee that the specific processing pipeline used by a social network will stay the same over time. A simple software update on the backend software of a social platform could dramatically change how the images are processed, thus making an existing classifier obsolete. For this reason, all the methods described in this section require a constant update where social-network-processed images are continuously used to retrain the models. This is, however, inevitable due to the lack of knowledge about the inner working of social platforms.

## 7. Anti-Forensics and Counter Anti-Forensics

Image anti-forensics are techniques that aim to make forensics algorithms fail by modifying the images in a visually imperceptible way. Their goal is to prompt the image forensics community to come up with more robust forensics schemes. Recent developments in deep learning research led to the development of generative adversarial networks (GANs), novel techniques that proved to be very effective in deceiving many existing image forensics approaches. Güera et al. [102] trained a DenseNet-40 model for source camera identification and verified its vulnerability to Fast Gradient Sign Method (FGSM) [103] and Jacobian-based Saliency Map Approach (JSMA) [104] attacks. Marra et al. [29] evaluated the vulnerability of deep learning-based source camera identification algorithms to adversarial attacks in a more comprehensive manner. In this work, four kinds of deep learning-based methods (shallow CNN, DenseNet-40, DenseNet-121, XceptionNet) and two schemes of adversarial attacks (FGSM, Projected Gradient Descent (PGD) [105]) are tested to study their behaviour when classifying both pixel patches and full resolution images. According to the study, deep learning-based approaches are vulnerable to adversarial attacks. The authors report that, even if the robustness of deep learning-based methods can be improved by using adversarial training or training with JPEG-compressed images, the resulting networks are still vulnerable to targeted attacks. Zhao et al. [106] generated increasingly strong adversarial examples by using FGSM and Least-likely Class Method (LLCM) [107] and verified the vulnerability of deep learning-based image forensic algorithms. Fan et al. [108] proposed two kinds of gradient-based attacks against deep learning-based recaptured image forensic schemes: single attack and multiple attack. In order to verify the effectiveness of the proposed approach, the authors performed the attack on deep learning-based methods which employed adversarial training (as proposed by Szegedy et al. [109]) as a defense. The results indicated that single attack is ineffective on models that employ adversarial training, while multiple attack with a slow learning rate will obtain better results. Besides generating adversarial examples, generative adversarial networks attracted extensive attention because of their ability to generate photorealistic pictures and to achieve image-to-image translations. Two kinds of attacks against CNN models for source forensics have been proposed by modifying the GAN framework. Focusing on source camera identification, Chen et al. [110] defined a new loss function for the generator comprised of three terms: the perceptual loss, the classification loss, and the adversarial loss. More specifically, the perceptual loss describes the mean absolute difference between the original image and its falsified copy. The classification loss was designed to measure the difference between the output of the camera model identification classifier for the falsified image and the ideal output for the target camera model. The adversarial loss represents the standard loss function of GANs. In order to fool deep learning-based methods for recaptured image detection, Zhao et al. [111] proposed a Cycle-GAN-based scheme by fusing the adversarial loss, the cycle consistency loss and the low frequency consistency loss. In addition to the loss function used in Cycle-GAN, a low frequency consistency loss based on a median filter is proposed to keep the generated image similar to the original one.

Counter anti-forensics [112] methods have been proposed as a defense against anti-forensics techniques by improving the robustness of image forensics methods in case of anti-forensics attacks. Meanwhile, with the advent of adversarial examples, numerous approaches have been developed in

the field of computer vision to defend against adversarial attacks [113,114]. According to the review paper on the threat of adversarial attacks by Akhtar and Mian [113], until now defenses against adversarial attacks can be broadly divided into three categories. The first kind of defense is based on using modified examples either during training or during test. Adversarial training is based on this idea. The second kind resorts to modifying the networks by adding more layers, changing loss and so on. The last kind employs external models as network add-on when classifying unseen examples. It should be mentioned that deep learning and adversarial attacks greatly contributed to the cooperation between computer vision and image forensics communities, thus accelerating the development of related image forensics techniques. We will now introduce three preliminary works proposed by source image forensics researchers. Zhao et al. [106] combined adversarial training and regularization of input gradients as a defense against FGSM and LLCM attacks. Firstly, the regularization term of input gradients is added into the original loss function; then, the adversarial training strategy is applied to train the CNN model. Carrara et al. [115,116] used OverFeat, a well-known and successful deep convolutional network architecture, for the image representations. The features were extracted from the pool5 layer of the OverFeat and fed into a k-nearest neighbors (KNN) regressor to get a score. If the score is below threshold, the query image will be classified as an adversarial example and thrown away. From the perspective of image forensics, Schöttle et al. [117] presented an adversarial example detection scheme based on simple steganalysis features. Images that are not classified as adversarial examples will be distinguished by the CNN model [105]. It should be noted that those three approaches were only tested on some relatively old attack methods, such as FGSM, LLCM, FGS, PGD, which are easy to defend against according to the report by Carlini and Wagner [118].

More effective attack algorithms have been proposed in the field of computer vision; it is thus necessary to further study those methods in order to verify their performance in case of more powerful attacks. Thanks to the effort of Papernot et al. [119] a Python library which implements sixteen different kinds of adversarial attacks is available for other researchers to test their proposed defense schemes.

## 8. Evaluation Measures and Datasets

In this section we will describe the evaluation protocol and the datasets used in the reviewed papers.

The metric used in most of the papers is the accuracy:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

where  $TP, TN$  mean the number of correctly classified positive and negative cases,  $FP, FN$  represent the number of incorrectly classified positive and negative cases, respectively.

Besides, in order to fairly evaluate the performance for original and manipulated images in case of unbalanced datasets, a weighted accuracy is also used in some papers:

$$Acc_{\text{weighted}} = \frac{7}{10} Acc_{\text{unaltered}} + \frac{3}{10} Acc_{\text{altered}} \quad (4)$$

where  $Acc_{\text{unaltered}}, Acc_{\text{altered}}$  denote the accuracies in the case of unaltered images dataset and altered images dataset, respectively.

We will now introduce the publicly available image forensics datasets used for evaluating performance as shown in Tables 3 and 4. Datasets are organized according to the forensics topics. As some anti-forensics and counter anti-forensics methods use datasets introduced in the other sections, we will only report them once. Some of the datasets are freely downloadable from the authors' websites, while others can be obtained upon request.

### 8.1. Source Camera Identification

Dresden Image Dataset [62] was built for the purpose of developing and benchmarking camera-based digital forensic techniques. There are more than 14,000 images captured by 73 devices which belong to 25 different models. Dresden dataset was released in 2010, when smartphones were not yet a popular way to take pictures and therefore this dataset does not contain pictures taken with such devices.

VISION [64] is a video and image dataset for source identification. It is currently composed by 34,427 images and 1914 videos, both in the native format and in their social version (Facebook, YouTube, and WhatsApp are considered), from 35 modern smartphones/tablets of 11 major brands.

SPC2018 [7] is a dataset published in 2018 for the IEEE Signal Processing Cup competition whose topic was camera model identification. This dataset consists of 2750 images from ten different camera models (including point-and-shoot cameras, cell phone cameras, and digital single-lens reflex cameras), with 275 images captured using each camera model.

### 8.2. Recaptured Image Forensic

LS-D [53] is a large-scale dataset for evaluating recaptured image forensic. Four kinds of recapture attacks are considered: (1) photographing a printed copy of an image; (2) scanning a printed copy of an image; (3) photographing a displayed image; and (4) capturing a screen-grab of displayed image. This dataset consists of 145,000 pairs of original and recaptured images. A diverse set of devices has been used to recapture the images: 234 displays, 173 scanners, 282 printers, and 180 recaptured cameras.

NTU-Rose [65] is a collection of images recaptured from LCDs with good quality. It is composed of 2700 recaptured images captured by using three digital still cameras and three LCDs. The number of the original images is 300, including 100 images taken by the three cameras, 100 images downloaded from Flickr, and 100 tampered images.

ICL [68] is a dataset of images recaptured from a LCD and consists of 1035 original images taken by nine different cameras and 2520 images recaptured by using different devices. Camera settings were tuned in order to maintain a high image quality. Therefore, this dataset provides high-quality, high-resolution recaptured images.

ASTAR [67] is a smartphone images dataset for single image recaptured detection. The dataset is divided in three subsets. Subset A consists of 1094 real-scene images and 1137 recaptured images with real environment background. Subset B is built by cropping the real-scene images from Subset A, and by adding 1765 recaptured images without real environment backgrounds. Subset C consists of 587 pairs of single captured and recaptured images through transforming and cropping.

### 8.3. CG Image Detection

Columbia [69] consists of 800 CG images downloaded from Internet, 1200 images from personal collections, 800 original images from Google Image searches, and 800 recaptured CG images.

DSTok [71] includes CG and real photograph (PG) images collected from the Internet. There are 4850 pairs of CG and PG images. All of them were JPEG-compressed and the file sizes were between 12 KB and 1.8 MB.

WIFS [58] is built for new CG and PG images. There are 1800 CG images and 1800 PG images. CG images were downloaded from the Level Design Reference Database, which contains more than 60,000 good resolution video-game screenshots in JPEG format. Only five different video games were judged photo realistic enough to be included in WIFS, and thus only 1800 images were selected. PG images are high-resolution images taken from the RAISE dataset and directly converted to JPEG format.

3Dlink [55] consists of 6800 CG images download from the 3Dlink website and 6800 PG images captured under various environmental conditions by using different camera models.

#### 8.4. GAN-Generated Image Detection

MFS2018 [6] is a dataset released in 2018 for a media forensics challenge, which aims to help advance the state-of-the-art for image and video forensics techniques. For what concerns the works reviewed in this paper, only a subset of the dataset was used (GAN Crop and GAN Full).

#### 8.5. Social Network Identification

UCID social [75] is a UCID-based dataset for social network identification. Images from UCID were first JPEG-compressed with different quality factors. Then, those compressed images were uploaded to and subsequently downloaded from Flickr, Facebook, and Twitter. The UCID social dataset is composed of 30,000 images, 10,000 images from each social network.

The PUBLIC social dataset [75] consists of 3000 uncontrolled images with different sizes, JPEG quality factors and contents. Those images were directly downloaded from different social networks, including Flickr, Facebook, and Twitter, 1000 images for each one.

IPLAB [76] provides 2720 images in JPEG format. Ten social networks were considered for this dataset: Facebook, Google+, Twitter, Flickr, Instagram, Tumblr, Imgur, Tinypic, Whatsapp, and Telegram. Captured images were uploaded to the social networks either by using a web browser, or by using iOS and Android native apps.

### 9. Discussion and Conclusions

In this paper we presented a comprehensive survey of deep learning-based source image forensics, anti-forensics, and counter anti-forensics. According to our review, deep learning-based contributions for source image forensics can be divided into five categories, as shown in Figure 2: adopting traditional convolutional neural networks, or improving performance adopting strategies such as data enhancement, fusion and ensemble, patch selection, or using different classifiers. Most researchers based their methods on popular CNNs for computer vision, such as ResNet, XceptionNet, and DenseNet. Those architectures have proven to be effective when dealing with source image forensics tasks, even though they were originally developed for different scenarios. In particular, the most popular architecture for source image forensics appears to be ResNet, which strikes a balance between computational complexity and performance. As deep learning methods are data-driven, many contributions are focused on data enhancement techniques to improve the signal-to-noise ratio of data provided to CNNs. To do so, both separate pre-processing steps and customized network layers have been proposed. While many works reported an improvement in networks performance by using these methods, some authors obtained better results without using them. Unfortunately, these inconsistencies in different works are currently unavoidable: every method uses a different experimental protocol, thus making it impossible to compare the results. Moreover, as deep learning methods are heavily dependent on training data, it is difficult to declare a winner when comparing methods that have been evaluated on different datasets.

In conclusion, while many deep learning-based source image forensics methods have obtained remarkable results, there are still many research opportunities in this field worthy of being explored. Interested researchers may draw inspiration from the ever-expanding set of machine learning architectures and techniques to build new methods. Finally, it would be very important for the advancement of this research area to come up with a standard experimental protocol and shared datasets to make it possible to fairly compare the different proposed solutions.

**Funding:** This work was supported in part by the National Key Research and Development of China (No. 2016YFB0800404), the National NSF of China (Nos. 61672090, 61532005, 61332012) and the Fundamental Research Funds for the Central Universities (Nos. 2018JBZ001, 2017YJS054).

**Acknowledgments:** Pengpeng Yang would like to acknowledge the China Scholarship Council, State Scholarship Fund, that supports his joint Ph.D program.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhu, B.B.; Swanson, M.D.; Tewfik, A.H. When seeing isn't believing [multimedia authentication technologies]. *IEEE Signal Process. Mag.* **2004**, *21*, 40–49. [[CrossRef](#)]
2. Farid, H. Digital doctoring: How to tell the real from the fake. *Significance* **2006**, *3*, 162–166. [[CrossRef](#)]
3. Cao, Y.J.; Jia, L.L.; Chen, Y.X.; Lin, N.; Yang, C.; Zhang, B.; Liu, Z.; Li, X.X.; Dai, H.H. Recent Advances of Generative Adversarial Networks in Computer Vision. *IEEE Access* **2019**, *7*, 14985–15006. [[CrossRef](#)]
4. Beridze, I.; Butcher, J. When seeing is no longer believing. *Nat. Mach. Intell.* **2019**, *1*, 332–334. [[CrossRef](#)]
5. Piva, A. An overview on image forensics. *ISRN Signal Process.* **2013**, *2013*. [[CrossRef](#)]
6. Guan, H.; Kozak, M.; Robertson, E.; Lee, Y.; Yates, A.N.; Delgado, A.; Zhou, D.; Kheyrikhah, T.; Smith, J.; Fiscus, J. MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation. In Proceedings of the 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 63–72.
7. IEEE. *Signal Processing Society - Camera Model Identification*; IEEE: Piscataway, NJ, USA, 2018.
8. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [[CrossRef](#)]
9. Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2007; pp. 153–160.
10. Ranzato, M.; Poultney, C.; Chopra, S.; LeCun, Y. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2007; pp. 1137–1144.
11. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)]
12. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
13. Khan, A.; Sohail, A.; Zahoor, U.; Qureshi, A.S. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *arXiv* **2019**, arXiv:1901.06032.
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
15. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
16. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
19. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
20. Jie, H.; Li, S.; Albanie, S.; Gang, S.; Wu, E. Squeeze-and-Excitation Networks. *arXiv* **2017**, arXiv:1709.01507.
21. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*; IEEE Computer Society: Washington, DC, USA, 2005; CVPR '05; pp. 539–546. [[CrossRef](#)]
22. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroud, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [[CrossRef](#)]
23. Bondi, L.; Baroffio, L.; Güera, D.; Bestagini, P.; Delp, E.J.; Tubaro, S. Camera identification with deep convolutional networks. *arXiv* **2016**, arXiv:1603.01068.
24. Freire-Obregón, D.; Narducci, F.; Barra, S.; Castrillón-Santana, M. Deep learning for source camera identification on mobile devices. *Pattern Recognit. Lett.* **2019**, *126*, 86–91. [[CrossRef](#)]
25. Huang, N.; He, J.; Zhu, N.; Xuan, X.; Liu, G.; Chang, C. Identification of the source camera of images based on convolutional neural network. *Digit. Investig.* **2018**, *26*, 72–80. [[CrossRef](#)]

26. Yao, H.; Qiao, T.; Xu, M.; Zheng, N. Robust multi-classifier for camera model identification based on convolution neural network. *IEEE Access* **2018**, *6*, 24973–24982. [[CrossRef](#)]
27. Chen, Y.; Huang, Y.; Ding, X. Camera model identification with residual neural network. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 4337–4341.
28. Ding, X.; Chen, Y.; Tang, Z.; Huang, Y. Camera Identification based on Domain Knowledge-driven Deep Multi-task Learning. *IEEE Access* **2019**, *7*, 25878–25890. [[CrossRef](#)]
29. Marra, F.; Gragnaniello, D.; Verdoliva, L. On the vulnerability of deep learning to adversarial attacks for camera model identification. *Signal Process. Image Commun.* **2018**, *65*, 240–248. [[CrossRef](#)]
30. Stamm, M.; Bestagini, P.; Marcenaro, L.; Campisi, P. Forensic Camera Model Identification: Highlights from the IEEE Signal Processing Cup 2018 Student Competition [SP Competitions]. *IEEE Signal Process. Mag.* **2018**, *35*, 168–174. [[CrossRef](#)]
31. Kamal, U.; Rafi, A.M.; Hoque, R.; Das, S.; Abrar, A.; Hasan, M. Application of DenseNet in Camera Model Identification and Post-processing Detection. *arXiv* **2018**, arXiv:1809.00576.
32. Kuzin, A.; Fattakhov, A.; Kibardin, I.; Iglovikov, V.I.; Dautov, R. Camera Model Identification Using Convolutional Neural Networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 3107–3110.
33. Ferreira, A.; Chen, H.; Li, B.; Huang, J. An Inception-Based Data-Driven Ensemble Approach to Camera Model Identification. In Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, 11–13 December 2018; pp. 1–7.
34. Bayar, B.; Stamm, M.C. Towards open set camera model identification using a deep learning framework. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2007–2011.
35. Mayer, O.; Stamm, M.C. Learned forensic source similarity for unknown camera models. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2012–2016.
36. Bondi, L.; Baroffio, L.; Güera, D.; Bestagini, P.; Delp, E.J.; Tubaro, S. First steps toward camera model identification with convolutional neural networks. *IEEE Signal Process. Lett.* **2017**, *24*, 259–263. [[CrossRef](#)]
37. Bondi, L.; Güera, D.; Baroffio, L.; Bestagini, P.; Delp, E.J.; Tubaro, S. A preliminary study on convolutional neural networks for camera model identification. *Electron. Imaging* **2017**, *2017*, 67–76. [[CrossRef](#)]
38. Bayar, B.; Stamm, M.C. Augmented convolutional feature maps for robust cnn-based camera model identification. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 4098–4102.
39. Tuama, A.; Comby, F.; Chaumont, M. Camera model identification with the use of deep convolutional neural networks. In Proceedings of the 2016 IEEE International workshop on information forensics and security (WIFS), Abu Dhabi, UAE, 4–7 December 2016; pp. 1–6.
40. Zuo, Z. Camera Model Identification with Convolutional Neural Networks and Image Noise Pattern, 2018. Available online: <http://hdl.handle.net/2142/100123> (accessed on 2 July 2018).
41. Wang, B.; Yin, J.; Tan, S.; Li, Y.; Li, M. Source camera model identification based on convolutional neural networks with local binary patterns coding. *Signal Process. Image Commun.* **2018**, *68*, 162–168. [[CrossRef](#)]
42. Yang, P.; Zhao, W.; Ni, R.; Zhao, Y. Source camera identification based on content-adaptive fusion network. *arXiv* **2017**, arXiv:1703.04856.
43. Yang, P.; Ni, R.; Zhao, Y.; Zhao, W. Source camera identification based on content-adaptive fusion residual networks. *Pattern Recognit. Lett.* **2019**, *119*, 195–204. [[CrossRef](#)]
44. Pevny, T.; Bas, P.; Fridrich, J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 215–224. [[CrossRef](#)]
45. Bayar, B.; Stamm, M.C. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2691–2706. [[CrossRef](#)]
46. Güera, D.; Zhu, F.; Yarlagadda, S.K.; Tubaro, S.; Bestagini, P.; Delp, E.J. Reliability map estimation for CNN-based camera model attribution. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 964–973.
47. Tang, Y. Deep learning using linear support vector machines. *arXiv* **2013**, arXiv:1306.0239.

48. Chen, S.; Wang, Y. *Convolutional Neural Network and Convex Optimization*; Department of Electrical and Computer Engineering University of California San Diego: San Diego, CA, USA, 2014.
49. Abdulnabi, A.H.; Wang, G.; Lu, J.; Jia, K. Multi-task CNN model for attribute prediction. *IEEE Trans. Multimed.* **2015**, *17*, 1949–1959. [[CrossRef](#)]
50. Yang, P.; Ni, R.; Zhao, Y. Recapture image forensics based on Laplacian convolutional neural networks. In *International Workshop on Digital Watermarking*; Springer: Berlin, Germany, 2016; pp. 119–128.
51. Choi, H.Y.; Jang, H.U.; Son, J.; Kim, D.; Lee, H.K. Content Recapture Detection Based on Convolutional Neural Networks. In *International Conference on Information Science and Applications*; Springer: Berlin, Germany, 2017; pp. 339–346.
52. Li, H.; Wang, S.; Kot, A.C. Image recapture detection with convolutional and recurrent neural networks. *Electron. Imaging* **2017**, *2017*, 87–91. [[CrossRef](#)]
53. Agarwal, S.; Fan, W.; Farid, H. A diverse large-scale dataset for evaluating rebroadcast attacks. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 1997–2001.
54. Yu, I.J.; Kim, D.G.; Park, J.S.; Hou, J.U.; Choi, S.; Lee, H.K. Identifying photorealistic computer graphics using convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 4093–4097.
55. He, P.; Jiang, X.; Sun, T.; Li, H. Computer Graphics Identification Combining Convolutional and Recurrent Neural Networks. *IEEE Signal Process. Lett.* **2018**, *25*, 1369–1373. [[CrossRef](#)]
56. Yao, Y.; Hu, W.; Zhang, W.; Wu, T.; Shi, Y.Q. Distinguishing Computer-Generated Graphics from Natural Images Based on Sensor Pattern Noise and Deep Learning. *Sensors* **2018**, *18*, 1296. [[CrossRef](#)] [[PubMed](#)]
57. Quan, W.; Wang, K.; Yan, D.M.; Zhang, X. Distinguishing between natural and computer-generated images using convolutional neural networks. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2772–2787. [[CrossRef](#)]
58. Rahmouni, N.; Nozick, V.; Yamagishi, J.; Echizen, I. Distinguishing computer graphics from natural images using convolution neural networks. In Proceedings of the 2017 IEEE Workshop on Information Forensics and Security (WIFS), Rennes, France, 4–7 December 2017; pp. 1–6.
59. Mo, H.; Chen, B.; Luo, W. Fake Faces Identification via Convolutional Neural Network. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, Innsbruck, Austria, 20–22 June 2018; pp. 43–47.
60. Amerini, I.; Uricchio, T.; Caldelli, R. Tracing images back to their social network of origin: A cnn-based approach. In Proceedings of the 2017 IEEE Workshop on Information Forensics and Security (WIFS), Rennes, France, 4–7 December 2017; pp. 1–6.
61. Caldelli, R.; Amerini, I.; Li, C.T. PRNU-based Image Classification of Origin Social Network with CNN. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 1357–1361.
62. Gloe, T.; Böhme, R. The Dresden Image Database for benchmarking digital image forensics. In Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, 22–26 March 2010; pp. 1584–1590.
63. De Marsico, M.; Nappi, M.; Riccio, D.; Wechsler, H. Mobile iris challenge evaluation (MICHE)-I, biometric iris dataset and protocols. *Pattern Recognit. Lett.* **2015**, *57*, 17–23. [[CrossRef](#)]
64. Shullani, D.; Fontani, M.; Iuliani, M.; Al Shaya, O.; Piva, A. VISION: a video and image dataset for source identification. *EURASIP J. Inf. Secur.* **2017**, *2017*, 15. [[CrossRef](#)]
65. Cao, H.; Kot, A.C. Identification of recaptured photographs on LCD screens. In Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 14–19 March 2010; pp. 1790–1793.
66. Li, R.; Ni, R.; Zhao, Y. An effective detection method based on physical traits of recaptured images on LCD screens. In *International Workshop on Digital Watermarking*; Springer: Berlin, Germany, 2015; pp. 107–116.
67. Gao, X.; Qiu, B.; Shen, J.; Ng, T.T.; Shi, Y.Q. A smart phone image database for single image recapture detection. In *International Workshop on Digital Watermarking*; Springer: Berlin, Germany, 2010; pp. 90–104.
68. Thongkamwitton, T.; Muammar, H.; Dragotti, P.L. An image recapture detection algorithm based on learning dictionaries of edge profiles. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 953–968. [[CrossRef](#)]
69. Ng, T.T.; Chang, S.F.; Hsu, J.; Pepeljugoski, M. *Columbia Photographic Images and Photorealistic Computer Graphics Dataset*; ADVENT Technical Report Columbia University: New York, NY, USA, 2005; pp. 205–2004.

70. He, M. Distinguish computer generated and digital images: A CNN solution. *Concurrency Comput. Pract. Exp.* **2018**. [[CrossRef](#)]
71. Tokuda, E.; Pedrini, H.; Rocha, A. Computer generated images vs. digital photographs: A synergetic feature and classifier combination approach. *J. Vis. Commun. Image Represent.* **2013**, *24*, 1276–1292. [[CrossRef](#)]
72. Cui, Q.; McIntosh, S.; Sun, H. Identifying materials of photographic images and photorealistic computer generated graphics based on deep CNNs. *Comput. Mater. Contin.* **2018**, *55*, 229–241.
73. Nguyen, H.H.; Tieu, T.; Nguyen-Son, H.Q.; Nozick, V.; Yamagishi, J.; Echizen, I. Modular convolutional neural network for discriminating between computer-generated images and photographic images. In Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg, Germany, 27–30 August 2018; p. 1.
74. De Rezende, E.R.; Ruppert, G.C.; Theóphilo, A.; Tokuda, E.K.; Carvalho, T. Exposing computer generated images by using deep convolutional neural networks. *Signal Process. Image Commun.* **2018**, *66*, 113–126. [[CrossRef](#)]
75. Caldelli, R.; Becarelli, R.; Amerini, I. Image origin classification based on social network provenance. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1299–1308. [[CrossRef](#)]
76. Giudice, O.; Paratore, A.; Moltisanti, M.; Battiato, S. A classification engine for image ballistics of social data. In *International Conference on Image Analysis and Processing*; Springer: Berlin, Germany, 2017; pp. 625–636.
77. Ng, T.T.; Chang, S.F.; Hsu, J.; Xie, L.; Tsui, M.P. Physics-motivated features for distinguishing photographic images and computer graphics. In Proceedings of the 13th annual ACM international conference on Multimedia, Singapore City, Singapore, 6–11 November 2005; pp. 159–164.
78. Peng, F.; Zhou, D.L.; Long, M.; Sun, X.M. Discrimination of natural images and computer generated graphics based on multi-fractal and regression analysis. *AEU-Int. J. Electron. Commun.* **2017**, *71*, 72–81. [[CrossRef](#)]
79. Tolosana, R.; Vera-Rodriguez, R.; Fierrez, J.; Morales, A.; Ortega-Garcia, J. DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection. *arXiv* **2020**, arXiv:2001.00179.
80. Verdoliva, L. Media Forensics and DeepFakes: Sn overview. *arXiv* **2020**, arXiv:2001.06564.
81. Nguyen, T.T.; Nguyen, C.M.; Nguyen, D.T.; Nguyen, D.T.; Nahavandi, S. Deep Learning for Deepfakes Creation and Detection. *arXiv* **2019**, arXiv:1909.11573.
82. Marra, F.; Gragnaniello, D.; Cozzolino, D.; Verdoliva, L. Detection of GAN-generated fake images over social networks. In Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, FL, USA, 10–12 April 2018; pp. 384–389.
83. Fridrich, J.; Kodovsky, J. Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882. [[CrossRef](#)]
84. Cozzolino, D.; Poggi, G.; Verdoliva, L. Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection. In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, Paris, France, 3–5 July 2017; pp. 159–164.
85. Bayar, B.; Stamm, M.C. A deep learning approach to universal image manipulation detection using a new convolutional layer. In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, Vigo, Spain, 20–22 June 2016; pp. 5–10.
86. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
87. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
88. Haodong, L.; Han, C.; Bin, L.; Shunquan, T. Can Forensic Detectors Identify GAN Generated Images? In Proceedings of the APSIPA Annual Summit and Conference 2018, Honolulu, HI, USA, 12–15 November 2018; pp. 722–727.
89. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
90. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.

91. Li, H.; Li, B.; Tan, S.; Huang, J. Detection of deep network generated images using disparities in color components. *arXiv* **2018**, arXiv:1808.07276.
92. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision, Santiago, Chile, 7–13 December 2015; pp. 3730–3738.
93. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
94. Huang, G.B.; Mattar, M.; Berg, T.; Learned-Miller, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In Proceedings of the Workshop on faces in'Real-Life'Images: Detection, alignment, and recognition, Marseille, France, 17–18 October 2008.
95. Hou, X.; Shen, L.; Ke, S.; Qiu, G. Deep Feature Consistent Variational Autoencoder. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2016.
96. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of wasserstein GANs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
97. Boulkenafet, Z.; Jukka, K.; Abdenour, H. Face spoofing detection using colour texture analysis. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1818–1830. [[CrossRef](#)]
98. McCloskey, S.; Albright, M. Detecting GAN-generated Imagery using Color Cues. *arXiv* **2018**, arXiv:1812.08247.
99. Chen, C.; McCloskey, S.; Yu, J. Focus Manipulation Detection via Photometric Histogram Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1674–1682.
100. Marra, F.; Gragnaniello, D.; Verdoliva, L.; Poggi, G. Do GANs leave artificial fingerprints? *arXiv* **2018**, arXiv:1812.11842.
101. Wang, Q.; Zhang, R. Double JPEG compression forensics based on a convolutional neural network. *EURASIP J. Inf. Secur.* **2016**, *2016*, 23. [[CrossRef](#)]
102. Güera, D.; Wang, Y.; Bondi, L.; Bestagini, P.; Tubaro, S.; Delp, E.J. A counter-forensic method for cnn-based camera model identification. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1840–1847.
103. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572.
104. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. In Proceedings of the IEEE European Symposium on Security & Privacy, Saarbrucken, Germany, 21–24 March 2016.
105. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
106. Zhao, W.; Yang, P.; Ni, R.; Zhao, Y.; Wu, H. Security Consideration for Deep Learning-Based Image Forensics. *IEICE Trans. Inf. Syst.* **2018**, *101*, 3263–3266. [[CrossRef](#)]
107. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial Machine Learning at Scale. *arXiv* **2016**, arXiv:1611.01236.
108. Fan, W.; Agarwal, S.; Farid, H. Rebroadcast Attacks: Defenses, Reattacks, and Redefenses. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 942–946.
109. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
110. Chen, C.; Zhao, X.; Stamm, M.C. Misigan: An Anti-Forensic Camera Model Falsification Framework Using A Generative Adversarial Network. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 535–539.
111. Zhao, W.; Yang, P.; Ni, R.; Zhao, Y.; Li, W. Cycle GAN-Based Attack on Recaptured Images to Fool both Human and Machine. In *International Workshop on Digital Watermarking*; Springer: Berlin, Germany, 2018; pp. 83–92.
112. Barni, M.; Stamm, M.C.; Tondi, B. Adversarial Multimedia Forensics: Overview and Challenges Ahead. In Proceedings of the 26th European Signal Processing Conference (EUSIPCO), IEEE, Rome, Italy, 3–7 September 2018; pp. 962–966.

113. Akhtar, N.; Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **2018**, *6*, 14410–14430. [[CrossRef](#)]
114. Kurakin, A.; Goodfellow, I.; Bengio, S.; Dong, Y.; Liao, F.; Liang, M.; Pang, T.; Zhu, J.; Hu, X.; Xie, C.; et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*; Springer: Berlin, Germany, 2018; pp. 195–231.
115. Carrara, F.; Falchi, F.; Caldelli, R.; Amato, G.; Fumarola, R.; Becarelli, R. Detecting adversarial example attacks to deep neural networks. In Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, ACM, Florence, Italy, 19–21 June 2017; p. 38.
116. Carrara, F.; Falchi, F.; Caldelli, R.; Amato, G.; Becarelli, R. Adversarial image detection in deep neural networks. *Multimed. Tools Appl.* **2019**, *78*, 2815–2835. [[CrossRef](#)]
117. Schöttle, P.; Schlögl, A.; Pasquini, C.; Böhme, R. Detecting Adversarial Examples – a Lesson from Multimedia Security. In Proceedings of the 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 947–951.
118. Carlini, N.; Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, ACM, Dallas, TX, USA, 30 October–3 November 2017, pp. 3–14.
119. Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; et al. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv* **2018**, arXiv:1610.00768.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).