Chapter 7
Source Camera Model Identification

Sara Mandelli, Nicolò Bonettini, and Paolo Bestagini

Every camera model acquires images in a slightly different way. This may be due to differences in lenses and sensors. Alternatively, it may be due to the way each vendor applies characteristic image processing operations, from white balancing to compression. For this reason, images captured with the same camera model present a common series of artifacts that enable to distinguish them from other images. In this chapter, we focus on source camera model identification through pixel analysis. Solving the source camera model identification problem consists in identifying which camera model has been used to shoot the picture under analysis. More specifically, assuming that the picture has been digitally acquired with a camera, the goal is to identify the brand and model of the device used at acquisition time without the need to rely on metadata. Being able to attribute an image to the generating camera model may help forensic investigators to pinpoint the original creator of images distributed online, as well as to solve copyright infringement cases. For this reason, the forensics community has developed a wide series of methodologies to solve this problem.

7.1 Introduction

Given a digital image under analysis, we may ask ourselves a wide series of different questions about its origin. We may be interested in knowing whether the picture has been downloaded from a certain website. We may want to know which technology is used to digitalize the image (e.g., if it comes from a camera equipped with a rolling shutter or a scanner relying on a linear sensor). We may be curious about the model of camera used to shot the picture. Alternatively, we may need very specific details about the precise device instance that was used at image inception time. Despite all of these questions are related to source image attribution, they are very different in nature.

S. Mandelli · N. Bonettini · P. Bestagini (✉)
Politecnico di Milano, Milan, Italy
e-mail: paolo.bestagini@polimi.it

H. T. Sencar et al. (eds.), *Multimedia Forensics*, Advances in Computer Vision and Pattern Recognition, https://doi.org/10.1007/978-981-16-7621-5_7

Therefore, answering all of them is far from being an easy task. For this reason, the multimedia forensics community typically tackles one of these problems at a time.

In this chapter, we are interested in detecting the camera model that is used to acquire an image under analysis. Identifying the camera model which is used to acquire a photograph is possible thanks to the many peculiar traces left on the image at shooting time. To better understand which are the traces we are referring to, in this section we provide the reader with some background on the standard image acquisition pipeline. Finally, we provide the formal definition of the camera model identification problem considered in this chapter.

7.1.1 Image Acquisition Pipeline

We are used to shoot photographs everyday with our smartphones and cameras in a glimpse of an eye. Fractions of seconds pass from the moment we trigger the shutter to the moment we visualize the shot that we took. However, in this tiny amount of time, the camera performs a huge amount of operations.

The digital image acquisition pipeline is not unique, and may differ depending on the vendor, the device model and the available on-board technologies. However, it is reasonable to assume that a typical digital image acquisition pipeline is composed of a series of common steps (Ramanath et al. 2005), as shown in Fig. 7.1.

Light rays pass through a lens that focus them on the sensor. The sensor is typically a Charge-Coupled Device (CCD) or Complementary Metal-Oxide Semiconductor (CMOS), and can be imagined as a matrix of small elements geometrically organized on a plane. Each element represents a pixel, and returns a different voltage depending on the intensity of the light that hits it. Therefore, the higher the amount of captured light, the higher the output voltage, and the brighter the pixel value.

As these sensors react to light intensity, different strategies to capture color information may be applied. If multiple CCD or CMOS sensors are available, prisms can be used to split the light into different color components (typically red, green, and blue) that are directed to the different sensors. In this way, each sensor captures the intensity of a given color component, thus a color image can be readily obtained combining the output of each sensor. However, multiple sensors are typically available only on high-end devices, making this pipeline quite uncommon.

A more customary way of capturing color images consists in making use of a Color Filter Array (CFA) (or Bayer filter). This is a thin array of color filters placed

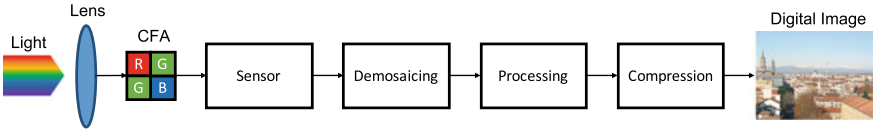


Fig. 7.1 Typical steps of a common image acquisition pipeline

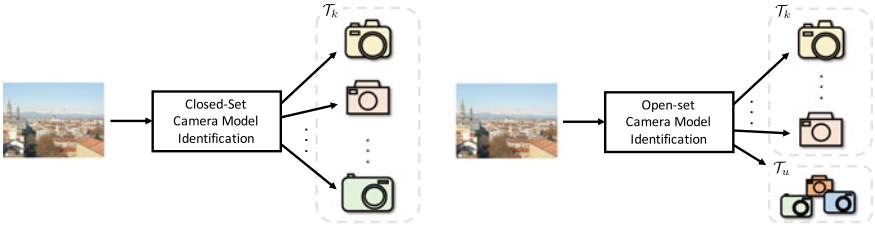
on top of the sensor. Due to these filters, each sensor's element is hit only by light in a narrow wavelength band corresponding to a specific color (typically red, green or blue). This means that the sensor returns the intensity of green light for certain pixels, the intensity of blue light for other pixels, and the intensity of red light for the remaining pixels. Which pixels capture which color depends on the shape of the CFA, which is a vendor choice. At this point, the output of the sensor consists of three partially sampled color layers in which only one color value is recorded at each pixel location. Missing color information, like the blue and red components for pixels that only acquired green light, are retrieved via interpolation from neighboring cells with the available color components. This procedure is known as debayering or demosaicing, and can be implemented using proprietary interpolation techniques.

After this raw version of a color image is obtained, a list of additional operations are in order. For instance, as lenses may introduce some kinds of optical distortion, most notably barrel distortion, pincushion distortion or combinations of them, it is common to apply some digital correction that may introduce forensic traces. Additionally, white balancing and color correction are other operations that are often applied and may be vendor-specific. Finally, lossy image compression is typically applied by means of JPEG standard, which again may vary with respect to compression quality and vendor-specific implementation choices.

Since a few years ago, these processing steps were the main sources of camera model artifacts. However, with the rapid proliferation of computational photography techniques, modern devices implement additional custom functionalities. This is the case of bokeh images (also known as portrait images) synthetically obtained through processing. These are pictures in which the background is digitally blurred with respect to the foreground object to obtain an artistic effect. Moreover, many devices implement the possibility of shooting High Dynamic Range (HDR) images, which are obtained by combining multiple exposures into a single one. Additionally, several smartphones are equipped with multiple cameras and produce pictures by mixing their outputs. Finally, many vendors introduce the possibility of shooting photographs with special filter effects that may enhance the picture in different artistic ways. All of these operations are custom and add traces to the pool of artifacts that can be exploited as a powerful asset for forensic analysis.

7.1.2 Problem Formulation

The problem of source camera model identification consists in detecting the model of the device used to capture an image. Although the definition of this problem seems pretty straightforward, depending on the working hypothesis and constraints, the problem may be cast in different ways. For instance, the analyst may only have access to a finite set of possible camera models. Alternatively, the forensic investigator may want to avoid using metadata. In the following, we provide the two camera model identification problem formulations that we consider in the rest of the chapter.



(a) Closed-set (b) Open-set

Fig. 7.2 Representation of closed-set (**a**) and open-set (**b**) camera model identification problem

In both formulations, we only focus on pixel-based analysis, i.e., we do not consider the possibility of relying on metadata information.

Closed-set Identification

Closed-set camera model identification refers to the problem of detecting which is the camera model used to shoot a picture within a set of known devices, as shown in Fig. 7.2a. In this scenario, the investigator assumes that the image under analysis has been taken with a device within a family of devices she/he is aware of. If the image does not come from any of those devices, the investigator will wrongly attribute the image to one of those known devices, no matter what.

Formally, let us define the set of labels of known camera models as $\mathcal{T}_k$. Moreover, let us define $\mathbf{I}$ as a color image acquired by the device characterized with label $t \in \mathcal{T}_k$. The goal of closed-set camera model identification is to provide an estimate $\hat{t}$ of t given the image $\mathbf{I}$ and the set $\mathcal{T}_k$. Notice that $\hat{t} \in \mathcal{T}_k$ by construction. Therefore, if the hypothesis that $t \in \mathcal{T}_k$ does not hold in practice, this approach is not applicable, as the condition $\hat{t} \in \mathcal{T}_k$ would imply that $\hat{t} \neq t$.

Open-set Identification

In many scenarios, it is not realistic to assume that the analyst has full control over the complete set of devices that may have been used to acquire the digital image under analysis. In this case, it is better to resort to the open-set problem formulation. The goal of open-set camera model identification is twofold, as shown in Fig. 7.2b. Given an image under analysis, the analyst aims

- To detect if the image comes from the set of known camera models or not.
- To detect the specific camera model, if the image comes from a known model.

This is basically a generalization of the closed-set formulation that accommodates for the analysis of images coming from unknown devices.

Formally, let us define the set of labels of known models as $\mathcal{T}_k$, and the set of labels of unknown models (i.e., all the other existing camera models) as $\mathcal{T}_u$. Moreover, let us define $\mathbf{I}$ as a color image acquired by the device characterized with label $t \in \mathcal{T}_k \bigcup \mathcal{T}_u$. The goal of open-set camera model identification is

- To estimate whether $t \in \mathcal{T}_k$ or $t \in \mathcal{T}_u$.
- To provide an estimate $\hat{t}$ of t in case $t \in \mathcal{T}_k$. In this case also $\hat{t} \in \mathcal{T}_k$.

Despite this problem formulation looks more realistic than its closed-set counterpart, the vast majority of camera model literature only copes with the closed-set problem. This is mainly due to the difficulty in well modeling the unknown set of models.

7.2 Model-Based Approaches

As mentioned in Sect. 7.1.1, multiple operations performed during image acquisition may be characteristic of a specific camera brand and model. This section provides an overview of camera model identification methods that work by specifically leveraging those traces. These methods are known as model-based methods, as they assume that each artifact can be modeled, and this model can be exploited to reverse engineer the used device. Methods that model each step of the acquisition chain are historically the first ones being developed in the literature (Swaminathan et al. 2009).

7.2.1 Color Filter Array (CFA)

As CCD/CMOS sensor elements of digital cameras are sensitive to the received light intensity and not to specific colors, the CFA is usually introduced in order to split the incoming light into three corresponding color components. Then, a three-channel color image can be obtained by interpolating the pixel information associated with the color components filtered by the CFA. There are several works in the literature that exploit specific characteristics associated with the CFA configuration, i.e., the specific arrangement of color filters in the sensor plane and the CFA interpolation algorithm to retrieve information about the source camera model.

CFA Configuration
Even without investigating the artifacts introduced by demosaicing, we can exploit information from the Bayer configuration to infer some model-specific features. For example, Takamatsu et al. (2010) developed a method to automatically identify CFA patterns from the distribution of image noise variances. In Kirchner (2010), the Bayer configuration was estimated by restoring the raw image (i.e., prior to demosaicing) from the output interpolated image. Authors of Cho et al. (2011) showed how to estimate the CFA pattern from a single image by extracting pixel statistics on 2×2 image blocks.

CFA Interpolation
In an image acquisition pipeline, the demosaicing step inevitably injects some inter-pixel correlations into the interpolated image. Existing demosaicing algorithms differ

in the size of their support region, in the way they select the pixel neighborhood, and in their assumptions about the image content and adaptability to this Gunturk et al. (2005), Menon and Calvagno (2011). Over the years, the forensics community has largely exploited these interpolation traces to discriminate among different camera models.

The first solution dates back to 2005 (Popescu and Farid 2005). The authors estimated the inter-pixel correlation weights from a small neighborhood around a given pixel, treating each color channel independently. They also observed that different interpolation algorithms present different correlation weights. Similarly, authors of Bayram et al. (2005) exploited the correlation weights estimated as shown in Popescu and Farid (2005) and combined them with frequency domain features. In Bayram et al. (2006), the authors improved upon their previous results by treating smooth and textured image regions differently. Their choice was motivated by the different treatment of distinct demosaicing implementations on high-contrast regions.

In 2007, Swaminathan et al. (2007) estimated the demosaicing weights only on interpolated pixels, i.e., without accounting for pixels which are relatively invariant to the CFA interpolation process. Authors found the CFA pattern by fitting linear filtering models and selecting the one which minimized the interpolation error. As done in Bayram et al. (2006), they considered three diverse estimations according to the local texture of the images. Interestingly, the authors' results pointed out some similarity in interpolation patterns among camera models from the same manufacturer.

For the first time, Cao and Kot (2009) did not explore each color channel independently, but instead exposed cross-channel pixel correlations caused by demosaicing. Authors reported that many state-of-the-art algorithms often employ color difference or hue domains for demosaicing, and this inevitably injects a strong correlation across image channels. In this vein, authors extended the work by Swaminathan et al. (2007) by estimating the CFA interpolation algorithm using a partial second-order derivative model to detect both intra-channel and cross-channel demosaicing correlations. They estimated these correlations by grouping pixels into 16 categories based on their CFA positions. In 2010, the same authors tackled the camera model identification problem on mobile phone devices, showing how CFA interpolation artifacts enable to achieve excellent classification results on dissimilar models, while confusing cameras of the same or very similar models (Cao and Kot 2010).

In line with Cao and Kot (2009), also Ho et al. (2010) exploited cross-channel interpolation traces. The authors measured the inter-channel differences of red and blue colors with respect to the green channel, and converted these into the frequency domain to estimate pixel correlations. They were motivated by the fact that many demosaicing algorithms interpolate color differences instead of color channels (Gunturk et al. 2005). Similarly, Gao et al. (2011) worked with variances of cross-channel differences.

Differently from previous solutions, in 2015, Chen and Stamm (2015) proposed a new framework to deal with camera model identification, inspired to the rich models proposed by Fridrich and Kodovský (2012) for steganalysis. Authors pointed out that previous solutions were limited by their essential use of linear or local linear parametric models, while modern demosaicing algorithms are both non-linear and

adaptive (Chen and Stamm 2015). The authors proposed to build a rich model of demosaicing by grouping together a set of non-parametric submodels, each capturing specific partial information on the interpolation algorithm. The resulting rich model could return a much more comprehensive representation of demosaicing compared to previous strategies.

Authors of Zhao and Stamm (2016) focused on controlling the computation cost associated with the solution proposed in Swaminathan et al. (2007), which relied on least squares estimation and could become impractical when a consistent number of pixels was employed. The authors proposed an algorithm to find the pixel set that yields the best estimate of the parametric model, still keeping the computational complexity feasible. Results showed that the proposed method could achieve higher camera model classification accuracy than Swaminathan et al. (2007), at a fixed computational cost.

7.2.2 Lens Effects

Every digital camera includes a sophisticated optical system to project the acquired light intensity on small CCD or CMOS sensors. Projection from the lens on to the sensor inevitably injects some distortions in the acquired image, usually known as optical aberrations. Among them, the most common optical aberrations are radial lens distortion, chromatic aberration, and vignetting. The interesting point from a forensics perspective is that different camera models use different optical systems, thus they reasonably introduce different distortions during image acquisition. For this reason, the forensics literature has widely exploited optical aberration as a model-based trace.

Radial Lens Distortion
Radial lens distortion is due to the fact that lenses in consumer cameras usually cannot magnify all the acquired regions with a constant magnification factor. Thus, different focal lengths and magnifications appear in different areas. This lens imperfection causes radial lens distortion which is a non-linear optical aberration that renders straight lines in the real image as curved lines on the sensor. Barrel distortion and pincushion distortion are the two main distortion forms we usually find in digital images. In San Choi et al. (2006), the authors measured the level of distortion of an image and used the distortion parameters as a feature to discriminate among different source camera models. Additionally, they investigated the impact of optical zoom on the reliability of the method, noticing that a classification beyond mid-range focal lengths can be problematic, due to the vanishing of distortion artifacts.

Chromatic Aberration
Chromatic aberration stems from wavelength-dependent variations of the refractive index of a lens. This phenomenon causes a spread of the color components over the sensor plane. The consequence of chromatic aberration is that color fringes appear

in high-contrast regions. We can identify axial chromatic aberration and lateral chromatic aberration. The former accounts for the variations of the focal point along the optical axis; the latter indicates the relative displacement of different light components along the sensor plane.

In 2007, Lanh et al. (2007) applied the model proposed in Johnson and Farid (2006) to estimate the lateral chromatic aberration using small patches extracted from the image center. Then, they fed the estimated parameters to a Support Vector Machine (SVM) classifier for identifying the source camera model. Later, Gloe et al. (2010) proposed to reduce the computational cost of the previous solution by locally estimating the chromatic aberration. The authors also pointed out some issues due to nonlinear phenomena occurring in modern lenses. Other studies were carried on in Yu et al. (2011), where authors pointed out a previously overlooked interaction between chromatic aberration and focal distance of lenses. Authors were able to obtain a stable chromatic aberration pattern distinguishing different copies of the same lens.

Vignetting

Vignetting is the phenomenon of light intensity fall-off around the corners of an image with respect to the image center. Usually, wide-aperture lenses are more prone to vignetting, because fewer light rays reach the sensor's edges.

The authors of Lyu (2010) estimated the vignetting pattern from images adopting a generalization of the vignetting model proposed in Kang and Weiss (2000). They exploited statistical properties of natural images in the derivative domain to perform a maximum likelihood estimation. The proposed method was tested among synthetically generated and real vignetting, showing far better results for lens model identification on synthetic data. The lower accuracy on real scenarios can be due to difficulties in correctly estimating the vignetting pattern whenever highly textured images are considered.

7.2.3 Other Processing and Defects

The traces left by the CFA configuration, the demosaicing algorithm, and the optical aberrations due to lens defects are only a few of the footprints that forensic analysts can exploit to infer camera model-related features. We report here some other model-based processing operations and defects that carry information about the camera model.

Sensor Dust

In 2008, Dirik et al. (2008) exploited the traces left by dust particles on the sensor of digital single-lens reflex cameras to perform source camera identification. The authors estimated the dust pattern (i.e., its location and shape) from the camera or from a number of images shot by the camera. The proposed methodology was robust

to both JPEG compression and downsizing operations. However, the authors pointed out some issues in correctly estimating the dust pattern for complex and non-smooth images, especially for low focal length values.

Noise Model

Modeling the noise pattern of acquired images can represent a valuable feature to distinguish among cameras of the same or different models.

Here, Photo Response Non Uniformity (PRNU) is arguably one of the most influential contributions to multimedia forensics. PRNU is a multiplicative noise pattern that occurs in any sensor-recorded digital image, due to imperfections in the sensor manufacturing process. In Lukás et al. (2006), Chen et al. (2008) the authors provided a complete modeling of the digital image at the sensor output as a function of the incoming light intensity and noise components. They discovered the PRNU noise to represent a powerful and robust device-related fingerprint, able to uniquely identify different devices of the same model. Since this chapter deals with model-level identification granularity, we do not further analyze the potential of PRNU for device identification.

Contrarily to PRNU-based methods, which model only the multiplicative noise term due to sensor imperfections, other methods focused on modeling the entire noise corrupting the digital image and exploited this noise to tackle the camera model identification task. For instance, Thai et al. (2014) worked with raw images at the sensor output. The authors modeled the complete noise contribution of natural raw images (known as the heteroscedastic noise Foi et al. 2009) with only two parameters, and used it as a fingerprint to discriminate camera models. Contrarily to PRNU noise, heteroscedastic noise could not separate different devices of the same camera model, thus it is more appropriate for model-level identification granularity.

The heteroscedastic noise consists of a Poisson-distributed component which accounts for the photon shot noise and dark current, and a Gaussian-distributed term which addresses other stationary noise sources like read-out noise (Foi et al. 2009). The proposed approach in Thai et al. (2014) involved the estimation of the two characteristic noise parameters per camera model considering 50 images shot by the same model. However, the method presented some limitations: first, it requires raw image format without post-processing or compression operations; second, non-linear processes like gamma correction modify the heteroscedastic noise model; finally, changes in ISO sensitivity and demosaicing operations worsen the detector performances.

In 2016, the authors improved upon their previous solution, solving the camera model identification from TIFF and JPEG compressed images and taking into account the non-linear effect of gamma correction (Thai et al. 2016). They proposed a generalized noise model starting from the previously exploited heteroscedastic noise but including also the effects of subsequent processing and compression steps. This way, each camera model could be described by three parameters. The authors tested their methodology over 18 camera models.

7.3 Data-Driven Approaches

Contrarily to model-based methods presented in Sect. 7.2, in the last few years we have seen the widespread adoption of data-driven approaches to camera model identification. All solutions that extract knowledge and insights from data without explicitly modeling the data behavior using statistical models fall into this category. These data-driven approaches have greatly outperformed multiple model-based solutions proposed in the past. While model-based solutions usually focus on a specific component of the image acquisition pipeline, data-driven approaches can capture model traces left by the interplay among multiple components. For instance, image noise characteristics do not only originate from sensor properties and imperfections, but are the result of CFA interpolation and other internal processing operations (Kirchner and Gloe 2015).

We can further divide data-driven approaches into two broad categories:

1. Methods based on hand-crafted features, which derive properties of data by extracting suitable data descriptors like repeated image patterns, texture, and gradient orientations.
2. Methods based on learned features, which directly exploit raw data, i.e., without extracting any descriptor, to learn distinguishable data properties and to solve the specific task.

7.3.1 Hand-Crafted Features

Over the years, the multimedia forensics community has often imported image descriptors from other research domains to solve the camera model identification problem. For instance, descriptors developed for steganalysis and image classification have been widely used to infer forensic traces on images. The reason behind their application in the forensics field is that, in general, these descriptors enable an effective representation of images and contain relevant information to distinguish among different image sources.

Surprisingly, the very first proposed solution to solve the camera model identification problem in a blind setup exploited hand-crafted features (Kharrazi et al. 2004). The authors extracted descriptors about color channel distributions, wavelet statistics and image quality metrics, then they fed these to an SVM classifier to distinguish among few camera models.

In the following, we illustrate the existing solutions based on hand-crafted feature extraction.

Local Binary Patterns

Local binary patterns (Ojala et al. 2002) represent a good example of a general image descriptor that can capture various image characteristics. In a nutshell, local binary patterns can be computed as the difference between a central pixel and its

local neighborhood, binary quantized and coded to produce an histogram carrying information about the local inter-pixel relations.

In 2012, the authors of Xu and Shi (2012) were inspired by the idea that the entire image acquisition pipeline could generate localized artifacts in the final image, and these characteristics could be captured by the uniform grayscale invariant local binary patterns (Ojala et al. 2002). The authors extracted features for red and green color channel in the spatial and wavelet domains, resulting in a 354-dimension feature per image. As commonly done in the majority of works previous to the wide adoption of neural networks, the authors fed these features to an SVM classifier to classify image sources among 18 camera models.

DCT Domain Features

Since the vast majority of cameras automatically stores the acquired images in JPEG format, many forensics works approach camera model identification by exploiting some JPEG-related features. In particular, many researches focused on extracting model-related features in the Discrete Cosine Transform (DCT) domain.

In 2009, Xu et al. (2009) extracted the absolute value of the quantized 8×8 DCT coefficient blocks, then computed the difference between the element blocks along four directions to look for inter-coefficient correlations. They fed these to Markov transition probability matrices to identify statistical differences inside images of distinct camera models. The elements of transition probability matrices were thresholded and fed as hand-crafted features to an SVM classifier.

DCT domain features have been exploited also in Wahab et al. (2012), where the authors extracted some conditional probability features from the absolute values of three 8×8 DCT coefficient blocks. From these blocks, they only used the 4×4 upper left frequencies which demonstrated to be most significant for the model identification task. The authors fed an SVM classifier to distinguish among 4 camera models.

More recently, the authors of Bonettini et al. (2018) have shown that JPEG eigenalgorithm features can be used for camera model identification as well. These features capture the differences among DCT coefficients obtained after multiple JPEG compression steps. A standard random forest classifier is used for the task.

Color Features

Camera model identification can be faced also by looking at some model-specific image color features. Such artifacts can be found according to the specific properties of the used CFA and color correction algorithms.

As previously reported, in 2004, Kharrazi et al. (2004) assumed that certain traits and patterns should exist between the RGB bands of an image, regardless of the scene content depicted. Therefore, they extracted 21 image features which aimed at capturing cross-channel correlations and energy ratios together with intra-channel average value, pixel neighbor distributions and Wavelet statistics. In 2009, this set of features was enlarged with 6 additional color features by Gloe et al. (2009), which included the dependencies between the average values of the color channels.

Specifically, the authors extended the feature set by computing the norms of the difference between the original and the white-point corrected image.

Image Quality Metrics Features

In 2002, Avcibas et al. (2002) proposed to extract some image quality metrics to infer statistical properties from images. This feature set included measures based on pixel differences, correlations, edges, spectral content, context, and human visual system. The authors proposed to exploit these features for steganalysis applications.

In 2004, Kharrazi et al. (2004) employed a subset of these metrics to deal with the camera model identification task. The authors motivated their choice by noticing that different cameras produce images of different quality, in terms of sharpness, light intensity and color quality. They computed 13 image quality metrics, dividing them into three main categories: those based on pixel differences, those based on correlations and those based on spectral distances. While Kharrazi et al. (2004) averaged these metrics over the three color channels, Çeliktutan et al. (2008) evaluated each color band separately resulting in 40 features.

Wavelet Domain Features

Wavelet-domain features are known to be effective to analyze the information content of images and noise components, enabling to capture interesting insights on their statistical properties for what concerns different resolutions, orientations, and spatial positions (Mallat 1989).

These features have been exploited as well to solve the camera model identification task. For instance, Kharrazi et al. (2004) exploited 9 Wavelet-based color features. Later on, Çeliktutan et al. (2008) added 72 more features by evaluating the four moments of the Wavelet coefficients over multiple sub-bands, and by predicting sub-band coefficients from their neighborhood.

Binary Similarity Features

Like image quality metrics, also binary similarity features were first proposed to tackle steganalysis problems and then adopted in forensics applications. For instance, Avcibas et al. (2005) investigated statistical features extracted from different bit planes of digital images, measuring their correlations and their binary texture characteristics.

Binary similarities were exploited in 2008 by Çeliktutan et al. (2008), which considered the characteristics of the neighborhood bit patterns among the less significant bit planes. The authors investigated histograms of local binary patterns by accounting for occurrences within a single bit plane, across different bit planes and across different color channels. They resulted in a feature vector of 480 elements.

Co-occurrence Based Local Features

Together with binary similarity features and image quality metrics, also rich models, or co-occurrence-based local features (Fridrich and Kodovský 2012), were imported from steganalysis to forensics. Three main steps compose the extraction of co-occurrences of local features (Fridrich and Kodovský 2012): (i) the computation

of image residuals via high-pass filtering operations; (ii) the quantization and truncation of residuals; (iii) the computation of the histogram of co-occurrences.

In this vein, Chen and Stamm (2015) built a rich model of a camera demosaicing algorithm inspired by Fridrich and Kodovský (2012) to perform camera model identification. The authors generated a set of diverse submodels so that every submodel conveyed different aspects of the demosaicing information. Every submodel was built using a particular baseline demosaicing algorithm and computing co-occurrences over a particular geometric structure. By merging all submodels together, the authors obtained a feature vector of 1,372 elements to feed an ensemble classifier.

In 2017, Marra et al. (2017) extracted a feature vector of 625 co-occurrences from each color channel and concatenated the three color bands, resulting in 1,875 elements. The authors investigated a wide range of scenarios that might occur in real-world forensic applications, considering both the closed-set and the open-set camera model identification problems.

Methods with Several Features

In many state-of-the-art works, several different kinds of features were extracted from images instead of just one. The motivation lied in the fact that merging multiple features could better help finding model-related characteristics, thus improving the source identification performance.

For example, (as already reported) Kharrazi et al. (2004) combined color-related features with Wavelet statistics and image quality metrics, ending with 34-element features. Similarly, Gloe (2012) used the same kinds of features but extended the set up to 82 elements. On the other hand, Çeliktutan et al. (2008) exploited image quality metrics, Wavelet statistics, and binary similarity measures, resulting in a total of 592 features.

Later on, in 2018, Sameer and Naskar (2018) investigated the dependency of source classification performance with respect to illumination conditions. To infer model-specific traces, they extracted image quality metrics and Wavelet features from images.

Hand-crafted Features in Open-set Problems

The open-set identification problem started to be investigated in 2012 by Gloe (2012), which proposed two preliminary solutions based on the extraction of 82-element hand-crafted features from images. The first approach was based on a one-class SVM, trained for each available camera model. The second proposal trained a binary SVM in one-versus-one fashion, for all pairs of known camera models. However, the achieved results were unconvincing and revealed practical difficulties to handle unknown models. Nonetheless, this represented a very first attempt to handle unknown source models and highlighted the need of further investigations on the topic.

Later in 2017, Marra et al. (2017) explored the open-set scenario considering two case studies: (i) the limited knowledge situation, in which only images from one camera model were available; (ii) the zero knowledge situation, in which authors had no clue on the model used to collect images. In the former case, the authors aimed at

detecting whether images came from the known model or from an unknown one; in the latter case, the goal was to retrieve a similarity among images shot by the same model.

To solve the first issue, the authors followed a similar approach to Gloe (2012) by training a one-class SVM on the co-occurrence based local features extracted from known images. They solve the second task by looking for K nearest neighbor features in the testing image dataset using a kd-tree-based search algorithm. The performed experiments reported acceptable results but definitively underlined the urgency of new strategies to handle realistic open-set scenarios.

7.3.2 Learned Features

We have recently assisted in the rapid rise of solutions based on learned forensic features which have quickly replaced hand-crafted forensic feature extraction. For instance, considering the camera model identification task, we can directly feed digital images to a deep learning paradigm in order to learn model-related features to associate images with their original source. Actually, we can differentiate between two kinds of learning frameworks:

1. Two-step sequential learning frameworks, which first learn significant features from data and successively learn how to classify data according to the extracted features.
2. End-to-end learning frameworks, which directly classify data by learning some features in the process.

Among deep learning frameworks, Convolutional Neural Networks (CNNs) are now the widespread solution to face several multimedia forensics tasks. In the following, we report state-of-the-art solutions based on learned features which tackle the camera model identification task.

Two-step Sequential Learning Frameworks
One of the first contributions to camera model identification with learned features was proposed in 2017 by Bondi et al. (2017a). The authors proposed a data-driven approach based on CNNs to learn model-specific data features. The proposed methodology was based on two steps: first, training a CNN with image color patches of 64×64 pixels to learn a feature vector of 128 elements per patch; second, training an SVM classifier to distinguish among 18 camera models from the Dresden Image Database (Gloe and Böhme 2010). In their experimental setup, the authors outperformed state-of-the-art methods based on hand-crafted features (Marra et al. 2017; Chen and Stamm 2015).

In the same year, Bayar and Stamm (2017) proposed a CNN-based approach robust to resampling and recompression artifacts. In their paper, the authors proposed an end-to-end learning approach (completely based on CNNs) as well as a two-step learning approach. In the latter scenario, they extracted a set of deep features

from the penultimate CNN layer and fed an Extremely Randomized Trees (ET) classifier, purposely trained for camera model identification. The method was tested on 256 × 256 image patches (retaining only the green color channel) from 26 camera models of the Dresden Image Database (Gloe and Böhme 2010). To mimic realistic scenarios, JPEG compression and resampling were applied to the images. The two-step approach returned slightly better performances than the end-to-end approach.

The same authors also started to investigate open-set camera model identification using deep learning (Bayar and Stamm 2018). They compared different two-step approaches based on diverse classifiers, showing that the ET classifier outperformed the other options.

Meanwhile, Ferreira et al. (2018) proposed a two-step sequential learning approach in which the feature extraction phase was composed of two parallel CNNs, an Inception-ResNet (Szegedy et al. 2017) and an Xception (Chollet 2017) architecture. First, the authors selected 32 patches of 229 × 229 pixels per image according to an interest criterion. Then, they separately trained the two CNNs returning 256-element features each. They merged these features and passed them through a secondary shallow CNN for classification. The proposed method was tested over the IEEE Signal Processing Cup 2018 dataset (Stamm et al. 2018; IEEE Signal Processing Cup 2018 Database 2021). The authors considered data augmentations as well, including JPEG compression, resizing, and gamma correction.

In 2019, Rafi et al. (2019) proposed another two-step pipeline. The authors first trained a 201-layer DenseNet CNN (Huang et al. 2017) with image patches of 256 × 256 pixels. They proposed the DenseNet architecture since dense connections could help in detecting minute statistical features related to the source camera model. Moreover, the authors augmented the training set by applying JPEG compression, gamma correction, random rotation, and empirical mode decomposition (Huang et al. 1998). The trained CNN was used to extract features from image patches of different sizes; these features were concatenated and employed to train a second network for classification. The proposed method was trained over the IEEE Signal Processing Cup 2018 dataset (Stamm et al. 2018; IEEE Signal Processing Cup 2018 Database 2021) but tested on the Dresden Image Database as well (Gloe and Böhme 2010).

Following the preliminary analysis performed by Bayar and Stamm (2018), Júnior et al. (2019) proposed an in-depth study on open-set camera model identification. The authors pursued a two-steps learning approach, comparing several feature extraction algorithms and classifiers. Together with hand-crafted feature extraction methodologies, the CNN-based approach proposed in Bondi et al. (2017a) was investigated. Not much surprisingly, this last approach revealed to be the best effective for feature extraction.

An interesting two-step learning framework was proposed by Guera et al. (2018). The authors explored a CNN-based solution to estimate the reliability of a given image patch, i.e., its likelihood to be used for model identification. Indeed, saturated or dark regions might not contain sufficient information on the source model, thus could be discarded to improve the attribution accuracy. The authors borrowed the CNN architecture from Bondi et al. (2017a) as feature extractor and built a multi-layer

network for reliability estimation. They compared an end-to-end learning approach with two different sequential learning strategies, showing that two-step approaches largely outperformed the end-to-end solution. The authors suggested the lower performance of the end-to-end approach could be due to the limited amount of training data, probably not enough to learn all the CNN parameters in end-to-end fashion. The proposed method was tested on the same Dresden's models employed in Bondi et al. (2017a). By discarding unreliable patches from the process, an improvement on 8% was measured on the identification accuracy.

End-to-end Learning Frameworks

The very first approach to camera model identification with learned features was proposed in 2016 by Tuama et al. (2016). Differently from the above presented works, the authors developed a complete end-to-end learning framework, where feature extraction and classification were performed in a unified framework exploiting a single CNN. The proposed method was quite innovative considering that the literature was based on feature extraction and classification steps. In their work, the authors proposed to extract 256×256 color patches from images, to compute a residual through high-pass filtering, and then to feed a shallow CNN architecture which returned a classification score for each camera model. The method was tested on 26 models from the Dresden Image Database (Gloe and Böhme 2010) and 6 further camera models from a personal collection.

In 2018, Yao et al. (2018) proposed a similar approach to Bondi et al. (2017a), but the authors trained a CNN using an end-to-end framework. 64×64 color patches were extracted from images, fed to a CNN and then passed through majority voting to classify the source camera model. The authors evaluated their methodology on 25 models from the Dresden Image Database (Gloe and Böhme 2010), investigating also the effects of JPEG compression, noise addition, and image re-scaling on a reduced set of 5 devices.

Another end-to-end approach was proposed in 2020 by Rafi et al. (2021), which put particular emphasis on the use of CNNs as pre-processing block prior to a classification network. The pre-processing CNN was composed of a series of "RemNant" blocks, i.e., 3-layer convolutional blocks followed by batch normalization. The output to the pre-processing step contained residual information about the input image and preserved model-related traces from a wide range of spatial frequencies. The whole network, named "RemNet", included the pre-processing CNN and a shallow CNN for classification. The authors tested their methodology on 64×64 image patches, considering 18 models from the Dresden Image Database (Gloe and Böhme 2010) and the IEEE Signal Processing Cup 2018 dataset (Stamm et al. 2018; IEEE Signal Processing Cup 2018 Database 2021).

A very straightforward end-to-end learning approach was deployed in Mandelli et al. (2020), where the authors did not aim at exploring novel strategies to boost the achieved identification accuracy, but investigated how JPEG compression artifacts can affect the results. The authors compared four state-of-the-art CNN architectures in different training and testing scenarios related to JPEG compression. Experiments were computed on 512×512 image patches from 28 camera models of the Vision

Image Dataset (Shullani et al. 2017). Results showed that being careful to the JPEG grid alignment and to the compression quality factor of training/testing images is of paramount importance, independently on the chosen network architecture.

Learned Features in Open-set Problems

The majority of the methods presented in previous sections tackled closed-set camera model identification, leaving open-set challenges to future investigations. Indeed, the open-set literature still counts very few works and lacks proper algorithm comparisons with respect to closed-set investigations. Among learned feature methodologies, only Bayar and Stamm (2018) and Júnior et al. (2019) tackled open-set classification. Here, we illustrate the proposed methodologies in greater more detail.

In 2018, Bayar and Stamm (2018) proposed two different learning frameworks for open-set model identification. The common paradigm behind the approaches was two-step sequential learning, where the feature extraction block included all the layers of a CNN that precede the classification layer. Following their prior work (Bayar and Stamm 2017), the authors trained a CNN for camera model identification and selected the resulting deep features associated with the second-to-last layer to train a classifier. The authors investigated 4 classifiers: (i) one fully connected layer followed by the softmax function, which is the standard choice in end-to-end learning frameworks; (ii) an ET classifier; (iii) an SVM-based classifier; (iv) a classifier based on the cosine similarity distance.

In the first approach, the authors trained the classifier in closed-set fashion over the set of known camera models. In the testing phase, they thresholded the maximum score returned by the classifier. If that score exceeded a predefined threshold, the image was associated with a known camera model, otherwise it was declared to be of unknown provenance.

In the second approach, known camera models were divided in two disjoint sets: "known-known" models and "known-unknown" models. A closed-set classifier was trained over the reduced set of "known-known" models, while a binary classifier was trained to distinguish "known-known" models from "known-unknown" models. In the testing phase, if the binary classifier returned the "known-unknown" class, the image was linked to an unknown source model, otherwise, the image was associated with a known model.

In any case, when the image was determined to belong to a known model, the authors estimated the actual source model as in a standard closed-set framework.

Experimental results were evaluated on 256×256 grayscale image patches selected from 10 models of the Dresden Image Database (Gloe and Böhme 2010) and other 15 personal devices. The authors showed that the softmax-based classifier performed worst among the 4 proposals. The ET classifier achieved the best performance for both approaches both in identifying the source model in closed-set fashion and in known-vs-unknown model detection.

An in-depth study on the open-set problem has been pursued by Júnior et al. (2019). The authors thoroughly investigated this issue by comparing 5 feature extraction methodologies, 3 training protocols and 12 open-set classifiers. Among feature extractors, they selected hand-crafted based frameworks (Chen and Stamm 2015;

Marra et al. 2017) together with learned features (Bondi et al. 2017a). As for training protocols, they considered the two strategies presented by Bayar and Stamm (2018) and one additional strategy which included more "known-unknown" camera models. Several open-set classifiers were explored, ranging from different SVM-based approaches to one classifier proposed in the past by the same authors and to those previously investigated by Bayar and Stamm (2018).

The training dataset included 18 camera models from the Dresden Image Database (Gloe and Böhme 2010), i.e., the same models used by Bondi et al. (2017a) in closed-set scenario, as known models, and the remaining Dresden's models as "known-unknown" to be used for the additional proposed training protocol. Furthermore, 35 models from the Image Source Attribution UniCamp Dataset (Image Source Attribution UniCamp Dataset 2021) and 250 models from the Flickr image hosting service were used to simulate unknown models.

As performed by Bondi et al. (2017a), the authors extracted 32 non-overlapping 64×64 patches from images. Results demonstrated the highest effectiveness of CNN-based learned features over hand-crafted features. Many open-set classifiers returned high performances and, as reported in Bayar and Stamm (2018), ET classifier was one of the best performing. Interestingly, the authors noticed that the third additional training protocol that required more "known-unknown" models was outperformed by a simpler solution, corresponding to the second training approach proposed by Bayar and Stamm (2018).

Learned Forensics Similarity

An interesting research work which parallels camera model identification was proposed in 2018 by Mayer and Stamm (2018). In their paper, the authors proposed a two-step learning framework that did not aim at identifying the source camera model of an image, but aimed at determining whether two images (or image patches) were shot by the same camera model.

The main novelty did not lie in the feature extraction step, which was CNN-based as many other contemporaneous works, but in the second step developing a learned forensics similarity measure between the two compared patches. This was implemented by training a multi-layer neural network that mapped the features from patches into a single similarity score. If the score overcame a predefined threshold, the two patches were said to come from the same model.

The authors did not limit the comparison over known camera models, but also demonstrated the effectiveness of the method on unknown models. In particular, they trained the two networks over disjoint camera model sets, exploiting grayscale image patches with size 256×256, selected from 20 models of the Dresden Image Database (Gloe and Böhme 2010) and 45 further models. Results showed that authors could accurately detect a model similarity even if both the two images came from unknown sources.

7.4 Datasets and Benchmarks

This section provides a template with the fundamental characteristics an image dataset should have in order to explore camera model identification. Then, it shows an overview of available datasets in the literature. It also explains how to properly deal with these datasets for a fair evaluation.

7.4.1 Template Dataset

The task of camera model identification assumes that we are not interested in retrieving information on the specific device used for capturing a photograph neither on the scene content depicted. Given these premises, we can define some "good" features a template dataset should include (Kirchner and Gloe 2015):

- Images of similar scenes taken with different camera models.
- Multiple devices per camera model.

Collecting several images with similar scenes shot by different models avoids any possible bias on the results due to the depicted scene content. Moreover, capturing images from multiple devices of the same camera model accounts for realistic scenarios. Indeed, we should expect to investigate query images shot by devices never seen at algorithm development phase, even though their camera models were known.

A common danger that needs to be avoided is to confuse model identification with device identification. Hence, as we want to solve the source identification task at model-based granularity, we must not confuse device-related traces with model-related ones. In other words, query images coming from an unknown device of a known model should not be confused as coming from an unknown source. In this vein, a dataset should consist of multiple devices from the same model. This enables to evaluate a feature set or a classifier for its ability to detect models independently from individual devices. Including more devices per model in the image dataset allows to check for this requirement and helps keeping the problem at bay.

7.4.2 State-of-the-art Datasets

The following datasets are frequently used or have been specifically designed for camera model identification:

- Dresden Image Database (Gloe and Böhme 2010).
- Vision Image Dataset (Shullani et al. 2017).
- Forchheim Image Database (Hadwiger and Riess 2020).
- Dataset for Camera Identification on HDR images (Shaya et al. 2018).
- Raise Image Dataset (Nguyen et al. 2015).

Table 7.1 Main datasets' characteristics

Dataset	No. of models	No. of images	Image formats
Dresden (Gloe and Böhme 2010)	26	18,456	JPEG, Uncompressed-Raw
Vision (Shullani et al. 2017)	28	34,427	JPEG
Forchheim (Hadwiger and Riess 2020)	25	23,106	JPEG
HDR (Shaya et al. 2018)	21	5,415	JPEG
Raise (Nguyen et al. 2015)	3	8,156	Uncompressed-Raw
Socrates (Galdi et al. 2019)	60	9,700	JPEG
IEEE Cup (Stamm et al. 2018; IEEE Signal Processing Cup 2018 Database 2021)	10	2,750	JPEG

- Socrates Dataset (Galdi et al. 2019);
- the dataset for the IEEE Signal Processing Cup 2018: Forensic Camera Model Identification Challenge (Stamm et al. 2018; IEEE Signal Processing Cup 2018 Database 2021).

To highlight the differences among the datasets in terms of camera models and images involved, Table 7.1 summarizes the main datasets' characteristics. In the following, we illustrate in detail the main features of each dataset.

Dresden Image Database

The Dresden Image Database (Gloe and Böhme 2010) has been designed with the specific purpose of investigating the camera model identification problem. This is a publicly available dataset, including approximately 17,000 full-resolution natural images stored in the JPEG format with the highest available JPEG quality and 1,500 uncompressed raw images. The image acquisition process has been carefully designed in order to provide image forensics analysts a dataset which satisfies the two properties reported in Sect. 7.4.1. Images were captured under controlled conditions from 26 different camera models considering up to 5 different instances per model. For each motif, at least 3 scenes were captured by varying the focal length.

Thanks to the careful design process and to the considerable amount of images included, the Dresden Image Database has become over the years one of the most used image datasets for tackling image forensics investigations. The use of this dataset as a benchmark has favored the spreading of research works on the topic, easing the comparison between different methodologies and their reproducibility. Focusing on the research on the camera model identification task, the Dresden Image Database has been used several times by state-of-the-art works (Bondi et al. 2017b; Marra et al. 2017; Tuama et al. 2016).

Vision Image Dataset

The Vision dataset (Shullani et al. 2017) is a recent image and video dataset, purposely designed for multimedia forensics investigations. Specifically, Vision dataset

has been designed to follow the trend on image and video acquisition and social sharing. In the last few years, photo-amateurs have rapidly transitioned to hand-held devices as preferred mean to capture images and videos. Then, the acquired content is typically shared on social media platforms like WhatsApp or Facebook. In this vein, Vision dataset collects almost 12,000 native images captured by 35 modern smartphones/tablets, including also their related social media version.

The Vision dataset well satisfies the first requirement presented in Sect. 7.4.1 about capturing images of similar scenes taken with different camera models. Moreover, it represents a substantial improvement with respect to Dresden Image Database, since it collects images from modern devices and social media platforms. However, a minor limitation regards the second requirement provided in Sect. 7.4.1: there are only few camera models with two or more instances. Indeed, over the 35 available modern devices, we have 28 different camera models.

In the literature, Vision dataset has been used many times for investigations on the camera model identification problem. Among them, we can cite (Cozzolino and Verdoliva 2020; Mandelli et al. 2020).

Forchheim Image Database

The Forchheim Image Database (Hadwiger and Riess 2020) consists of more than 23,000 images of 143 scenes shot by 27 smartphone cameras of 25 models and 9 brands. It has been proposed to cleanly separate scene content and forensic traces, and to support realistic post-processing like social media recompression. Indeed, six different qualities are provided per image, collecting different copies of the same original image passed through social networks.

Dataset for Camera Identification on HDR Images

The proposed dataset collects standard dynamic range and HDR images captured in different conditions, including various capturing motions, scenes, and devices (Shaya et al. 2018). It has been proposed to investigate the source identification problem on HDR images, which usually introduce some difficulties due to their complexity and wider dynamic range. 23 mobile devices were used for capturing 5,415 images in different scenarios.

Raise Image Dataset

The Raise Image Dataset (Nguyen et al. 2015) concerns 8,156 high-resolution raw images, depicting various subjects and scenarios. 3 different camera of diverse models were employed.

Socrates Dataset

The Socrates Dataset (Galdi et al. 2019) has been built to investigate the source camera identification problem on images and videos coming from smartphones. Images and videos have been collected directly by smartphone owners, ending up with about 9,700 images and 1000 videos captured with 104 different smartphones of 15 different makes and about 60 different models.

Dataset for the IEEE Signal Processing Cup 2018: Forensic Camera Model Identification Challenge
The IEEE Signal Processing Cup (SP Cup) is a student competition in which undergraduate students form teams to work on real-life challenges. In 2018, the camera model identification goal was selected as the topic for the SP Cup (Stamm et al. 2018). Participants were provided with a dataset consisting of JPEG images from 10 different camera models (including point-and-shoot cameras, cell phone cameras, and digital single-lens reflex cameras), with 200 images captured using each camera model. In addition, also post-processed operations (e.g., JPEG recompression, cropping, contrast enhancement) were applied to the images. The complete dataset can be downloaded from IEEE DataPort (IEEE Signal Processing Cup 2018 Database 2021).

7.4.3 Benchmark Protocol

The benchmark protocol commonly followed by forensics researchers is to divide the available images into three disjoint image datasets: the training dataset $\mathcal{I}_t$, the validation dataset $\mathcal{I}_v$ and the evaluation dataset $\mathcal{I}_e$. This split ensures that images seen during the training process (i.e., images belonging to either $\mathcal{I}_t$ or $\mathcal{I}_v$) are never used in testing phase, thus they do not introduce any bias in the results.

Moreover, it is reasonable to assume that query images under analysis might not be acquired with the same devices seen in training phase. Therefore, what is usually done is to pick a selection of devices to be used for the training process, namely the training device set $\mathcal{D}_t$. Then, the proposed method can be evaluated over the total amount of available devices. A realistic and challenging scenario is to include only one device per known camera model in the training set $\mathcal{D}_t$.

7.5 Case Studies

This section is devoted to numerical analysis of some selected methods in order to showcase the capabilities of modern camera model identification algorithms. Specifically, we consider a set of baseline CNNs and co-occurrences of image residuals (Fridrich and Kodovský 2012) analyzing both closed-set and open-set scenarios. The impact of JPEG compression is also investigated.

7.5.1 Experimental Setup

To perform our experiments, we select natural JPEG compressed images from the Dresden Image Database (Gloe and Böhme 2010) and the Vision Image Dataset

(Shullani et al. 2017). Regarding the Dresden Image Database, we consider images from "Nikon_D70" and "Nikon_D70s" camera models as coming from the same model, as the differences between these two models are negligible due to a minor version update (Gloe and Böhme 2010; Kirchner and Gloe 2015). We pick the same number of images per device, and use as reference the device with the lowest image cardinality. We end up with almost 15,000 images from 54 different camera models, including 108 diverse devices.

We work in a patch-wise fashion, extracting N patches with size $P \times P$ pixels from each image. We investigate four different networks. Two networks are selected from the recently proposed EfficientNet family of models (Tan and Le 2019), which achieves very good results both in computer vision and multimedia forensics tasks. Specifically, we select EfficientNetB0 and EfficientNetB4 models. The other networks are known in literature as ResNet50 (He et al. 2016) and XceptionNet (Chollet 2017). Following a common procedure in CNN training, we initialize the network weights using those trained on ImageNet database (Deng et al. 2009). All CNNs are trained using cross-entropy loss and Adam optimizer with default parameters. The learning rate is initialized to 0.001 and is decreased by a factor 10 whenever the validation loss does not improve for 10 epochs. The minimum accepted learning rate is set to 10^{-8}. We train the networks for at most 500 epochs, and training is stopped if loss does not decrease for more than 50 epochs. The model providing the best validation loss is selected.

At test time, classification scores are always fed to the softmax function. In the closed-set scenario, we assign the query image to the camera model associated with the highest softmax score. We use the average accuracy of correct predictions as evaluation metrics. In the open-set scenario, we evaluate results as a function of the detection accuracy of "known" versus "unknown" models. Furthermore, we also provide the average accuracy of correct predictions over the set of known camera models. In other words, given that a query image was taken with one camera model belonging to the known class, we evaluate the classification accuracy as in a closed-set problem reduced to the "known" categories.

Concerning the dataset split policy, we always keep 80% of the images for training phase, further divided in 85%/15% for training set $\mathcal{I}_t$ and validation set $\mathcal{I}_v$, respectively. The remaining 20% of the images are used in the evaluation set $\mathcal{I}_e$. All tests have been run on a workstation equipped with one Intel® Xeon Gold 6246 (48 Cores @3.30 GHz), RAM 252 GB, one TITAN RTX (4608 CUDA Cores @1350 MHz), 24 GB, running Ubuntu 18.04.2. We resort to Albumentation (Buslaev et al. 2020) as data augmentation library for applying JPEG compression to images, and we use Pytorch (Paszke et al. 2019) as Deep Learning framework.

7.5.2 Comparison of Closed-Set Methods

In this section, we compare closed-set camera model identification methodologies. We do not consider model-based approaches, since data-driven methodologies have

extensively outperformed them in the last few years (Bondi et al. 2017a; Marra et al. 2017). In particular, we compare 4 different CNN architectures with a well-known state-of-the-art method based on hand-crafted feature extraction. Specifically, we extracted the co-occurrences of image residuals as suggested in Fridrich and Kodovský (2012). Indeed, it has been shown that exploiting these local features provides valuable insights on the camera model identification task (Marra et al. 2017).

Since data-driven methods need to be trained on a specific set of images (or image-patches), we consider different scenarios in which the characteristics of the training dataset change.

Training with Variable Patch-sizes, Proportional Number of Patches per Image In this setup, we work with images from both the Dresden Image Database and the Vision Image Dataset. We randomly extract N patches per image with size $P \times P$ pixels. We consider patch-sizes $P \in \{256, 512, 1024\}$. As first experiment, we would like to maintain a constant number of image pixels seen in training phase. In doing so, we can compare the methods' performance under the same number of input pixels. Hence, the smaller the patch-size, the more image patches are provided. In case of $P = 256$, we randomly extract $N = 40$ patches per image; for $P = 512$, we randomly extract $N = 10$ patches per image; when $P = 1024$, we randomly extract $N = 3$ patches per image. The number of input image pixels remains constant in the first two situations, while the last scenario includes few pixels more.

Co-occurrence Based Local Features. We extract co-occurrences features of 625 elements (Fridrich and Kodovský 2012) from each analyzed patch independent of the input patch-size P. More precisely, we extract features based on the third order filter named "s3-spam14hv", as suggested in Marra et al. (2017). We apply this filter to the luminance component of the input patches.

Then, to associate the co-occurrences features with one camera model, we train a 54-classes classifier composed of a shallow neural network. The classifier is defined as

- a fully connected layer with 625 input channels, i.e., the dimension of co-occurrences, and 256 output channels;
- a dropout layer with 0.5 as dropout ratio;
- a fully connected layer with 256 input channels and 54 output channels.

We train this elementary network using Adam optimization with initial learning rate of 0.1, following the same paradigm described in Sect. 7.5.1. Results on evaluation images are shown in Table 7.2.

Table 7.2 Accuracy of camera model identification in closed-set scenario using co-occurrences features

Patch-size P	256	512	1024
Accuracy (%)	68.77	78.81	82.77

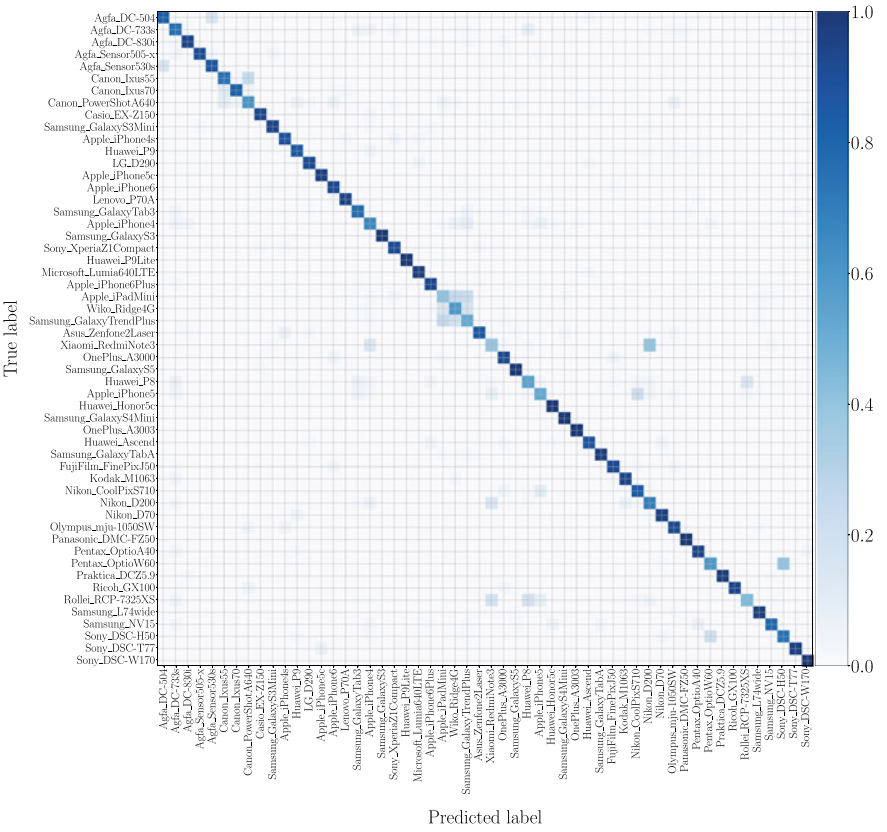


Fig. 7.3 Confusion matrix achieved in closed-set classification by co-occurrences extraction, $P = 1024$

Notice that the larger the patch-size, the higher the achieved accuracy. Even though the number of image pixels seen by the classifier in training phase is essentially constant, the features extracted from the patches become more significant as the pixel region fed to the co-occurrence extraction grows. Figure 7.3 shows the achieved confusion matrix for $P = 1024$.

CNNs. CNN results are shown in Table 7.3. Differently from co-occurrences, CNNs seem to be considerably less dependent on the specific patch-size, as long as the number of image pixels fed to the network remains the same. All the network models perform slightly worse on larger patches (i.e., $P = 1024$). This lower performance may originate from a higher difficulty of the networks in converging during the training process. Larger patches inevitably require reduced batch-size during training. Less samples in the batch means less results' average in one training epoch and reduced CNN capabilities in converging to the best parameters.

Table 7.3 Accuracy of closed-set camera model identification using 4 different CNN architectures as a function of the patch-size. The number of extracted patches per image varies such that the overall image pixels seen by CNNs is almost constant

Patch-size P	256	512	1024
EfficientNetB0 (%)	94.68	94.60	94.28
EfficientNetB4 (%)	94.29	94.57	93.20
ResNet50 (%)	93.71	92.48	91.70
XceptionNet (%)	93.74	94.21	91.23

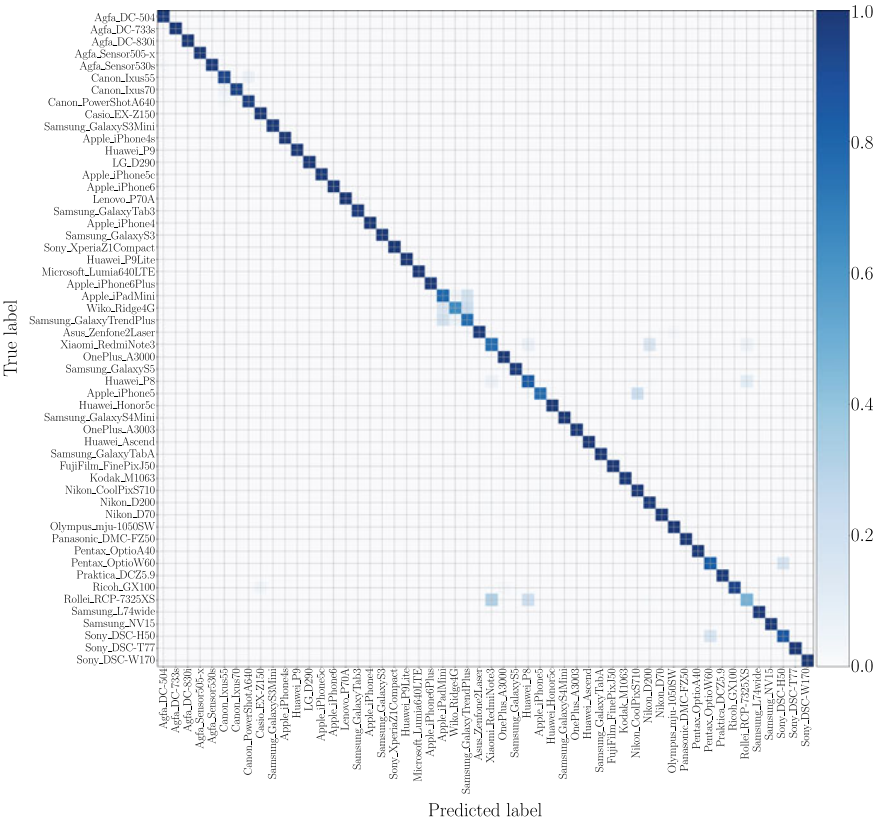


Fig. 7.4 Confusion matrix achieved in closed-set classification by EfficientNetB0, $P = 512$, $N = 10$

Overall, results on CNNs strongly outperform co-occurrences. Fixing a patch-size, all the CNNs return very similar results. The EfficientNet family of models slightly outperforms the other CNN architectures, as it has been shown several times in the literature (Tan and Le 2019).

To provide an example of the results, Fig. 7.4 shows the achieved confusion matrix by EfficientNetB0 architecture for $P = 512$, $N = 10$.

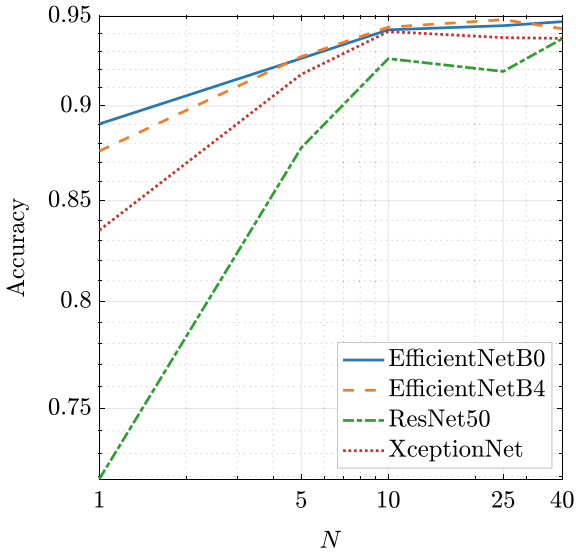


Fig. 7.5 Closed-set classification accuracy achieved by CNNs as a function of the number of extracted patches per image, patch-size $P = 256$

Training with Fixed Patch-sizes, Variable Number of Patches per Image
In this setup, we work again with images from both the Dresden Image Database and the Vision Image Dataset. Differently from the previous setup, we do not evaluate results of co-occurrences since they are significantly outperformed by the CNN-based framework. Instead of exploring diverse patch-sizes, we now fix the patch-size $P = 256$. We aim at investigating how CNN performance changes by varying the number of extracted patches per image. N can vary among $\{1, 5, 10, 25, 40\}$.

Figure 7.5 depicts the closed-set classification accuracy results. For small values of N (i.e., $N < 10$), all the 4 networks enhance their performance as the number of extracted patches increases. When N further increases, CNNs achieve a performance plateau and accuracy does not change too much across $N \in \{10, 25, 40\}$.

Training with Variable Patch-size, Fixed Number of Patches per Image
We now fix the number of extracted patches per image to $N = 10$. Then, we vary the patch-size P among $\{64, 128, 256, 512\}$. The goal is to explore whether results change as a function of the patch-size.

Figure 7.6 depicts the results. All the CNNs achieve accuracies beyond 90% when patch-size is larger or equal to 256. When the data fed to the network are too small, CNNs are not able to estimate well the classification parameters.

Investigations on the Influence of JPEG Grid Alignment
When editing a photograph or uploading a picture over a social media platform, it may be cropped with respect to its original size. After the cropping, a JPEG compression is usually applied to the image. These operations may de-synchronize the 8×8 characteristic pixel grid of the original JPEG compression. Usually, a new 8×8 grid

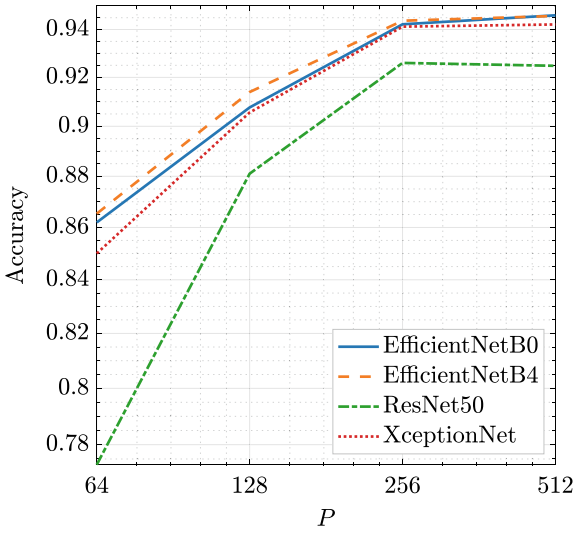


Fig. 7.6 Closed-set classification accuracy achieved by CNNs as a function of the patch-size P, the number of extracted patches per image is fixed to $N = 10$

non-aligned with the original one is generated. When training a CNN to solve an image classification problem, the presence of JPEG grid misalignment on images can strongly impact the performance. In this section, we summarize the experiments performed in Mandelli et al. (2020) to investigate on the influence of JPEG grid alignment on the closed-set camera model identification task.

During training and evaluation steps, we investigate two scenarios:

1. working with JPEG compressed images whose JPEG grid is aligned to the 8×8 pixel grid starting from upper-left corner;
2. working with JPEG compressed images whose JPEG grid starts in random pixel position.

To simulate these scenarios, we first compress all the images with the maximum JPEG Quality Factor (QF), i.e., QF $= 100$. This process generates images with JPEG lattice and does not impair the image visual quality. The first scenario includes images cropped such that the extracted image patches are always aligned to the 8×8 pixel grid. In the latter scenario, we extract patches in random positions. Differently from previous sections, here we are considering a reduced image dataset, the same used in Mandelli et al. (2020) from which we are reporting the results. Images are selected only from the Vision dataset and $N = 10$ squared patches of 512×512 pixels are extracted from the images.

Figure 7.7 shows the closed-set classification accuracy for all CNNs as a function of the considered scenarios. In particular, Fig. 7.7a depicts results in case we train and test on randomly cropped patches; Fig. 7.7b shows what happens when training on JPEG-aligned images and testing on randomly cropped images; Fig. 7.7c explores training on randomly cropped images and testing on JPEG-aligned images; Fig. 7.7d draws results in case we train and test on JPEG-aligned images. It is worth

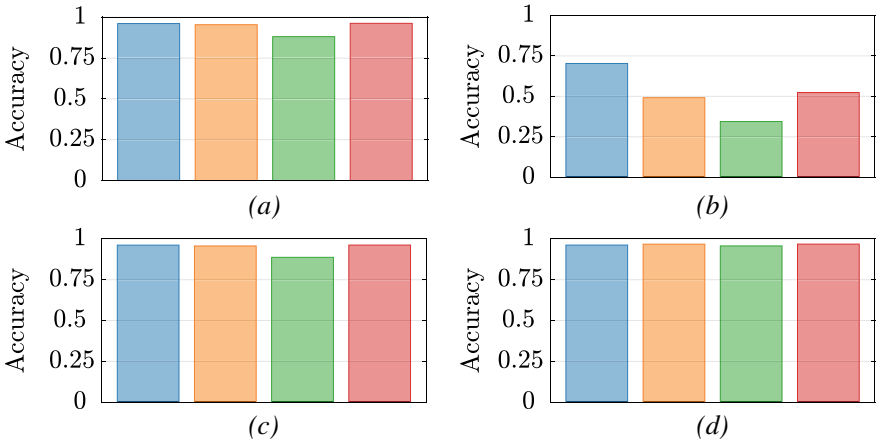


Fig. 7.7 Closed-set classification accuracy achieved by CNNs as a function of JPEG grid alignment in training and/or evaluation phases. The bars represent, respectively, from left to right: EfficientNetB0; EfficientNetB4; ResNet50; XceptionNet. In **a**, we train and test on randomly cropped patches; in **b**, we train on JPEG-aligned and test on randomly cropped patches; in **c**, we train on randomly cropped and test on JPEG-aligned patches; in **d**, we train and test on JPEG-aligned patches

noticing that being careful to the JPEG grid alignment is paramount for achieving good accuracy. When training a detector on JPEG-aligned patches and testing on JPEG-misaligned ones, results drop consistently as shown in Fig. 7.7b.

Investigations on the Influence of JPEG Quality Factor

In this section, we aim at investigating how the QF of JPEG compression affects CNN performance in closed-set camera model identification. As done in the previous section, we illustrate the experimental setup provided in Mandelli et al. (2020), where the effect of JPEG compression quality is studied for images belonging to the Vision dataset. We JPEG-compress the images with diverse QFs, namely QF $\in \{50, 60, 70, 80, 90, 99\}$. Then, we extract $N = 10$ squared patches of 512×512 pixels from images. Following previous considerations, we randomly extract the patches both for the training and evaluation datasets.

To explore the influence of JPEG QF on CNN results, we train the network in two ways:

1. we use only images of the original Vision Image dataset;
2. we perform some training data augmentation. Half of the training images are taken from the original Vision Image dataset; the remaining part is compressed with a JPEG QF picked from the reported list.

Notice that the second scenario assumes some knowledge on the JPEG QF of the evaluation images, and can thus improve the achieved classification results.

Closed-set classification accuracy achieved by the CNNs is reported in Fig. 7.8. In particular, we show results as a function of the QF of the evaluation images.

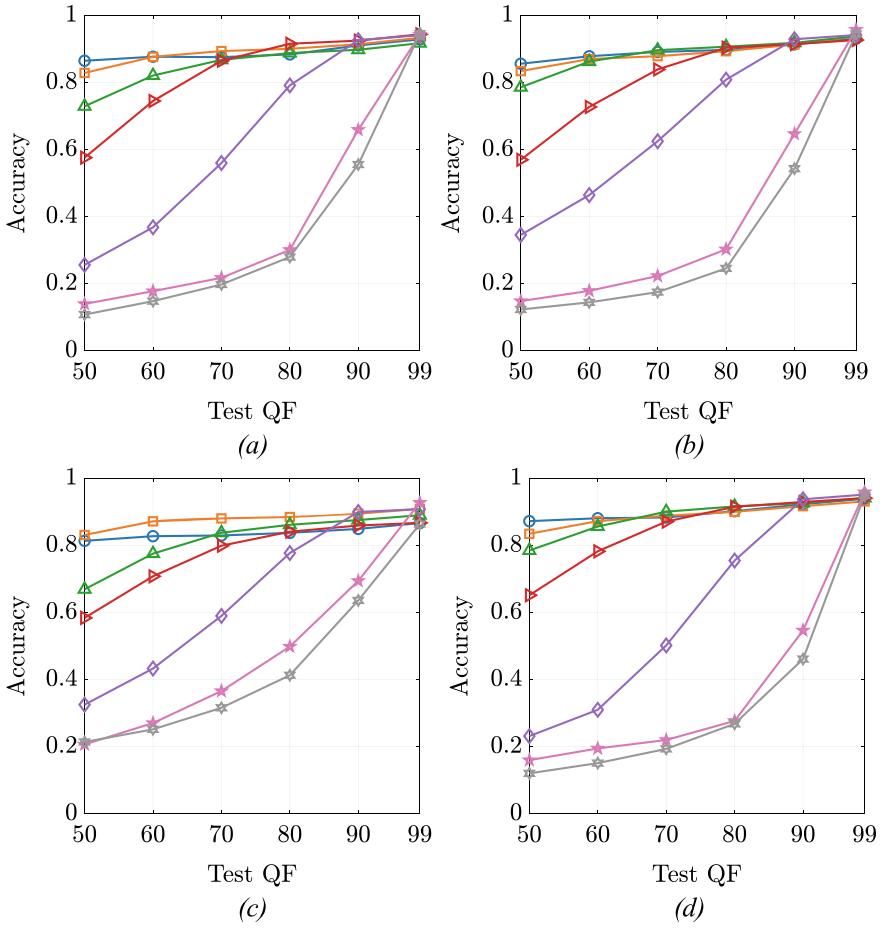


Fig. 7.8 Accuracy as a function of test QF for **a** EfficientNetB0, **b** EfficientNetB4, **c** ResNet50, **d** XceptionNet. The curves are drawn in accordance to the following legend: ● train on QF = 50; ■ train on QF = 60; ▲ train on QF = 70; ▶ train on QF = 80; ♦ train on QF = 90; ★ train on QF = 99; ★ No augmentation

The gray curve represents the first training scenario, i.e., in absence of training data augmentation. Note that this setup always draws the worst results. Also training on data augmented with QF = 99 almost corresponds to absence of augmentation and achieves acceptable results only when the test QF matches the training one. On the contrary, training on data augmented with low JPEG QFs (i.e., QF $\in \{50, 60\}$) returns acceptable results for all the possible test QFs. For instance, by training on data compressed with QF = 50, evaluation accuracy can improve from 0.2 to more than 0.85.

7.5.3 Comparison of Open-Set Methods

In the open-set scenario some camera models are unknown, so we cannot train CNNs on the complete device set as information can be extracted only from the known devices. Let us represent the original camera model set with $\mathcal{T}$: known models are denoted as $\mathcal{T}_k$, unknown models are denoted as $\mathcal{T}_u$.

To perform open-set classification, two main training strategies can be pursued (Bayar and Stamm 2018; Júnior et al. 2019):

1. Consider all the available set $\mathcal{T}_k$ as "known" camera models. Train one closed-set classifier over $\mathcal{T}_k$ camera models.
2. Divide the set of $\mathcal{T}_k$ camera models into two disjoint sets: the "known-known" set $\mathcal{T}_{k_k}$, and the "known-unknown" set $\mathcal{T}_{k_u}$. Train one binary classifier to identify $\mathcal{T}_{k_k}$ camera models from $\mathcal{T}_{k_u}$ camera models.

Results are evaluated accordingly to two metrics (Bayar and Stamm 2018):

1. the accuracy of detection between known camera models and unknown camera models;
2. the accuracy of closed-set classification among the set of known camera models. This metric is validated only for images belonging to camera models in set $\mathcal{T}_k$.

Our experimental setup considers only EfficientNetB0 as network architecture, since EfficientNet family of models (both EfficientNetB0 and EfficientNetB4) always reports higher or comparable accuracies with respect to other architectures. We choose EfficientNetB0 which is lighter than EfficientNetB4 to reduce training and testing time. We exploit images from both the Dresden Image Database and the Vision Image Dataset. Among the pool of 54 camera models, we randomly extract 36 camera models for the "known" set $\mathcal{T}_k$ and leave the remaining 18 to the "unknown" set $\mathcal{T}_u$. Following considerations, we randomly extract $N = 10$ squared patches per image with patch-size $P = 512$.

Training One Closed-set Classifier over "Known" Camera Models
We train one closed-set classifier over models belonging to $\mathcal{T}_k$ set by following the same training protocol previously seen for closed-set camera model identification. In testing phase, we proceed with two consequential steps:

1. detect if the query image is taken by a "known" camera model or an "unknown" model;
2. if the query image comes from a "known" camera model, identify the source model.

Regarding the first step, we follow the approach suggested in Bayar and Stamm (2018). Given a query image, if the maximum score returned by the classifier exceeds a predefined threshold, we assign the image to the category of "known" camera models. Otherwise, the image is associated with an "unknown" model. Following this procedure, the closed-set classification accuracy across models in $\mathcal{T}_k$ is 95%. Figure 7.9 shows the confusion matrix achieved by closed-set classification.

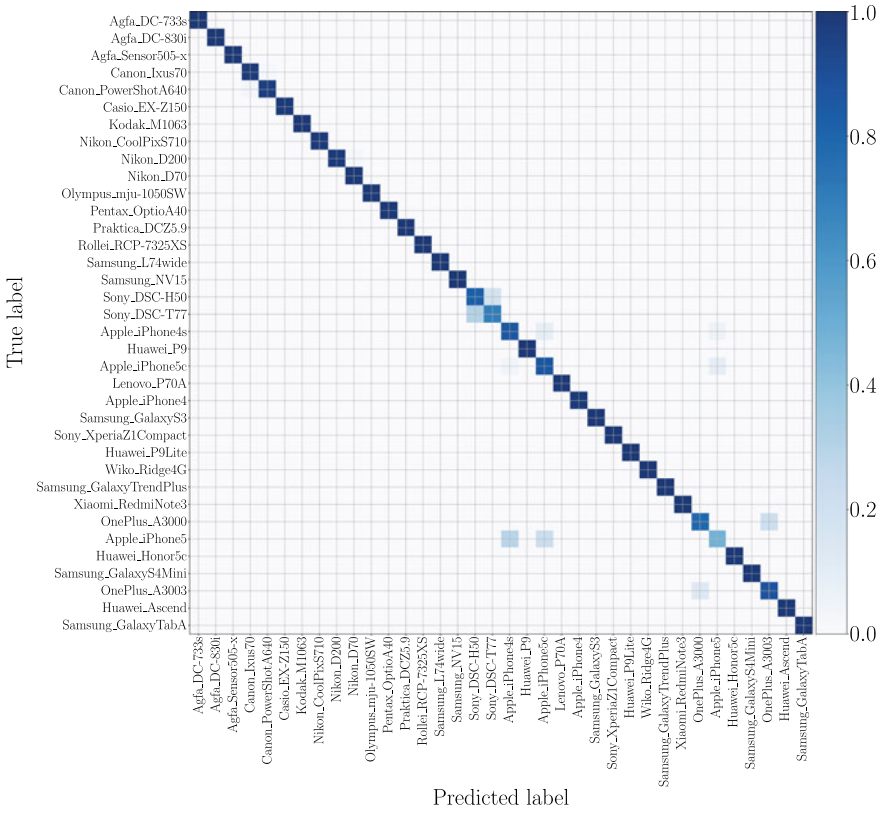


Fig. 7.9 Confusion matrix achieved in closed-set classification by training one classifier (in closed-set fashion) over the 36 camera models in $\mathcal{T}_k$. The average closed-set accuracy is 95%

The detection accuracy varies with the threshold used to classify "known" versus "unknown" camera models. Figure 7.10 depicts the ROC curve achieved by the proposed method. Specifically, we see the behavior of True Positive Rate (TPR) as a function of the False Positive Rate (FPR). The maximum accuracy value is 86.19%.

Divide the Set of Known Camera Models into "Known-known" and "Known-unknown"

In this experiment, we divide the set of known camera models into two disjoint sets: the set of "known-known" models $\mathcal{T}_{k_k}$ and the set of "known-unknown" models $\mathcal{T}_{k_u}$. Then, we train a binary classifier which discriminates between the two classes. In testing phase, all the images assigned to the "known-unknown" class will be classified as unknown. Notice that by training one binary classifier we are able to recognize as "known" category only images taken from camera models in $\mathcal{T}_{k_k}$. Therefore, even if camera models belonging to $\mathcal{T}_{k_u}$ are known, their images will be classified as unknown in testing phase.

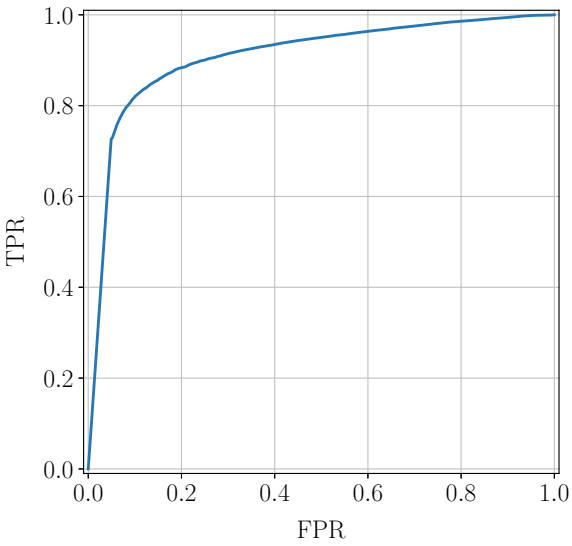


Fig. 7.10 ROC curve achieved by training one classifier (in closed-set fashion) over camera models in $\mathcal{T}_k$

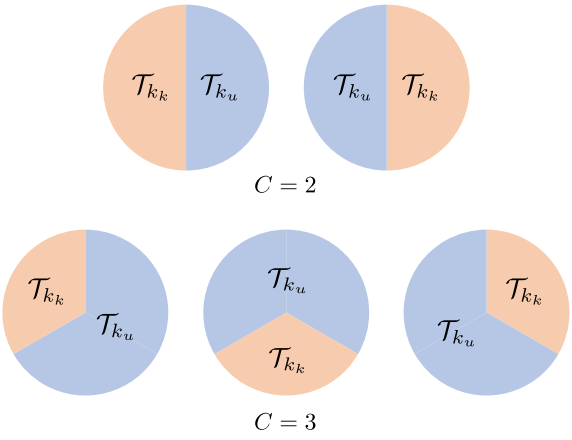


Fig. 7.11 Training dataset split policy applied to set $\mathcal{T}_k$ in case $C = 2$ and $C = 3$

To overcome this limitation, we explore the possibility of training a binary classifier in different setups. In practice, we divide the known camera models' set $\mathcal{T}_k$ into C disjoint subsets containing images of $|\mathcal{T}_k|/C$ models, where $|\cdot|$ is the cardinality of the set. We define every disjoint set as $\mathcal{T}_{k_c}$, $c \in [1, C]$. Then, for each disjoint set $\mathcal{T}_{k_c}$, we train one binary classifier telling known models and unknown models apart. Set $\mathcal{T}_{k_c}$ represents the "known-known" class, while the remaining sets are joined and form the "known-unknown" class. Figure 7.11 depicts the split policy applied to $\mathcal{T}_k$ for $C = 2$ and $C = 3$. For instance, when $C = 3$, we end up with 3 different training setups: the first considers $\mathcal{T}_{k_1}$ as known class and models in $\{\mathcal{T}_{k_2}, \mathcal{T}_{k_3}\}$ as unknowns;

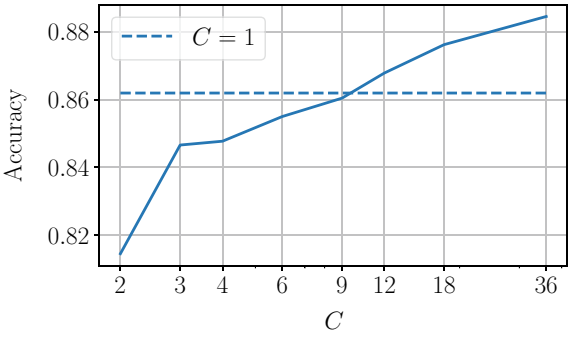


Fig. 7.12 Maximum detection accuracy achieved in the ROC curve by training C classifiers over camera models in $\mathcal{T}_k$

the second considers $\mathcal{T}_{k_2}$ as known class and models in $\{\mathcal{T}_{k_1}, \mathcal{T}_{k_3}\}$ as unknowns; the third considers $\mathcal{T}_{k_3}$ as known class and models in $\{\mathcal{T}_{k_1}, \mathcal{T}_{k_2}\}$ as unknowns.

In testing phase, we assign a query image to the "known-known" class only if at least one classifier returned a score associated with the "known-known" class which is sufficiently confident. In practice, we threshold the maximum score associated to the "known-known" class returned by the C classifiers. Whenever the maximum score overcomes the threshold, we assign the query image to a known camera model; otherwise we associate it with an unknown model.

Notice that the number of classifiers can vary from $C = 2$ to $C = 36$ (i.e., the total number of available known camera models). In our experiments, we consider C equal to all the possible divisors of 36, i.e., $C \in \{2, 3, 4, 6, 9, 12, 18, 36\}$. In order to work with balanced training datasets, we always fix the number of images for each class to be equal to the image cardinality of the known set $\mathcal{T}_{k_k}$.

The maximum detection accuracy achieved as a function of the number of classifiers used is shown in Fig. 7.12. The case $C = 1$ corresponds to training only one classifier in closed-set fashion. Notice that when the number of classifiers starts to increase (i.e., when $C > 9$), the proposed methodology can outperform the closed-set classifier.

We also evaluate the accuracy achieved in closed classification for images belonging to known models in $\mathcal{T}_k$. For each disjoint set $\mathcal{T}_{k_c}$, we can train a $|\mathcal{T}_k|/C$-class classifier in closed-set fashion. If one of the C binary classifiers assigns the query image to the $\mathcal{T}_{k_c}$ set, we can identify the source camera model by exploiting the related closed-set classifier. This procedure is not needed in case $C = |\mathcal{T}_k|$, i.e., $C = 36$. In this case, we can exploit the maximum score returned by the C binary classifiers. If the maximum score is related to the "known-known" class, the estimated source camera model is associated with the binary classifier providing the highest score. If the maximum score is related to the "known-unknown" class, the estimated source camera model is unknown. For example, Fig. 7.13 shows the confusion matrix achieved in closed-set classification by training 36 classifiers. The average closed-set accuracy is 99.56%.

Exploiting $C = 36$ binary classifiers outperforms the previously presented closed-set classifier, both in known-vs-unknown detection and closed-set classification. This

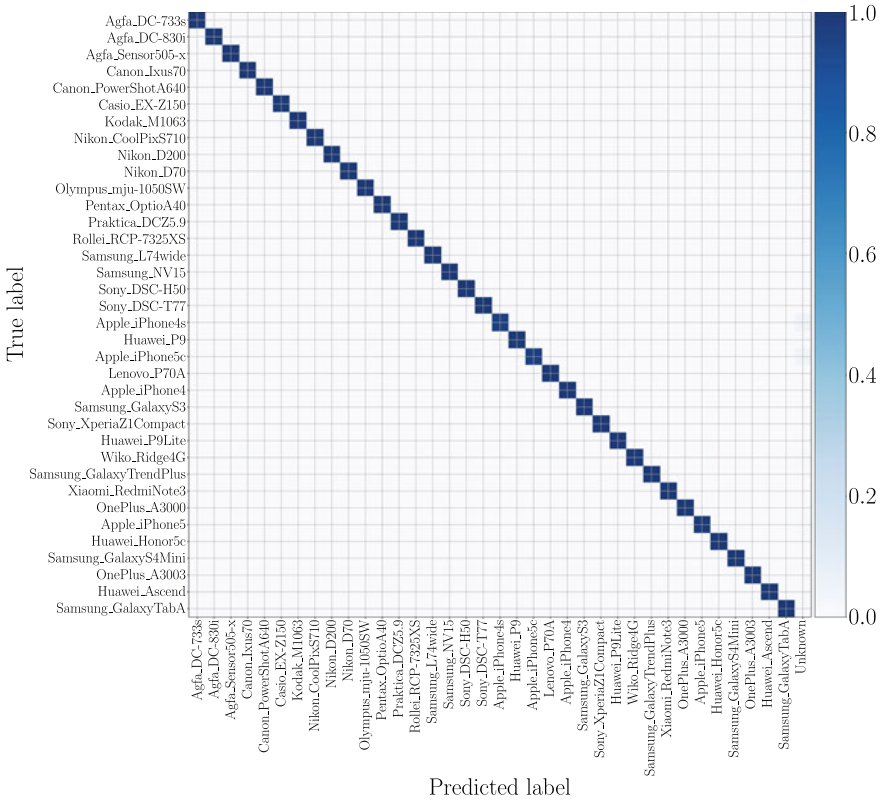


Fig. 7.13 Confusion matrix achieved in closed-set classification by training 36 classifier, considering the 36 camera models in $\mathcal{T}_k$. The average closed-set accuracy is 99.56%

comes at the expense of training 36 binary classifiers instead of one C-class classifier. However, each binary classifier is trained over a reduced amount of images, i.e., twice the number of images of the "known-known" class. On the contrary, the C-class classifier is trained over the total amount of known images. Therefore, the required computation time by training 36 classifiers does not significantly increase and it is still acceptable.

7.6 Conclusions and Outlook

In this chapter, we have reported multiple examples of camera model identification algorithms developed in the literature. From this report and the showcased experimental results, it is possible to understand that the camera model identification problem can be well solved in some situations, but not always.

For instance, good solutions exist for solving the camera model identification task in a closed-set scenario, when images have not been further post-processed or edited. Given a set of known camera models and a pool of training images, standard CNNs can solve the task in a reasonable amount of time with high accuracy.

However, this setup is too optimistic. In practical situations, it is unlikely that images never undergo any processing step after acquisition. Moreover, it is not realistic to assume that an analyst has access to all possible camera models needed for an investigation. In this situation, further research is needed. As an example, compression artifacts and other processing operations like cropping or resizing can strongly impact on the achieved performance. For this reason, a deeper analysis on images shared through social media networks is worthy of investigations.

Moreover, open-set scenarios are still challenging even considering original (i.e., not post-processed) images. The proposed methods in the literature are not able to approach closed-set accuracy yet. As the open-set scenario is more realistic than its closed-set counterpart, further work in this direction should be done.

Furthermore, the newest smartphone technologies based on HDR images, multiple cameras, super-dense sensors and advanced internal processing (e.g., "beautify" filters, AI-guided enhancements, etc.) can further complicate the source identification problem, due to an increased image complexity (Shaya et al. 2018; Iuliani et al. 2021). For instance, it has been already shown that sensor-based methodologies to solve the camera attribution problem may suffer for the portrait mode employed in smartphones of some particular brands (Iuliani et al. 2021). When portrait modality is selected to capture a human subject in foreground and blur the remaining background, sensor traces risk to be hindered (Iuliani et al. 2021).

Finally, we have not considered scenarios involving attackers. On one hand, antiforensic techniques could be applied to images in order to hide camera model traces. Alternatively, working with data-driven approaches, it is important to consider possible adversarial attacks to the used classifiers. In the light of these considerations, forensic approaches for camera model identification in modern scenarios still have to be developed. Such future algorithmic innovations can be further fostered with novel datasets that include these most recent challenges.

References

Avcibas I, Sankur B, Sayood K (2002) Statistical evaluation of image quality measures. J Electron Imaging 11(2):206–223

Avcibas I, Kharrazi M, Memon ND, Sankur B (2005) Image steganalysis with binary similarity measures. EURASIP J Adv Signal Process 17:2749–2757

Bayar B, Stamm MC (2017) Augmented convolutional feature maps for robust cnn-based camera model identification. In:2017 IEEE international conference on image processing, ICIP 2017, Beijing, China, September 17–20, 2017. IEEE, pp 4098–4102

Bayar B, Stamm MC (2018) Towards open set camera model identification using a deep learning framework. In: 2018 IEEE international conference on acoustics, speech and signal processing, ICASSP 2018, Calgary, AB, Canada, April 15–20, 2018. IEEE, pp 2007–2011

Bayram S, Sencar HT, Memon ND, Avcibas I (2005) Source camera identification based on CFA interpolation. In: Proceedings of the 2005 international conference on image processing, ICIP 2005, Genoa, Italy, September 11–14, 2005. IEEE, pp 69–72

Bayram S, Sencar HT, Memon N, Avcibas I (2006) Improvements on source camera-model identification based on cfa interpolation. Proc WG 11(9)

Bondi L, Baroffio L, Guera D, Bestagini P, Delp EJ, Tubaro S (2017a) First steps toward camera model identification with convolutional neural networks. IEEE Signal Process Lett 24(3):259–263

Bondi L, Lameri S, Guera D, Bestagini P, Delp EJ, Tubaro S (2017b) Tampering detection and localization through clustering of camera-based CNN features. In: 2017 IEEE conference on computer vision and pattern recognition workshops, CVPR workshops 2017, Honolulu, HI, USA, July 21–26, 2017. IEEE Computer Society, pp 1855–1864

Bonettini N, Bondi L, Bestagini P, Tubaro S (2018) JPEG implementation forensics based on eigenalgorithms. In: 2018 IEEE international workshop on information forensics and security, WIFS 2018, Hong Kong, China, December 11–13, 2018. IEEE, pp 1–7

Buslaev A, Iglovikov VI, Khvedchenya E, Parinov A, Druzhinin M, Kalinin AA (2020) Albumentations: fast and flexible image augmentations. Information 11(2):125

Cao H, Kot AC (2009) Accurate detection of demosaicing regularity for digital image forensics. IEEE Trans Inf Forensics Secur 4(4):899–910

Cao H, Kot AC (2010) Mobile camera identification using demosaicing features. In: International symposium on circuits and systems (ISCAS 2010), May 30–June 2, 2010, Paris, France. IEEE, pp 1683–1686

Çeliktutan O, Sankur B, Avcibas I (2008) Blind identification of source cell-phone model. IEEE Trans Inf Forensics Secur 3(3):553–566

Chen M, Fridrich JJ, Goljan M, Lukás J (2008) Determining image origin and integrity using sensor noise. IEEE Trans Inf Forensics Secur 3(1):74–90

Chen C, Stamm MC (2015) Camera model identification framework using an ensemble of demosaicing features. In: 2015 IEEE international workshop on information forensics and security, WIFS 2015, Roma, Italy, November 16–19, 2015. IEEE, pp 1–6

Choi C-H, Choi J-H, Lee H-K (2011) Cfa pattern identification of digital cameras using intermediate value counting. In: Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security, pp 21–26

Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: 2017 IEEE conference on computer vision and pattern recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017. IEEE Computer Society, pp 1800–1807

Cozzolino D, Verdoliva L (2020) Noiseprint: a cnn-based camera model fingerprint. IEEE Trans Inf Forensics Secur 15:144–159

Deng J, Dong W, Socher R, Li L-J, Li K, Li F-F (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE computer society conference on computer vision and pattern recognition (CVPR 2009), 20–25 June 2009, Miami, Florida, USA. IEEE Computer Society, pp 248–255

Dirik AE, Sencar HT, Memon ND (2008) Digital single lens reflex camera identification from traces of sensor dust. IEEE Trans Inf Forensics Secur 3(3):539–552

Ferreira A, Chen H, Li B, Huang J (2018) An inception-based data-driven ensemble approach to camera model identification. In: 2018 IEEE international workshop on information forensics and security, WIFS 2018, Hong Kong, China, December 11–13, 2018. IEEE, pp 1–7

Foi A (2009) Clipped noisy images: heteroskedastic modeling and practical denoising. Signal Proc 89(12):2609–2629

Fridrich JJ, Kodovský J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensics Secur 7(3):868–882

Galdi C, Hartung F, Dugelay J-L (2019) SOCRatES: a database of realistic data for source camera recognition on smartphones. In: De Marsico M, di Baja GS, Fred ALN (eds) Proceedings of the 8th international conference on pattern recognition applications and methods, ICPRAM 2019, Prague, Czech Republic, February 19–21, 2019. SciTePress, pp 648–655

Gao S, Xu G, Hu RM (2011) Camera model identification based on the characteristic of CFA and interpolation. In: Shi Y-Q, Kim H-J, Pérez-González F (eds) Digital forensics and watermarking - 10th international workshop, IWDW 2011, Atlantic City, NJ, USA, October 23–26, 2011, Revised Selected Papers, vol 7128 of Lecture Notes in computer science. Springer, pp 268–280

Gloe T (2012) Feature-based forensic camera model identification. Trans Data Hiding Multim Secur 8:42–62

Gloe T, Böhme R (2010) The Dresden image database for benchmarking digital image forensics. J Digit Forensic Pract 3(2–4):150–159

Gloe T, Borowka K, Winkler A (2009) Feature-based camera model identification works in practice. In: Katzenbeisser S, Sadeghi A-R (eds) Information hiding, 11th international workshop, IH 2009, Darmstadt, Germany, June 8–10, 2009, Revised Selected Papers, vol 5806 of Lecture notes in computer science. Springer, pp 262–276

Gloe T, Borowka K, Winkler A (2010) Efficient estimation and large-scale evaluation of lateral chromatic aberration for digital image forensics. In: Memon ND, Dittmann J, Alattar AM, Delp EJ (eds) Media forensics and security II, part of the IS&T-SPIE electronic imaging symposium, San Jose, CA, USA, January 18–20, 2010, Proceedings, vol 7541 of SPIE Proceedings. SPIE, p 754107

Guera D, Zhu F, Yarlagadda K, Tubaro S, Bestagini P, Delp EJ (2018) Reliability map estimation for cnn-based camera model attribution. In: 2018 IEEE winter conference on applications of computer vision, WACV 2018, Lake Tahoe, NV, USA, March 12–15, 2018. IEEE Computer Society, pp 964–973

Gunturk BK, Glotzbach JW, Altunbasak Y, Schafer RW, Mersereau RM (2005) Demosaicking: color filter array interpolation. IEEE Signal Proc Mag 22(1):44–54

Hadwiger B, Riess C (2020) The forchheim image database for camera identification in the wild. arXiv:abs/2011.02241

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016. IEEE Computer Society, pp 770–778

Ho JS, Au OC, Zhou J, Guo Y (2010) Inter-channel demosaicking traces for digital image forensics. In: Proceedings of the 2010 IEEE international conference on multimedia and Expo, ICME 2010, 19–23 July 2010, Singapore. IEEE Computer Society, pp 1475–1480

Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Zheng Q, Yen N-C, Tung CC, Liu HH (1998) The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. Proc R Soc Lond Ser A: Math Phys Eng Sci 454(1971):903–995

Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017. IEEE Computer Society, pp 2261–2269

IEEE Signal Processing Cup 2018 Database - Forensic Camera Model Identification. https://cutt.ly/acK1lg2. Accessed 06 April 2021

Image Source Attribution UniCamp Dataset. http://www.recod.ic.unicamp.br/~filipe/dataset. Accessed 09 April 2021

Iuliani M, Fontani M, Piva A (2021) A leak in PRNU based source identification. Questioning fingerprint uniqueness. IEEE Access 9:52455–52463

Johnson MK, Farid H (2006) Exposing digital forgeries through chromatic aberration. In: Voloshynovskiy S, Dittmann J, Fridrich JJ (eds) Proceedings of the 8th workshop on multimedia & security, MM&Sec 2006, Geneva, Switzerland, September 26–27, 2006. ACM, pp 48–55

Júnior PRM, Bondi L, Bestagini P, Tubaro S, Rocha A (2019) An in-depth study on open-set camera model identification. IEEE Access 7:180713–180726

Kang SB, Weiss RS (2000) Can we calibrate a camera using an image of a flat, textureless lambertian surface? In: Vernon D (ed) Computer vision - ECCV 2000, 6th European conference on computer vision, Dublin, Ireland, June 26–July 1, 2000, Proceedings, Part II, vol 1843 of Lecture notes in computer science. Springer, pp 640–653

Kharrazi M, Sencar HT, Memon ND (2004) Blind source camera identification. In: Proceedings of the 2004 international conference on image processing, ICIP 2004, Singapore, October 24–27, 2004. IEEE, pp 709–712

Kirchner M (2010) Efficient estimation of CFA pattern configuration in digital camera images. In: Memon ND, Dittmann J, Alattar AM, Delp EJ (eds) Media forensics and security II, part of the IS&T-SPIE electronic imaging symposium, San Jose, CA, USA, January 18–20, 2010, Proceedings, vol 7541 of SPIE Proceedings. SPIE, p 754111

Kirchner M, Gloe T (2015) Forensic camera model identification. In: Handbook of digital forensics of multimedia data and devices. Wiley-IEEE Press, pp 329-374

Lanh TV, Emmanuel S, Kankanhalli MS (2007) Identifying source cell phone using chromatic aberration. In: Proceedings of the 2007 IEEE international conference on multimedia and Expo, ICME 2007, July 2–5, 2007, Beijing, China. IEEE Computer Society, pp 883–886

Lukás J, Fridrich JJ, Goljan M (2006) Digital camera identification from sensor pattern noise. IEEE Trans Inf Forensics Secur 1(2):205–214

Lyu S (2010) Estimating vignetting function from a single image for image authentication. In: Campisi P, Dittmann J, Craver S (eds) Multimedia and security workshop, MM&Sec 2010, Roma, Italy, September 9–10, 2010. ACM, pp 3–12

Mallat S (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans Pattern Anal Mach Intell 11(7):674–693

Mandelli S, Bonettini N, Bestagini P, Tubaro S (2020) Training cnns in presence of JPEG compression: multimedia forensics vs computer vision. In: 12th IEEE international workshop on information forensics and security, WIFS 2020, New York City, NY, USA, December 6–11, 2020. IEEE, pp 1–6

Marra F, Poggi G, Sansone C, Verdoliva L (2017) A study of co-occurrence based local features for camera model identification. Multim Tools Appl 76(4):4765–4781

Mayer O, Stamm MC (2018) Learned forensic source similarity for unknown camera models. In: 2018 IEEE international conference on acoustics, speech and signal processing, ICASSP 2018, calgary, AB, Canada, April 15–20, 2018. IEEE, pp 2012–2016

Menon D, Calvagno G (2011) Color image demosaicking: an overview. Signal Proc Image Commun 26(8–9):518–533

Nguyen DTD, Pasquini C, Conotter V, Boato G (2015) RAISE: a raw images dataset for digital image forensics. In: Ooi WT, Feng W-C, Liu F (eds) Proceedings of the 6th ACM multimedia systems conference, MMSys 2015, Portland, OR, USA, March 18–20, 2015. ACM, pp 219–224

Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R (eds) Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada, pp 8024–8035

Popescu AC, Farid H (2005) Exposing digital forgeries in color filter array interpolated images. IEEE Trans Signal Proc 53(10):3948–3959

Rafi AM, Kamal U, Hoque R, Abrar A, Das S, Laganière R, Hasan K (2019) Application of densenet in camera model identification and post-processing detection. In: IEEE conference on computer vision and pattern recognition workshops, CVPR workshops 2019, Long Beach, CA, USA, June 16–20, 2019. Computer Vision Foundation/IEEE, pp 19–28

Rafi AM, Tonmoy TI, Kamal U, Wu QMJ, Hasan K (2021) Remnet: remnant convolutional neural network for camera model identification. Neural Comput Appl 33(8):3655–3670

Ramanath R, Snyder WE, Yoo Y, Drew MS (2005) Color image processing pipeline. IEEE Signal Process Mag 22(1):34–43

Sameer VU, Naskar R (2018) Eliminating the effects of illumination condition in feature based camera model identification. J Vis Commun Image Represent 52:24–32

San Choi K, Lam EY, Wong KKY (2006) Automatic source camera identification using the intrinsic lens radial distortion. Optics Express 14(24):11551–11565

Shaya OA, Yang P, Ni R, Zhao Y, Piva A (2018) A new dataset for source identification of high dynamic range images. Sensors 18(11):3801

Shullani D, Fontani M, Iuliani M, Shaya OA, Piva A (2017) VISION: a video and image dataset for source identification. EURASIP J Inf Secur 2017:15

Stamm MC, Bestagini P, Marcenaro L, Campisi P (2018) Forensic camera model identification: Highlights from the IEEE signal processing cup 2018 student competition [SP competitions]. IEEE Signal Process Mag 35(5):168–174

Swaminathan A, Wu M, Liu KJR (2007) Nonintrusive component forensics of visual sensors using output images. IEEE Trans Inf Forensics Secur 2(1):91–106

Swaminathan A, Wu M, Liu KJR (2009) Component forensics. IEEE Signal Proc Mag 26(2):38–48

Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Singh SP, Markovitch S (eds) Proceedings of the thirty-first AAAI conference on artificial intelligence, February 4–9, 2017, San Francisco, California, USA. AAAI Press, pp 4278–4284

Takamatsu J, Matsushita Y, Ogasawara T, Ikeuchi K (2010) Estimating demosaicing algorithms using image noise variance. In: The twenty-third IEEE conference on computer vision and pattern recognition, CVPR 2010, San Francisco, CA, USA, 13–18 June 2010. IEEE Computer Society, pp 279–286

Tan M, Le QV (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: Chaudhuri K, Salakhutdinov R (eds) Proceedings of the 36th international conference on machine learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA, vol 97 of Proceedings of machine learning research. PMLR, pp 6105–6114

Thai TH, Cogranne R, Retraint F (2014) Camera model identification based on the heteroscedastic noise model. IEEE Trans Image Proc 23(1):250–263

Thai TH, Retraint F, Cogranne R (2016) Camera model identification based on the generalized noise model in natural images. Digit Signal Proc 48:285–297

Tuama A, Comby F, Chaumont M (2016) Camera model identification with the use of deep convolutional neural networks. In: IEEE international workshop on information forensics and security, WIFS 2016, Abu Dhabi, United Arab Emirates, December 4–7, 2016. IEEE, pp 1–6

Wahab AWA, Ho ATS, Li S (2012) Inter-camera model image source identification with conditional probability features. In: Proceedings of IIEEJ 3rd image electronics and visual computing workshop (IEVC 2012)

Xu G, Gao S, Shi Y-Q, Hu RM, Su W (2009) Camera-model identification using markovian transition probability matrix. In: Ho ATS, Shi YQ, Kim HJ, Barni M (eds) Digital watermarking, 8th international workshop, IWDW 2009, Guildford, UK, August 24–26, 2009. Proceedings, vol 5703 of Lecture notes in computer science. Springer, pp 294–307

Xu G, Shi YQ (2012) Camera model identification using local binary patterns. In: Proceedings of the 2012 IEEE international conference on multimedia and Expo, ICME 2012, Melbourne, Australia, July 9–13, 2012. IEEE Computer Society, pp 392–397

Yao H, Qiao T, Ming X, Zheng N (2018) Robust multi-classifier for camera model identification based on convolution neural network. IEEE Access 6:24973–24982

Yu J, Craver S, Li E (2011) Toward the identification of DSLR lenses by chromatic aberration. In: Memon ND, Dittmann J, Alattar AM, Delp III EJ (eds) Media forensics and security III, San Francisco Airport, CA, USA, January 24–26, 2011, Proceedings, vol 7880 of SPIE Proceedings. SPIE, p 788010

Zhao X, Stamm MC (2016) Computationally efficient demosaicing filter estimation for forensic camera model identification. In: 2016 IEEE international conference on image processing, ICIP 2016, Phoenix, AZ, USA, September 25–28, 2016. IEEE, pp 151–155